The Islamic University–Gaza
Deanship of Research and Graduate Studies
Faculty of Information Technology
Master of Information Technology

الجامعـــــــة الإســـــلاميـة – غـزة
عمادة البحث العلمي والدراسات العليا
كـليــــــــــــــة تكنولوجيا المعلومات
ماجستيـــــر تكنولوجيا المعلومات

A Framework of Guidelines for Adaptive Mobile Apps User Interfaces Based on User Model Factor

إطار من الضوابط لواجهات المستخدم ذاتية التكيف بناء على عوامل نموذج المستخدم

By

Ahmed Adnan Qazzaz

Supervised by

Dr. Ashraf M. Alattar
Dr. of Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Information Technology

August/2018

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

A Framework of Guidelines for Designing Adaptive Mobile Apps User Interface Elements Based on User Model Factor

إطار من الضوابط لتصميم عناصر واجهات المستخدم ذاتية التكيف بناء على عوامل نموذج المستخدم

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل الاخرين لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

Declaration

I understand the nature of plagiarism, and I am aware of the University's policy on this.

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted by others elsewhere for any other degree or qualification.

Student's name:	أحمد عدنان القزاز	اسم الطالب:
Signature:		التوقيع:
Date:	2018/ — / —	التاريخ:

بسم الله الرحمن الرحيم

الجامعة الإسلامية بغزة
The Islamic University of Gaza
عمادة البحث العلمي والدراسات العليا
هاتف داخلي: 1150

الجامعة الإسلامية بغزة
Islamic University of Gaza

الرقم ج س غ/35/ Ref

التاريخ 2018/12/01م Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ **أحمد عدنان محمد القزاز** لنيل درجة الماجستير في كلية **تكنولوجيا المعلومات/** برنامج **تكنولوجيا المعلومات** وموضوعها:

إطار من الضوابط لواجهات المستخدم ذاتية التكيف بناء على عوامل نموذج المستخدم

A Framework of Guidelines for Adaptive Mobile Apps User Interfaces Based on User Model Factors

وبعد المناقشة التي تمت اليوم **الثلاثاء 19 صفر 1440**هـ الموافق **2018/10/30**م الساعة **الحادية عشرة صباحاً**، في قاعة اجتماعات الكلية اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. أشرف محمد العطار	**مشرفاً ورئيساً**	....
د. إياد محمد الأغا	**مناقشاً داخلياً**	....
د. سناء وفا الصايغ	**مناقشاً خارجياً**	....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في **كلية تكنولوجيا المعلومات/برنامج تكنولوجيا المعلومات.**

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله تعالى ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق،،،

عميد البحث العلمي والدراسات العليا

أ.د. مازن إسماعيل هنية

عمادة البحث العلمي والدراسات العليا
الجامعة الإسلامية بغزة
The Islamic University
Deanship of Research and Graduate Studies

الرقم العام للنسخة	31069443	اللغة	E

التاريخ: 2 / 12 / 2018

الموضوع/ استلام النسخة الإلكترونية لرسالة علمية

قامت إدارة المكتبات بالجامعة الإسلامية باستلام **النسخة الإلكترونية** من رسالة

الطالب/ أحمد حمدان محمد القزاز

رقم جامعي: 120140231 قسم: تكنولوجيا المعلومات كلية: تكنولوجيا المعلومات

وتم الاطلاع عليها، ومطابقتها بالنسخة الورقية للرسالة نفسها، ضمن المحددات المبينة أدناه:

399

- تم إجراء جميع التعديلات التي طلبتها لجنة المناقشة.
- تم توقيع المشرف/المشرفين على النسخة الورقية لاعتمادها كنسخة معدلة ونهائية.
- تم وضع ختم "عمادة الدراسات العليا" على النسخة الورقية لاعتماد توقيع المشرف/المشرفين.
- وجود جميع فصول الرسالة مجمَّعة في ملف (WORD) وآخر (PDF).
- وجود فهرس الرسالة، والملخصين باللغتين العربية والإنجليزية بملفات منفصلة (PDF +WORD)
- تطابق النص في كل صفحة ورقية مع النص في كل صفحة تقابلها في الصفحات الإلكترونية.
- تطابق التنسيق في جميع الصفحات (نوع وحجم الخط) بين النسخة الورقية والإلكترونية.

ملاحظة: ستقوم إدارة المكتبات بنشر هذه الرسالة كاملة بصيغة (PDF) على موقع المكتبة الإلكتروني.

والله ولي التوفيق،

توقيع الطالب

إدارة المكتبة المركزية

Abstract

It’s obvious that mobile user interfaces can adapt to a number of user characteristics in order to be more usable. However, software developers seeking to create adaptive user interfaces which respond to user characteristics have little guidance as to how to map variations in user characteristics with the choice of interface elements such as layout, choice of elements, and appearance.

In this research we plan to study the various user characteristics such as age, gender, experience level, etc. and map these characteristics with interface elements, to develop a framework of guidelines to advise designers on the decisions related to the choices mentioned above. Furthermore, our research will include the development of a prototype software in order to illustrate and evaluate our framework. Evaluation will also encompass a set of experiments and a feedback questionnaire.

The user characteristics will base on two types of user modeling which are the static info and we will fetch it from user Facebook account, and the dynamic info which will be recorded by tracking the user events while using the app.

Our method will be rule based system, and it will relay on the concept of constraints on elements, so the elements will adapt according to these constraints and the user factors. Also we will make a practical evaluation for the developed app on group of selected users to measure the effectiveness of the idea of adaptive mobile apps interfaces.

The experiment include a group of users who will use a non-adaptive application for a specific period of time, and then we will make the app adaptive, and we will ask the users to use it again at the same period of time, at the end of each time users need to fill a questionnaire to express their appeal and satisfaction about the changes, and the idea as a whole.

The experiments results show a satisfying level of users’ acceptance to the idea, and to the changes made on the app based on the framework guideline in both genders, and different types of ages, also the changes made on the app due to users skills.

The research shows the importance of adaptive interfaces on the aesthetic level, and we recommend that developers should take these variations between users into account, and try as much as possible to apply adaptive interface on their apps.

Keywords: intelligent user interfaces, adaptive user interfaces, user-models, interface design frameworks, usability, human-computer interaction.

ملخص الدراسة

من الواضح ان واجهات المستخدم في تطبيقات الموبايل يمكن ان تتكيف بناء على خصائص المستخدم لكي تصبح أكثر سهولة في الاستخدام، وبالتالي فإن المطورين يبحثون عن كيفية تطوير واجهات المستخدم بحيث تتكيف مع متغيرات خصائص المستخدم، والتي يوجد القليل من الضوابط التي تحدد وتربط بين متغيرات خصائص المستخدم وعناصر الواجهة، مثل ترتيب العناصر وأنواعها وأشكالها.

وفي هذه الدراسة سوف نقوم بدراسة هذه المتغيرات مثل النوع والعمر والخبرة في استخدام التطبيق، وربط هذه المتغيرات بعناصر الواجهة، لنحصل على اطار من الضوابط و التوجيهات التي تساعد المصممين في اتخاذ قرارات مرتبطة بعناصر الواجهة المختلفة، بالإضافة إلى ان هذه الدراسة سوف تتضمن تطوير نموذج تطبيق ليعبر عن هذه الضوابط ولنتمكن من تقييم هذه الضوابط. سوف يرتكرز التقييم على تجربة الضوابط من خلال التطبيق ومن خلال استبيانات للمستخدمين.

نموذج المستخدم سوف يحتوي على نوعين من الخصائص وهى الخصائص الثابتة والتي يمكن الحصول عليها من تطبيق فيسبوك، أو يدخلها المستخدم بنفسه، والخصائص المتغيرة التي يتم جمعها من خلال مراقبة استخدام المستخدم للتطبيق.

سوف نعتمد في هذه الدراسة على نموذج مبني على القواعد، والقيود لكل عنصر، وبالتالي فإن العناصر في الوجهة سوف تتغير بناء على هذه القواعد والقيود المعبرة عن متغيرات المستخدم. أيضا سوف نقوم بعمل تجربة عملية على مجموعة من المستخدمين لقياس مدى فاعلية الفكرة، ومدى صحة هذه القيود، وستكون التجربة تتضمن تطوير تطبيق لا يدعم تكييف الواجهات ويقوم المستخدمين باستخدامه لمدة محددة، بعد ذلك سوف نجعل هذا التطبيق يدعم تكييف الوجهات، ويقوم المستخدمين استخدام التطبيق بنفس الفترة الزمنية، وفي نهاية كل جولة من الجولتين، سوف يقوم المستخدمين بتعبئة استبيان لقياس مدى رضا المستخدمين عن التطبيق، والفكرة بشكل عام، وبالتالي مدى فاعلية هذه الضوابط.

سوف يكون التغير معتمد على جنس المستخدم ومختلف الأعمار، وايضا مهارة المستخدم في استخدام التطبيق.

أظهرت التجربة نسبة قبول ورضا المستخدمين عن الفكرة والتغير المنعكس على الواجهات وبالتالي أهمية فكرة تكييف الواجهات على المستوى الجمالي، و لذا نوصي المطورين والمصممين بأخذ هذه التغيرات في خصائص المستخدمين بالحسبان، ومحاولة اتباع هذه القواعد بقدر الإمكان.

Dedication

I dedicate this thesis to the greatest heart ever, and the endless kindness of my mother and to great icon my father, for their continuous support and endless encouraging.

I dedicate this to my love, to my wife, for her patience, support, and help.

I dedicate this work to my new family, my father-in-law, and my mother-in-law

I dedicate this to my big brother, to my all brothers and sisters

I dedicate this to my teachers, and instructors

I dedicate this to my best friends

Acknowledgment

First of all, thanks always go to Allah, God Almighty the Merciful, for granting me the opportunity to pursue my graduate study and complete this thesis. To Him gratitude and praise is due for our success and all blessings.

The thesis can be seen now in its final format and complete pieces with guidance and assistance of many people, I would like to thank them all for that, and I admit that I owe you one, and I hope someday that I can pay you back.

The greatest gratitude for my supervisor, Dr. Ashraf Alattar, for guiding and helping me in all my steps; for his patience, and cooperation.

I would like to thank my family, and above them all my great father Dr. Adnan Qazzaz, for raising my on moral, ethics, and religion, for his continuance support, and for financial and moral support, and of course I can’t ever forget the greatest heart, and the endless kindness of my great mother, for pushing me up to this step, and keep encouraging my to move forward. And for sure I would like to thank my brothers, and sisters, for their efforts and encouraging.

I would like to express my deepest grateful and thankful to my lovely wife, for everything she done to me, for supporting me to complete my thesis, and for sure I would like to thank my father-in-law, and my mother-in-law.

I would take the opportunity to express my gratitude to the academic staff of Information-Technology faculty in the Islamic University – Gaza. Especially for Dr. Tawfiq Barhoom, Dr. Iyad Alagha, Dr. Alla ElHaless, and Dr. Ashraf Maghari.

Special Thanks to all my friends for their helps and support, especially Mr. Mohammed Hamad for assistance in applying the experiment and analyzing results, Mr. Hytham ElShair for desiging the testing App and other supportive designs, Mr. Abdulhameid Alfayoumi the head of Sanabil TV for his assistance in evaluating the App, Mr.Ismail Aziza, and Mr. Hani Mortaga for allowing me and helping me in applying the experiment in Gaza Sky Geeks.

Table of Contents

Table of Figures

Table of tables

Table of Equations

List of abbreviations

ANN	Artificial Neural Networks
app2Vec	Application to Vector
AUI	Abstract User Interface
	Adaptive User Interface
CRE	Corner Radius Effect
CSS	Cascading Style Sheet
CUI	Concert User Interface
EV	Event
GA	Genetic Algorithms
GUMS	Generic User Modeling System
HCI	Human Computer Interaction
HSB	Hue-Saturation-Brightness
IUI	Intelligent User Interface
LME	Layer Mask Effect
plist	Properties List
RGB	Red-Green-Blue
RGBA	Red-Green-Blue-Alpha
UI	User Interface
UX	User Experience
WHO	World Health Organization
XUL	Extendable User Interface Language

Chapter 1

Introduction

Chapter 1

Introduction

It is widely known that to make a product success it must meet the user's needs, it should reflect his desires, so the user can be persuaded and buy it, this concept is known as User-Oriented Design, which means to design a product based on user factors, emotions, and psychology, this principle known as Kansei Engineering (Hsin-Hsi, Yang-Cheng, Chung-Hsing, & Chein-Hung).

This Chapter is about the origin of the thesis subject, and it describes the terms needed to understand the work presented, also it will introduce the relationship between the user factors and design elements in user interface, and the importance of matching between these factors and elements. This chapter will define the problem statement, the objectives, and the significance of thesis, furthermore, it will present the limitation, and scope of the study.

Let us first define the user experience (UX) and the product, and show the relation between product, and design. The product is the output of a process started by the designer, which is choosing and combining a group of features such as (presentation, style, content … etc.) to create a particular indented product, that was the designer perspective of relation between design and product, on the other hand the user perspective is different he looks to the product design as the satisfaction, pleasure and appeal he can feel towards the product design. (Hassenzahl, 2005)

Mobile Users vary from wide range of different users' characteristics, for example, we find youth and elders are both using same mobile apps, we find males, and females, also using same apps, and we can found these changes on demographics can affect the percentage of a group using an app (Dogtiev, 2018), but we can't deny that there is differences in their preferences based on the human nature and human psychology, To take these differences into account, a user model for the current user should be defined.

User Modeling is a form of personalization, which means to gather information about the user to know much more about this user (User Modeling, 2016), this information is called the factors of user model, such as the age, gender, IT experience, educational level, and etc.

User Modelling can be achieved by asking questions or filling a forms, like in case of users profiles, or it can be done be learning from the user and record events, and behaviours to form a user model such as in case of recommender systems (Brusilovsky, 2007).

User Modelling is a subdivision of human-computer interaction field, and it aims to build up and modify a conceptual understanding of the user, and there is different design pattern for user models (User Modeling, 2016)

Static User Models: in this model there is no learning algorithm used, and the data gathered about the user is unchangeable.

Dynamic user models: this model allows the users to change their interest, thus the data in this model can be changed and take the user goals and needs into account.

Stereotype based user models: based on demographic statistics, the gathered information users are classified into common stereotypes, The application therefore can make assumptions about a user even though there might be no data about that specific area, this model relay on statistics and do not take into account that personal attributes might not match the stereotype

Highly adaptive user models: Highly adaptive user models try to represent one particular user and therefore allow a very high adaptively of the system. In contrast to stereotype based user models they do not relay on demographic statistics but aim to find a specific solution for each user, this kind of model needs to gather a lot of information first.

Automatic User Models can be created using machine learning techniques such Genetic algorithms (GA), and Artificial Neural Networks (ANN), or Rule-Based Expert Systems, and data mining techniques and algorithms will play the role of classifying users in case of stereotypes based user models.

User modelling application can be found in advertisements and marketing industry, so ads systems need to know much more about each user, and need to understand their needs and requirements, so they can show the perfect ads for them to make sure that they will click on it or may read it at least, also we can find that in the Google™ search results, and the YouTube™

recommended videos, all recommender systems in general needs to form a user model based on different factors to enhance its recommendations.

Another application of user modeling could be the automatic user interface personalization which called the Adaptation of User Interface.

Intelligent User Interface (IUI) is a trend of Human Computer Interaction (HCI) that discuss the adaptation of the user interface to personalize the system to match the user intentions and desires, Interface customization is widely exists in many systems, and applications in different forms, one of the most known form is the customizations options which allow the user to edit the interface manually to match his needs, but IUI focuses on automatically adjusting the interface without user inferring (Ehlert, 2003)

Adaptive User interface (AUI) can be defined as the Graphical User Interfaces (GUIs) that automatically change their appearance to address the individual needs of specific users. AUIs can successfully solve common problems in HCI, such as solving the conflicts between different users groups, focus on the wanted and unwanted features, enabling the uses to find what they use frequently, or find what they don’t know so far. (Deuschel & Scully, 2016)

Another definition of AUI could be a software artifact that improves its ability to interact with a user by constructing a user model based on partial experience with that user (Langley, 1991), AUI In general can be categorized into three main categories, which are the User-based, Device-Based, and Environment-Based, the device factors can be seen in different applications right now, such as the Responsive design in Web, and the layout systems in Mobile Application (iOS, and Android) based on the screen size, some applications can benefit from the internet connection media if the user is using Wi-Fi, 3G, or 4G, the application should load images, or should not, also other connections media like Bluetooth, airdrop, …etc., can be found in different apps, The Environment factors are based on different sensors, such as the temperature, noise, motion, … etc.

The AUI can be based on many variables, the user variable usually adapt the interface based on user skills, and knowledge, and it can affect three boards, which are, the Information to be selected and presented to the user, the presentation of information which called the visualization, the User Interface meaning the elements and controls in the interface (Wesson, Singh, & van Tonder, 2009), which is our target in this research.

We indent to apply the adaptation of user interface on mobile devices, by providing a base of events-actions mapping on the layout level, choice of elements, and appearance level, the mapping will represent the best actions should be taken for particular event, and we indent to provide a framework for the developers to achieve this adaptation on mobile devices, our framework will support the interface appearance beside the layout of the interface.

So the main outcome of our study is to create a guidelines framework for developers, and designers telling them the best adaptation should be applied to interface elements based on user profile, the developer, has the ability to choose any user modeling technique, to form a knowledge about the user, also he can create his own methods for adapting elements, we indent to give an example of automatic adaptive interfaces on iOS apps.

Our evaluation methodology will focus on the satisfaction of users, so we are going to make a test-bed application without using adaptation and make an experiment on a group of users, then we will use the same application with adaptation on the same group, to measure their satisfaction, and pleasant about the app before and after adaptation using survey.

We will explore in details some of the related works and the state of the art in chapter two, and the next section will identify our problem statement, and our objectives.

1.1 Problem Statement

Software are widely used over the world, and different types of users can use this software, and with mobile devices spreader out more users are engaged in software apps, these users vary based on gender, culture, experience, or educational level, which means that some app could not be usable for a group of users due to their abilities. Therefore, developers still need a design framework to guide them while designing and developing applications to accessible for all types of users, and match the different preferences for each users' variables, yet is still there is a lack of guidance for designers and developers to map these characteristics with user interface's elements.

1.2 Objectives

The objective of this research is composed of a main objective and a set of specific objectives.

1.2.1 Main Objective

Our objective is to develop a theoretical design framework that maps between user factors and adaptive user interface components, and demonstrate this framework by developing a pilot mobile app.

1.2.2 Specific Objectives

To achieve our main objective, we need to achieve some specific objective.

- Study the user models, and check out the gender, age, and IT skills, and define each of them, and define the effects for each factor.
- Search in user modeling techniques, and intelligent user modeling methods, to finally use one of them to create our base model that we will use in our study.
- Form our technique for mapping between user model, and adaptation for the interface.
- Develop a small application that can gather information about user, and form a user model, and will apply our framework on this application.
- Run a survey and experimental tests to check the efficiency of our technique.

1.3 Research Methodology

To achieve the main and specific objectives, this research project has been composed of a the following sequence of activities:

1- Define the Framework

1.a) Define user factors, and the interface elements characteristics.

In this step we will define the factors, and how it can make a difference between users, and how it can affect the user psychology, and his reaction to specific element in the app. Also we will define the element characteristics that we will include in our scope, and why did we pick them.

1.b) Find the relations between each user-factor, and each interface elements, to form our meant framework.

It is the core of our study to find the relation between the user-model factor and each interface element characteristics, and this will lead us the find out how could the user model affect this characteristic, or what characteristics can be affected by a specific factor, and what characteristics never affected by a factor.

2- Implement the framework by a test-bed application.

It is worthy to implement the theoretical framework that we concluded up with it in the previous step, and we are going to automate it in an iOS app that could be used with any other iOS app, so we can test our theory. In this step we will implement a suitable type of app, than can test most of the framework guidelines, and be ready for testing and evaluation, this application should serve both genders and be usable by different ages.

3- Evaluate the application with, and without the framework.

To evaluate our theory, we need to select a group of users that covers all the differences of users factors, and then let them use the app without adaptation for a limited period of time, then they will start use the app with adaptation for another period of time, after that we will estimate their satisfaction about the adaptations that happens while they are using the app.

1.4 Scope and Limitations

As mentioned before, we will focus on a group of factors, and we will ignore others, also we will not cover all the interface components.

Our scope can be summarized in group of points:

- The framework will take into account the genders, ages, and usage patterns (Skill level) of the users, and will ignore any other factors, such as device-factors, and environment-factors.
- The adaptable UI components that will be considered are the background and foreground colors for UI items, shapes, appearance and disappearance for items, items order, and types of items
- The framework covers only the adaptation part of the process, and will not search in the user molding and data gathering theories and techniques.

The limitation can also be a summarized in a set of points:

- The framework will take iOS UI components only, and ignore the Android UI components.
- The framework will not be able to consider all possible user-factors, such as culture, education, religion …etc.
- The framework testing will not cover all users ages groups, and the interface adaptations mentioned in the framework due to the lack of iPhone users especially in kids, and elders.

1.5 Significance of thesis

Due to the extreme and ubiquitous use of softwares all over the world, and the variety of devices, and platforms, it become a challenge to make accessibility to these softwares easy for every user regardless their experience level, their age, their culture or gender, every user want to use these softwares, and he or she want the software meet their needs, understand them and taking responsibility over them, even colors, and shapes of items they might need it to meet their preferences.

The findings of this research will bring a new style for application presentation, and lead us to new technique in adaptability researches, the result method desired to make the software integration in the human life more and more, and this will add values on saving times, and efforts, to learn, or understand a new software, also in using the software and performing tasks.

Clearly we indent to make the software learn the user preferences, favorites, and behaviors, so it will give him what he want with the easiest way possible, instead of the user learn the software to use it.

Also this study opens the way for other researchers to search in these guidelines and framework, to enhance, fix, and update the suggested framework, or any previous frameworks

Chapter 2
Literature Review

Chapter 2

Literature Review

Many applications and researches were made searching on adaptive interfaces based on the device, and environment factors, and it starts early in eighties, and nineties, in 1986 Tim Finin published his framework “Generic User Modeling System” (GUMS) which allows the programmers of user-adaptive applications, to define simple stereotypes hierarchies, later in 1994 Branjnik and Tasso publish UTM which allows the user model developer to define a hierarchical ordered user stereotypes, and rules of user model inferences (Hall & Hothi, 1998) The research in this field continues till now, in 2013 “Cross-system user modeling and personalization on the Social Web” discussed the social web which provides opportunities to gather user data from outside the system itself, and it distributed form-based and tag-based user profiles, based on a large dataset aggregated from the Social Web.

2.1 Adaptive User Interfaces

Cedar is an open online research started in 2013 by Alkiki, Pierre, Brandara, Arosha and Yu Yijun, and it is an integrated development environment (IDE) supporting adaptive model-driven user interfaces for enterprise applications. In: Fifth ACM SIGCHI Symposium on Engineering Interactive Computing Systems (Akiki, Bandara, & Yu, 2014)

Cedar IDE based on model-driven approach, and it provide a UI environment to be used by developers to create the adaptive user interfaces, Cedar focuses on the tool that will be used to achieve the UI adaptation. Cedar is an evolution of previous works, it supports the lack of modeling, generation and synchronization, of all levels of abstraction.

Cedar and other previous research focuses on the UI adaptation on the context of use, and level of user experience, and most of these researches focus on desktop application, and some researches are looking in the web applications based on semantic web (Hervás & Bravo, 2011), and other researches searches in the mobile UI, which is most popular industry nowadays.

Cedar studio: An IDE supporting adaptive model-driven user interfaces for enterprise applications (Akiki, Bandara, & Yu, 2014)

Cedar studio is an IDE for intelligent adaptive user interface, and it aims to create an interface that can be changed automatically according the user experience, the user could be novice, medium, or expert, the components of the interface will have changed dynamically, and converted to other components.

Cedar IDE allows developers to create an interface and define the roles, and options for each component, so expert users will see the complex interface, but novice users will see the simple interface.

The Cedar IDE is a great example for adaptive user interface, it customizes the layout and components based on user experience, but classifying users into novice, medium or expert is done manually, and there is an admin should tell the system the experience level of each user, another note about Cedar is handling the experience factor only. But still many factor of user factors could affect the resulting layout.

Cedar IDE did not propose or assume some basic guidelines for developer to follow, its providing a tool but the developer should expect the suitable adaptation and apply it.

We aim to extends this experience to be based on user gender, and age, also we will apply the changes on the layout, components, and colors, and an important added value to our research is the automation of user profile and classification to novice, medium, or expert user automatically based on his interactions, Also or core contribution is to guide the developer for the best actions to be applied on interface elements, not providing a tool.

Model-based layout generation (Feuerstack, Blumendorf, Schwartze, & Albayrak , 2008)
This paper offers a model-based layout generation that will be evaluated and generated at run-time, to ensure consistent user interface layout, and to allow more flexibility for context-of-use adaptation.

The author here, uses five layout statements to interpret the layout model which in turn stimulate the screens to be present, these statements are the Context, Tree Task, Abstract User Interface (AUI), Concert User Interface (CUI), and Dialog model.

The layout modeling basically consists of main six properties, which is characteristic, Design model, Context-of-use, Priority, Condition-type, and scope, each one of these properties contains a group of possible factors.

This approach can be achieved by sequence of steps, starting from addressing the layout characteristic, defining new layout statements, weight the relations between elements, then applying constraints automatically to all screens based on the layout statements.

The approach present acceptable results according to their measurements, and evaluations, but it based on device factors, which is the screen-size, and the screen orientation with additional possible priorities added by the user, this approach is similar to the current Auto-Layout technique which is used in Apple™ XCode IDE, to layout the components on the screen for iOS™ and MAC OS X™ application, based on the screen size, and the device orientation, in addition to priority factors.

We aim to use the user-model or user profile to make adaptation on screen instead of device-model, and we will not adapt the layout of the elements, we will adapt the components types, appearance, style, and colors

Design by Example of Graphical User Interfaces adapting to available screen size (Demeure, Meskens, Luyten, & Coninx, 2009)

The design by example of adaptive user interface is technique proposed in 2009, it defines a design space as a set of user interfaces that have similar behavior and goals and support the same set of interaction tasks, Each UI in this design space is appropriate for a certain range of screen sizes.

The example in the paper and it is also shown in YouTube™ vide (Alexdmr, 2008) o illustrate a component for presentation application (e.g. PowerPoint, keynote) allows the user to select slide, and using this tool you will define the behavior of the components on the screen at different situations based on some factors, like expanding horizontally or vertically, or compressing the screen in both side, or otherwise.

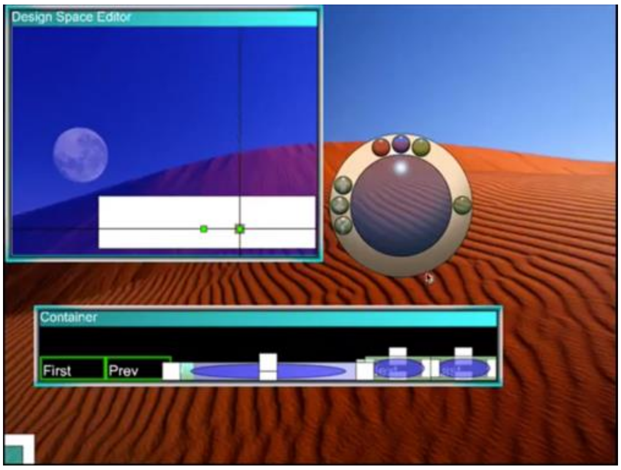
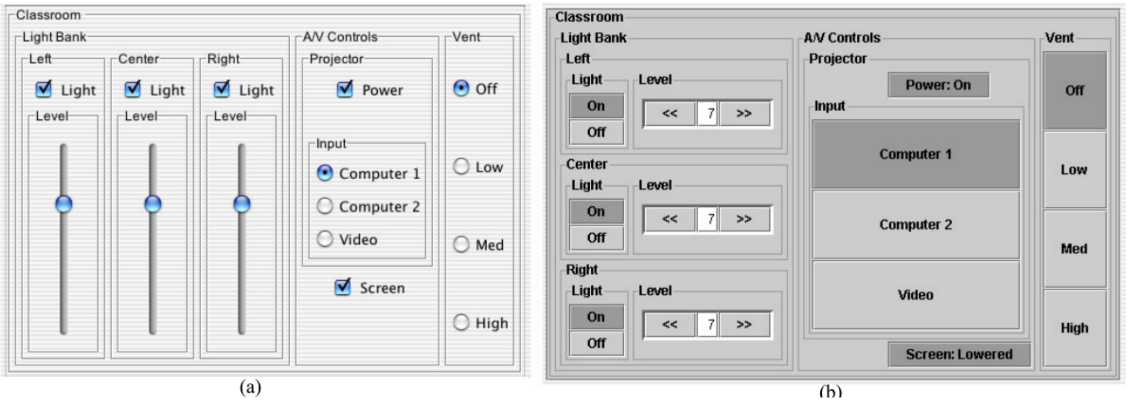


Figure 2.1-1 Design by Example (Alexdmr, 2008)

figure 2.1-7 illustrate three cases if the target screen and the adaptation reflection on each case, this method focuses only on the screen size, for adaptation, and it similar to the old technique used by Apple™ in XCode IDE, which is Auto Resizing Mask, but lately Apple™ replace the Auto Resizing Mask techniques with Auto Layout technique, which is based on defining constraints and priorities for each element, and defines the layout statement for the relations between elements.

SUPPLE: Automatically Generating User Interfaces (Gajos & Weld, 2004), this study proposed an optimal solution for interface rendering based on several factors, and it takes into account the device factors, environmental factors, and usage pattern factor. These factors are mainly classified into three variables, which are the device model (D), Interface specifications (I), and the user trace (T), the study applies the branch-and-bound constrained search algorithm that guarantees to find the optimal solution for interface rendering.



(a)

(b)

Figure 2.1-2 Pointer-based device v.s. touch-panel device (Gajos & Weld, 2004)

Figure 2.1-4 shows the two different interfaces rendered on two devices with the same screen size, but with different interaction method, and there are many examples in this paper that has been tested on different device, different environmental factors (lights), and different usage patterns.

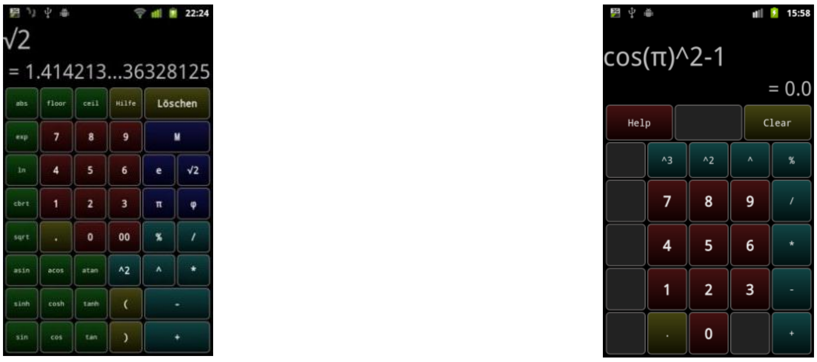
The study based on mathematical formulas and fixed decisions to find out the optimal interfaces for different cases, we are going to make it more fixable to since we are not going to fix the decisions that could be taken for different cases, and we mix many interface elements characteristics with only user model factors, also we will not assume that it is a fully automatic generated interface since there is a cold start for each app, so the designer still have the original concept of design over the adaptation framework.

2.2 Adaptive Mobile User Interfaces

On the Development of Smart Adaptive User Interfaces for Mobile e-Business Applications (Holzinger, Geier, & Germanakos, 2012)

This paper applied on mobile apps, to test the users' satisfaction of adaptive mobile app, the testing app was an adaptive calculator which can adaptively change the buttons of operations based on the current proceeding operation, this change of buttons allows for more space in the interface as shown in figure 2.1-4 and this lead to large button in this calculator, and large button preferred by large group of users, based on the research experiment. On the other hand, the users think this may become confusing when the buttons changes locations, appearing, and disappearing.

In general, the experiments result was most users are satisfied with the adaptation of interface. By comparing this experiment with our intended work we found that we share the idea of change element positions, and appearance of elements, but we will try as much as we can describe the guidelines of changing elements, and appearance to protect the app design from unsatisfying results.



a

b

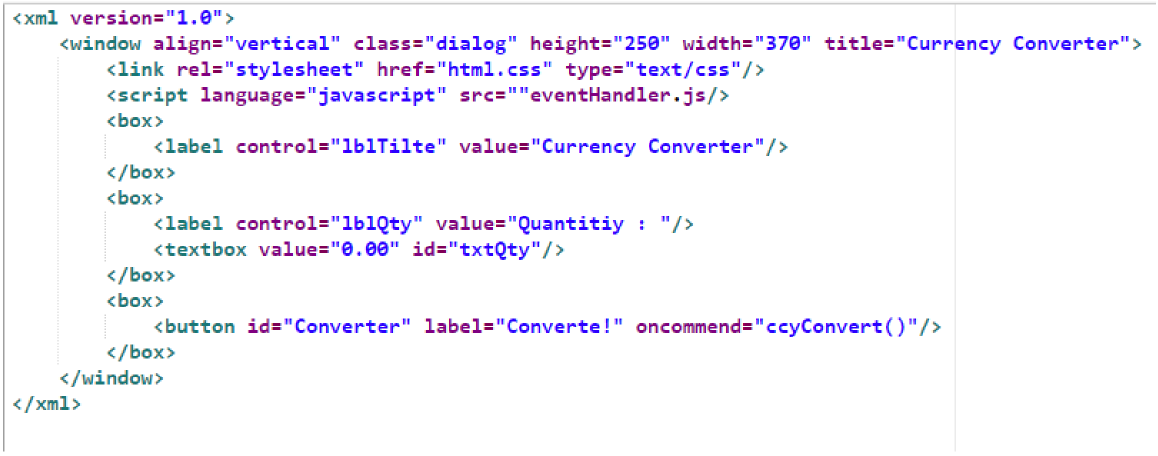
Figure 2.2-1 (a) AdaptiveCalc App Interface, (b) Non-Adaptive calcuator app interface   (Holzinger, Geier, & Germanakos, 2012)

Adaptive User Interface for Mobile Devices (Metrovic & Mena, 2002).

The main goal of this research was to build a solution for cross platform GUI, so the application could build it own GUI based on the device it works on, their solution based on mobile agents, these agents use third-party languages such as, eXtensible User Interface Language (XUL), jXUL middleware and XSL transformations as script shown in figure 2.1-5.

The idea of this work is to transparently adapt graphical user interface by using mobile agent systems. Agents are highly mobile, and are often hosted by platforms that support different models of user interface or have different processing capabilities.

The solution built on Extensible User-interface Language - XUL, which uses the Cascading Style Sheets (CSS) to define the layout of elements, and the JavaScript files to hold functionality



```
<xml version="1.0">
    <window align="vertical" class="dialog" height="250" width="370" title="Currency Converter">
        <link rel="stylesheet" href="html.css" type="text/css"/>
        <script language="javascript" src=""eventHandler.js/>
        <box>
            <label control="lblTilte" value="Currency Converter"/>
        </box>
        <box>
            <label control="lblQty" value="Quantitiy : "/>
            <textbox value="0.00" id="txtQty"/>
        </box>
        <box>
            <button id="Converter" label="Converte!" oncommend="ccyConvert()"/>
        </box>
    </window>
</xml>
```

Figure 2.2-2 XML for Currency Converter Window, with two labels, one text field, and one button (Metrovic & Mena, 2002)

This research is closer to the responsive design principles, since it aims to run the application on different platforms, and different devices and re-arrange items to fit the screen, it uses the XML as implementation language, and the idea of using CSS for style and JavaScript for functionality is similar to Phone Gap and Ionic cross platforms.

It clear that we will focus on the user model instead of device or platform factors, and we will not focus on layout only, but will check the appearance, and selection of elements too.

2.3 Adaptive User Interface Based on User Model Factors

Exploring the Design Space for Adaptive Graphical User Interfaces (Gajos K. a., 2006)

The study was two experiments for three adaptive user interfaces, evaluated against the non-adaptive baseline. The results of these experiments was to find out why some adaptation can produce satisfaction, and other may not.

The researches here design three adaptive techniques would represent three different points in cost-benefit space.

The three techniques are the Split-Interface, the Moving-Interface, and the Visual Popout interface as shown in figure 2.1-1, 2.1-2, and 2.1-3 respectively.

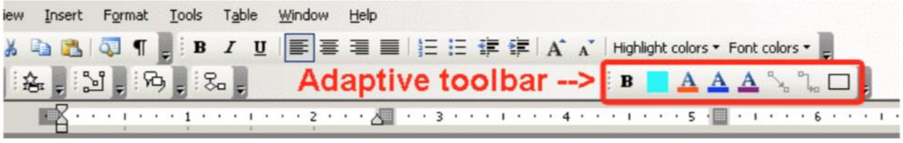


Figure 2.3-1 The Split Interface (Gajos K. a., 2006)

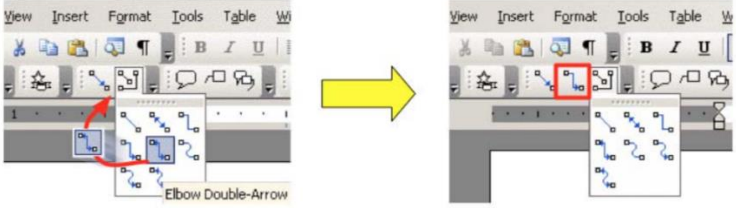


Figure 2.3-2 The Moving Interface (Gajos K. a., 2006)

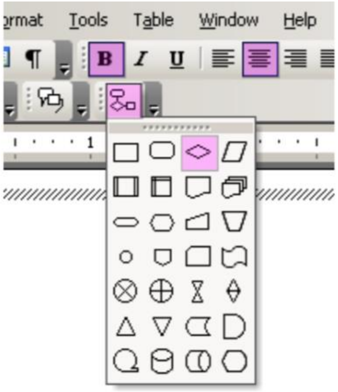


Figure 2.3-3 The Visual Popout Interface (Gajos K. a., 2006)

Their experiments' questionnaires where focusing on three issues: easy of discovery, use of the adapted function, and the confusion caused by the adaptive interface. In general users where less satisfied with the split-interface because users need to look in two different places for one functionality, and they think there is no benefits from grouping frequently used elements together. The moving-interface was more convenient, and some of the experiment's participant prefer it overall.

The researches here focuses only on the usage pattern which sometimes will not be enough for full interface adaptation, also they take the functionality into account and deny the aesthetic of interface adaptation, we can consider that the part of usage pattern is mutual between our indented research and this existing research, but we are going to add other users' factors inconsideration, and we will mainly force on the aesthetic concepts not the functionality of system, and user efforts.

Improve performance, perceived usability, and aesthetics with culturally adaptive user interfaces, this paper aims to check the feasibility of adaptive user interfaces based on culture background, the term culture can be influenced by several factors, stating from the place which is the current residence country, the original traditions of user nationality, user's parents original nationality, and language, event political views can affect the thinking and preferences for the user and all can be considered as culture's factors. . (Reinecke & Bernstein, 2011).

The paper here considered the residence country, and the nationality as the factors of users' culture, they derive rules to map between culture's factors and interface influences, then apply these rules on a web application called MOCCA, the develop a U.S. version of web site, and an adaptive version, and let the users use these websites, by experiment they found out that user's satisfaction increased by 22% with the adaptive version.

We are going to do almost the same steps that have been done on this paper, but we will consider that the factors are the user gender, age, and skill instead of gender, also we will test our hypotheses on mobile app instead of web application. The result we expect to be similar to this paper that we might have an increased level of satisfaction on the adaptive app, and we expect to deliver a summarized rules of interface adaptation guidelines.

2.4 Adaptive User Interface Guidelines

Culturally adaptive user interfaces (Reinecke, 2010)

This thesis introduces a model for intelligent adaptive user interface based on the user culture, the factors that are included in this model, are the current and previous residence, the nationalities of parents, religion, and educational level. The thesis focuses deeply on the culture aspects, and try to find out the relations between the users' cultures background, and the influences to the user interface, and it tried to map these aspects with the interface elements.

The interface adaptation described in this thesis, was about the aesthetic elements such as colors, images, and elements arrangement.

First the study tried to introduce a cultural user model for each user including the nationality, father nationality, mother nationality, mother tongue, the second language, country of residence, educational level, religion, and other variables.

The second step was to match between the user interface adaptations, and the user model aspects, such as in case of religion the adaptation might be applied on religious symbols, color schemes, brightness and contrast.

The study introduces a framework that acquire user culture background and then make a calculation for each aspect to find out the effectiveness of this aspect. Next the framework will apply a group of rules for adaptation, to finally generate the adaptive interface.

This thesis is closed enough to our research, since both are based on user-model, and both targeting to give a framework for mapping between user-model variables, and the interface elements aesthetics. The only difference between both studies is the user model variables, since we focus on age, gender, and usage pattern, but she focuses on cultural aspects and ignored the age, gender, and skills completely.

2.5 User Modeling Adaptation Related Works

Generic User Modeling (ALFRED, 2001), reviews the user modeling systems over twenty years, within user-adaptive systems starting from the origin of user modeling in 1978 by Allen, Cohen and Perrault until 90’s, it discusses the early examples of user-modeling systems such as UMT system (Brajnik & Tasso, 1994) which is based on stereotype modeling, and it classifies the users. And the paper predicts the future generic user modeling in the 20’s to support smarter appliance, and to be multiple-purpose usage system.

According to this paper there is many different characteristics for user modeling systems, during the ages of 80’s and 90’s, in our study we are going to focus on some of there characteristics such as the non-redundancy of stored user information, and the priorities of user information, we might not benefit from the tools that was exhibited in the study because nowadays there is much better tools and methods.

Vector modeling of mobile apps and applications (Lopez, Merlino, & Rodriguez-Bocca, 2017), the paper proposed a new way to modeling users of mobile apps, during the user use apps, the study mainly about the sematic relation between apps that a user may use, in the study they present the apps as document, and apply the word2Vec modeling on these documents, and this approach called app2Vec, the study takes many variables that can makes apps related and takes the session time for each app into account.

In our study somehow we will define the sematic relations between app interface elements and we will apply the formula and the method of app2Vec but on interface elements user-interactions, and of course we will represent the interface elements interfaces as vectors.

Chapter 3

Proposed Framework

Chapter 3

Proposed Framework

As we pointed in the scope of the study we will focus on three user-factors, two of them are static factors, and the other is dynamic factor, in this chapter we will defined the relation between the three factors, gender, age, and skill/Experience with the interface elements characteristics which are the color, shape, icons, positions, visibility, and element type.

We choose the three user factors (gender, age, and skill level) for several reasons, one of the most important of these reasons is the variety of groups can be generated from combining these components, also by exploring the related works and previous researches we found that there is a few researches on these factors and that can be considered as a motivation to study them and use them in the framework, as we saw in the previous works, there is less studies focuses on these factors.

1- Gender factor: can be classified into three group for male, female, and the unknown case, each gender has different preferences and desires, as we will explain in details in this chapter.

2- Age factor: can be grouped in many ways based on several factors and characteristics, and we will explain each method in section 2.

3- User skills factor: sometimes we refer it as usage pattern since the user skill level can be detected from the usage of the app, and here we will not focus only on the ability of using the app, we also take the interest features in the app into account, since detecting the usage pattern can generate these too.

User Interface can contain several elements that may be affected by different factors, and become easy to use, these elements can be listed as in Google™ Material Design™ (Google Inc.)

1- Motion

2- Style

3- Layout

4- Components

5- Pattern

6- Gestures

In our study we will focus on style, which is the truly representation of interface, because style contains the elements of interface such as Colors, Icons, Images, and Typography. These elements are characteristics for the components mentioned in the components section, such as Labels, Buttons, Navigation bars, Dialogs, List, Check Boxes, and others.

All the interface elements can be placed in four main sets which are (Weiss, 1994)

* Presentation: the way that present or display information to the user

* Conversation: the communication between user and system

* Navigation: the way of moving between different parts of information

* Control: the way of making activities and events from the user on the system

These four sets should be taken into account in UX while designing an interface for any platform, especially on mobile platforms, since the mobiles are now used by different types of users with different characteristics, we select a group of these factors or characteristics as mentioned before, in this chapter we will identify the impact of each factor on the user interface.

3.1 Relation between Interface elements, and Gender

Gender can be classified to two concrete classes which are male, and female, also we can make a third class which is the unknown class as a cold start for our app, and because on first run of an application may not know the user gender, so we will take this possibility into account.

So The Gender class now contains three values, which are Unknown, Male, and Female

Each of these value, may have special effect on the interface elements we previously mentioned, or may not.

3.1.1 Impact of gender on colors

the first thing that designers think about when starting a design, is the color schema (Designing a user interface in 5 steps, 2014), and it is also the first thing that the human eye can recognize at the interface at first look, because the human eye can heavily affected by the colors bright (Color & Vision Matters), So we want to know the colors impact on human eye in general, then the effect of colors on each genders' emotions, and moods.

3.1.1.1 Differences between Men's, and Women's colors

We will divide the colors attributes to five attributes, and find the relation between gender and these attributes

The five attributes for colors are derived from Joe Hallock experiment (Hallock, 2003), this study based on the breaking down of the colors preferences experiment made by Neil Patel (Work, 2011)

1- Favorite Colors

Hallock experiment shows that 57% of men prefer the blue color, and 35% of the women prefer it two.

But 0% of men preferred the purple color, in the other hand 23 % of women preferred the purple color.

The second choice color for men was the green color with 14% of the experiment participants, and the same percentage of the green color preferred by women.

So the favorite color at all is the blue color, but the second option of men is green, and for women is purple.

2- Least Favorite Colors

men agreed on brown is the least favorite color with 27% of participants, but women choose the orange color to be the least favorite color with 33% of participants.

The colors black, blue, and red has no place in the least favorite colors for women, but men choose these colors with 1%, 1%, and 2% for each color respectively.

From the two previous attributes 26% of participants agreed on orange is the cheapest color between colors.

3- Bright vs. Soft

The experiment shows that women prefer the soft colors, and men prefer the bright colors, but both women, and men has the same preferences about the light, and dark colors.

4- Tints vs. Shades

The previous experiment shows that men prefer the shades colors, and women prefers the tints

5- Colors Naming

Most men can group collections of colors in one name, and may not differentiate these degrees of the colors, for example, the purple color according to men, could be grape, plum, Eggplant, Orchid, or Lavender according to women.

Based on these differences we can confirm now that apps for women, should differ from apps for men, and colors suitable for men, may not be suitable for women, as shown in figure 3.1-1 we design two different interfaces to illustrate the differences we established earlier based on Hallock experiment.

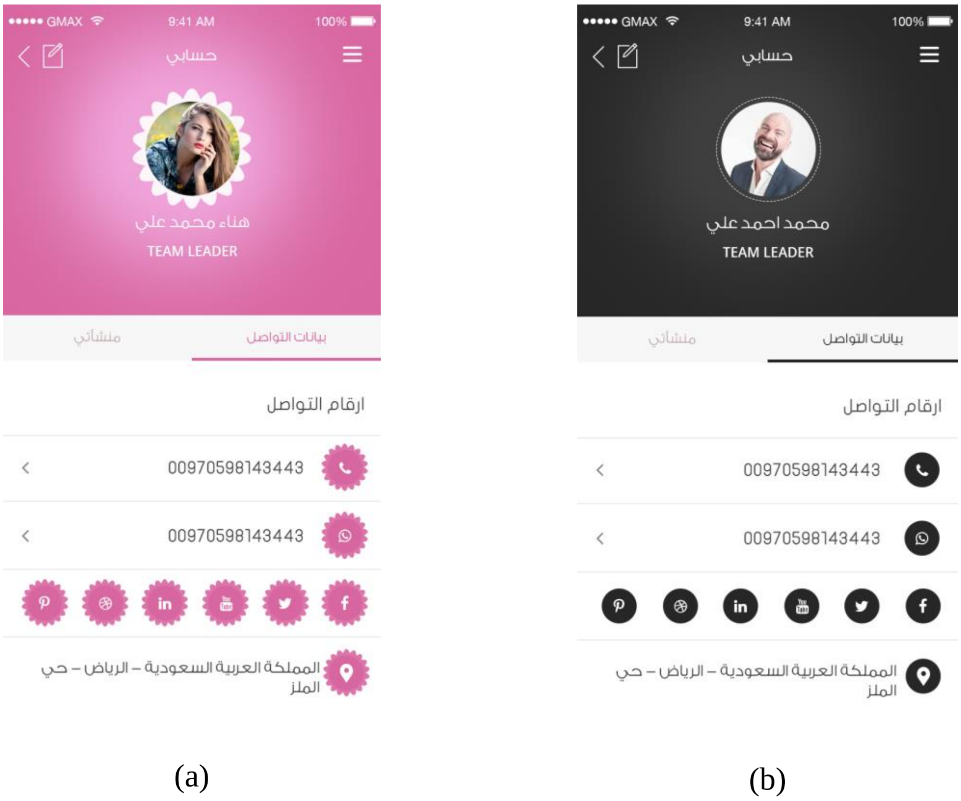


Figure 3.1-1 Feminine and masculine Profile Design, (a) female user, (b) male user

The user interface colored elements should be suitable to each gender since we did establish that there are differences between preferred colors for each gender.

Tint pink, and tint purple may be suitable for female interfaces, on the other hand, the blue color may be good for male users, but should not deny the fact, that each color may contain an emotion, or a reflection to the human psyche. i.e. if the designer meant to make the app blue to express trust, or dependability on the app, or strength of the service, so we cannot flip the color complete to a different hue and make the app pink for the girls' side, because pink, expresses the excitements and youth to the human emotions. We need to modify the tint, and shades of the color, or the brightness and saturation of the selected color.

Table 3.1-1, shows the expected colors for both genders as we can conclude from the previous studies, we set three possible priority for the colors, means that if the colors hue it self matter to the design or it can be altered to a complete different hue, the high priority means that hue is important and the adaptation should stay close as much as possible to the original hue value, and low priority is the complete opposites meaning that color's hue is not important and the only matter is to match the user preferences. Table 3.1-2 represents the same concept of altering the colors but in that case we take the gray scaled color as a special case, since the gray scale colors are colors with low saturation and they can be on the same hue, we supposed that male colors could stay on the shades range with a little more saturation, but the female colors would be better to increase brightness to be in the tint range.

Table 3.1-1 Colors Hue Adjustment

	Male	Female
High Priority	🟦 🟩 🟥 🟧 🟪	🟦 🟩 🟦 🟪 🟪
Medium Priority	🟦 🟩 🟫 🟫 🟪	🟦 🟩 🟥 🟨 🟪
Low priority	🟦 🟩 🟫 🟫 🟪	🟩 🟨 🟥 🟨 🟪

Original Colors 🟦 🟩 🟥 🟨 🟧 🟫

Table 3.1-2 Gray Scale Colors Adaptation

	Colors
Original	⬛ ⬛ ⬛ ⬜
Male	🟦 🟦 🟦 🟦
Female	🟦 🟦 🟦 🟦

You can notice a wide variation for girls' colors, and the reason behind that is the naming feature of color, male users may consider many green colors to be only green, but female will name them, Lime, spring, Fern, Clover, Flora, and Moss.

3.1.1.2Colors Adjustments Methods

There is many standers to represent colors in computer, one of the most known representations is the RGB, and RGBA colors, but this representation will not be useful in our colors adaptation since it need many complex calculation to adjust the three color channels to get the same color with higher brightness, or different exposure, so the appropriate choice for color representation to be used is the HSB colors, in this representation the color degree is held by the first channel which is the Hue, the saturation of the color held in the second channel, and the brightness of the color held in the last channel, so we can get a shader or tint from the same color by adjusting the saturation and brightness values. (Gardner, 2009)

Adjusting saturation or brightness need to be based on one value we called the shifting factor, this factor's value can be set by experiment. It range from 0 to 1
We choose the percent 15% (0.15) as our factor by assumption.

There is various methods to adjust saturation, brightness, and hue (Gardner, 2009), and we will explore some of these methods.

Adjust Saturation

Equation 3.1-1 $s' = s - (s * f)$

Where

s’: Adjusted saturation

s: Current saturation

f: Shifting Factor = 0.15

Adjusting Brightness

Equation 3.1-2 $b' = b + (b * f)$

Where

b’: Adjusted Saturation

b: Current brightness

f: Shifting Factor = 0.15

Adjusting Hue

Based on the True-Colors info-graphics that is derived from Hallock experiment, and previous researches on gender of colors (Work, 2011), we can conclude that the colors for masculine colors are in the range of 200˚ to 260˚ which is the range of blue colors, and the feminine colors are in the range of 261˚ to 340˚ which is the range of pick and purple colors, others colors considered to be acceptable for both genders.

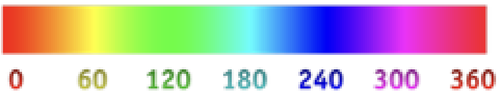


Figure 3.1-2 Colors Hue Spectrum (MCGREGOR, 2016)

3.1.1.3 Colors flipping mechanisms

Complementary (180)

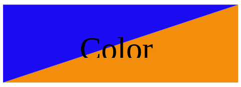
Complementary colors are a combination of two colors, one with H value and the other with H’ value which is a way from H about 180˚ on the Hue wheel.

Equation 3.1-3 $H' = |(H + 180) - 360|$

Split Complementary (150, 210)
Three-color combination that consists of a base color H, and H1, H2 colors

Equation 3.1-4 $H1 = |(H + 150) - 360|$

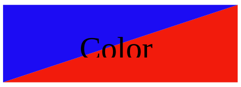
Equation 3.1-5 $H2 = |(H + 120) - 360|$



Triadic (120, 240)
Three-color combination that consists of a base color (H0) and two colors (H1 and H2) that are 120 degrees and 240 degrees apart from H0 respectively

Equation 3.1-6 $H1 = |(H + 120) - 360|$

Equation 3.1-7 $H2 = |(H + 240) - 360|$

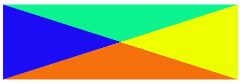


Tetradic (90, 180, 270)
Four-color combination that consists of a base color (H0) and three colors (H1, H2, and H3) that are 90 degrees, 180 degrees, and 270 degrees apart from H0 respectively.

Equation 3.1-8 $H1 = |(H + 90) - 360|$

Equation 3.1-9 $H2 = |(H + 180) - 360|$

Equation 3.1-10 $H3 = |(H + 220) - 360|$



Analogous (30, 60, 90)

Equation 3.1-11 $H1 = |(H + 30) - 360|$

Equation 3.1-12 $H2 = |(H + 60) - 360|$

Equation 3.1-13 $H3 = |(H + 90) - 360|$

Monochrome (Shades and Tints)

Shaders increase the saturation by 15%, and decrease the brightness by 15%

Tints increase the brightness by 15%, and decrease the saturation by 15%

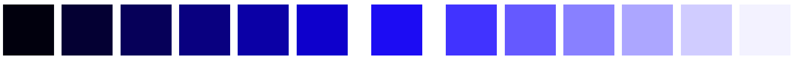


Figure 3.1-3 Brightness and saturation changes

Shaders will decrease the brightness by 15%, and Tints will increase the saturation

For shades, multiply each component by 1/4, 1/2, 3/4, etc., of its previous value. The smaller the factor, the darker the shade.

For tints, calculate (255 - previous value), multiply that by 1/4, 1/2, 3/4, etc. (the greater the factor, the lighter the tint), and add that to the previous value.

3.1.2 Impact of gender on icons

Icons considered to be one of the most important and common UI Element in mobile apps, and applications in general, some on these icons become very familiar to the public such as the save icon which is a floppy disk, even though the current generation did not use or even see the floppy disk, but they understand that the floppy icon mean save, or saving process.

But how far can icons can be understood by users, and what are the differences that will make a group of people understand an icon meaning over another group. We assumed that gender can be one of these differences, and some icons might be understandable, recognizable, or memorable than other for a specific gender (Chrysoula, Anastasios, & Dimitrois, 2012).

An experiment was made on a video game that contains some feminine icons, and masculine icons, and then each player asked to remember the icons were used in the game, the male users were able to remember the masculine icons over the feminine icons, and the female users were able to remember the feminine icons over masculine icons.

So we can judge that using masculine icons for male users will make the app more usable for them, but the only problem here is how to categorize the icons into feminine, and masculine categories.

Many icons classifications system are based on the graphic element, classify the icons into concrete, and abstract (Wang, Hung, & Liao, 2007)

Based on these classifications we can see the abstract icons as flat design icons, so we need to take color into account when designing an icon, on the other hand the concert icons which contains many details, and a mix of colors should also take this color mixture into account to present masculine and feminine icons.

Next figure shows the differences between abstract and detailed icons.

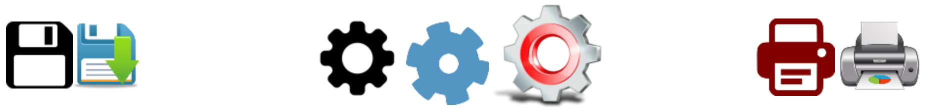


Figure 3.1-4 Abstract, flat, and detailed icons (save, settings, and print) (Flat Icons)

3.1.3 Impact of gender on visibility and shapes

We mean by visibility is the appearance or disappearance of an element that is a decoration element and has no relation to the functionality, i.e. that is not appear or disappear because of functional requirement. But it is an aesthetic element added to the interface.

We should know when to show the element, when to hide it, and when to replace it.

The aesthetic elements cannot be limited into a group because it is based on the designer imagination, and we cannot expect all the possible creations from them.

But we can make a guideline for basic shapes that can be used as for interface elements, and we can establish the relation between these shapes and gender preferences.

shapes or “views” could be differ from square, circle, or rounded square, as basic shapes for design in UX principles, and rounded-squares may appears more friendly to the user and smooth for the eye, on the other hand the sharp edges square appears to be harsh, and can be used for more attention (Lidwell, Holden, & Butler, 2003), Another opinion in the icons shapes is to make it with different outlines to be easily scan-able by the human eye (Icons - Style - Metrial Design), circles are excluded from this comparison because they waste too many space around the corner and that may lead to create more interfaces to hold all the required data to be presented.

Basically some shapes, can be defined as feminine shape, or masculine shape, since there is differences between male and female preferences, which men and women’s tastes may be polar opposites (Moss, 2003).

Shapes can add characteristics to views, such as the stability for squares, calmness for horizontal lines, the power for triangles, and harmony with circles (Carson, 2017), also it can express emotions to the human such as love, friendship, relationship, and community for circles, and ovals, balance, professionalism, strength, and efficiency for squares, and triangles, masculinity and aggressive for vertical lines, and femininity and calm from horizontal lines (Christie, 2017). These emotions and meanings for shapes can help the designers to define what icon could be masculine, or feminine.

Figure 3.1-5 shows an example of male, and female shapes as we assume, based on the previous definitions of shapes meanings and its relation to the male and female preferences.

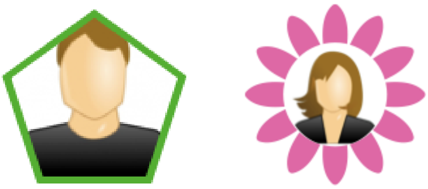
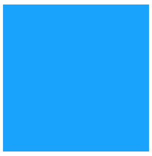
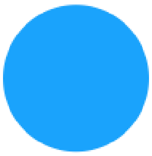
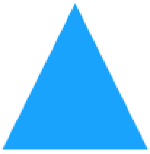
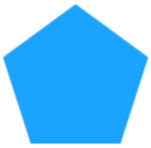
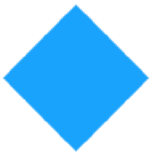
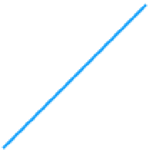


Figure 3.1-5 Male shapes vs. Female shapes

Basic Shapes

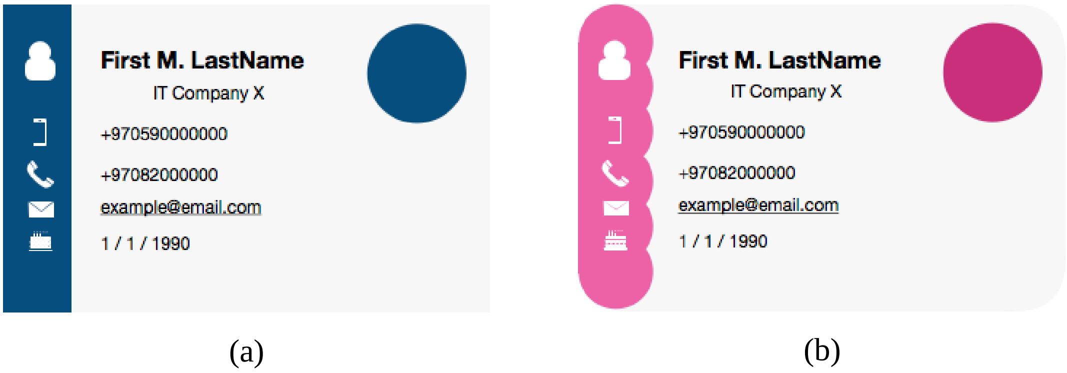
The designer can mix the basic shapes to create new composite shapes, and he can classify them into masculine or feminine based on the emotions that can express, also here the color plays a role of making the shape more feminine or masculine.

Table 3.1-3 Basic Shapes of views (based on Apple™ categorization in Pages™ Application)

Mixing Shapes with Colors

Table 3.1-3 present the basic shapes, and let's now try to compose them to create more meaningful shapes, at figure 3.1-6 (a) we use a square shape to hold the icons with a shade color of blue, and at figure 3.1-6 (b) we use a decoration circles on the icons holder to make it more calm and more lovely, we also use the tint color of pink, both color and shape together makes (a) more masculine and (b) more feminine.

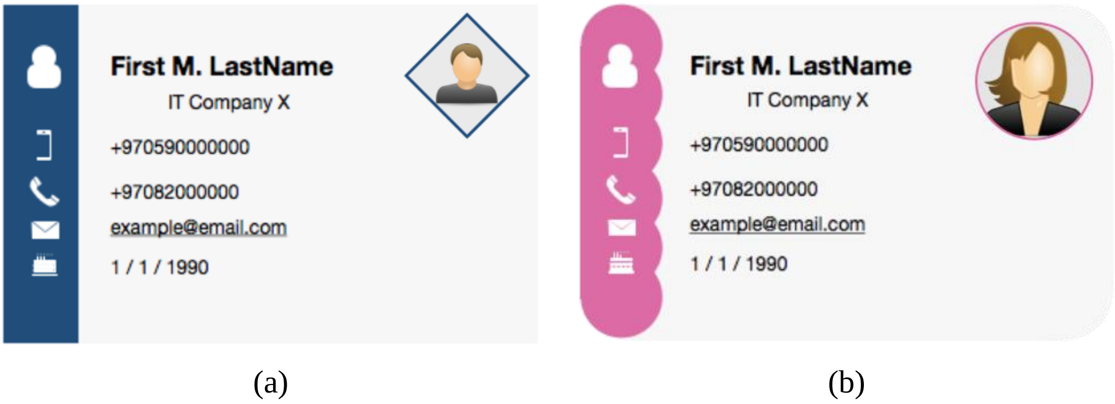


(a) (b)

Figure 3.1-6 User contact card design

When the designer decided to add decoration views, or images, he also should define if these are masculine or feminine additions, and he should define the opposite shapes to be appear, or just to disappear the elements. Our framework aims to smoothly remove the elements that should not appear, add the replacement elements, and check the corner radius for shapes. Also it will be able to decide if it need to change a shape or not even if the designer did not point to it, e.g. the circle of the profile picture in figure 3.1-6, should stay circle on both genders, or it should be replaced with another shape for each gender.

To achieve this decision, we need to take the usage of the element into account, since changing the decoration element will not affect the functionality, but changing a functional element's shape may affect its functionality.



(a) (b)

Figure 3.1-7 User contact card design (after circle shape adaptation)

It clear that circle shape is acceptable by both genders, and the diamond shape can also be acceptable, but the diamond here cuts more from the image than the circle did, in general we assume that diamond can be replacement for circle, but in case if did not affect the functionality or disturbed the design.

Shapes replacements

We need to define a global rule for basics shapes to be shifted from male to female design, in general squares and triangles are more masculine shapes, and we prefer to use them for male designs, circles as separate element can be used for both genders, but in combinations with other elements are feminine shapes.

So the Global rule that can be followed in shapes adaptation is the set the curvy views for feminine interfaces, and the shape edges shapes for masculine interfaces.

3.1.4 Impact of gender on position

The position of element and the easy access to a feature are very important in UX design, when the designer make a mistake in element position it could be annoying to the user, and may lead to not using the app or the game, also the steps needed to reach an element should be taken into account. But it is clear that the gender has no effect on the element position because the ability of seeking for an element in the screen can be different from a person to another regardless their gender, age, educational level, or any other user-factor, it is a psychological issue, and mental abilities.

So the gender has not effect on the element position, but the designer still need to find the appropriate position for an element for all users regardless then own abilities, or factors.

3.1.5 Impact of gender on types

Interface element has a limited group of types, and sometimes these types can be alternative somehow to each other, and here we want to check if it is possible for these elements to be affected by user gender, e.g. showing a pages instead of one long scrolling view based on user gender.

It is clear that gender has no effect on the selection of element type, it the same as the position it is not related to the gender so male, and female can both use all elements correctly.

Table 3.1-4 Summary of Gender-factor relation to interface elements

	Color	Visibility of shapes	Position (layout)	Types	Icons
Male	Shades Colors	Powerful, Sharpe edges, and strong lines	No Effect	No Effect	Masculine icons
Female	Tint Colors	Smooth, lovely, rounded corners, and clam			Feminine Icons

3.2 Relation between Interface elements, and Age

To define the relation between interface elements and age, first we should categorize users based on age to groups, the grouping of age usually based on the study domain, and there is different ways to identify these groups, first we can make a pre-study to find out each age preferences then make groups based on the results, but this required a full research on age first. Another way we can rely on a previous standard like the one from World Health Organization (WHO), but their groups don't satisfy our requirements, because the adaptation of elements will not be according to medical characteristics, so the best choice we found is to relay on the psychology of the user.

The psychology of user can be divided into 8 groups or stages, and these stages related to the human psychosocial. (Erikson, 1963)

The groups of Erikson's stages are (Erikson, 1963)**:**

1- Infancy: from age 0 to 1.5 years old, this stage of human psychology will be executed from our study because it not a usual user of mobile apps, and there are no targeted apps for this ages.

2- Early childhood: from age 2 to 4 years old, this stage also is not a usual user for mobile, and there are no targeted apps for them

3- Preschool: from age 4 to 5, the basic virtues of this stage is "purpose", the child at this age still learning and mastering the world around him, to know the purpose of things and the purpose for his existence.

4- School: from age 5 to 12, the basic virtues of this stage is “Competence”, the child at this stage become more aware of him/herself, so trying to avoid failure and faults, and starts to confirm his confidence.

5- Adolescence: from age 13 to 19, the basic virtues of this stage is “Fidelity”, at this stage of life the faithfulness increases in the user emotions, and become more trusting in himself, and confident.

6- Early Adulthood: from age 18 to 40, the basic virtues of this stage is “Love”, at this level the user seeking for love, and life, his emotions are heightened, and his life passion increases more.

7- Adulthood: from age 40 to 64, the basic virtues of this stage is “Care”, now the user start to care about everything around, he becomes thinking more reasonable, and start to prepare him to the be wise.

8- Maturity: from age 64 to death, the basic virtues of this stage is “Wisdom”, this is the last stage of human emotions changes, the user now has a lot of experiences and become more wise, and may not pay attention to the smallest, and trivial details.

3.2.1 Impact of age on colors

Age can be an effective factor on colors, since elder preferred colors can be different from young users by experiment. (Lee, Gong, & Leung, 2009). At each age level there is differences in colors preferences and it is hard to separate ages from genders (Miao, 2013), we can use experiments to establish a reliable relation between age and color, but experiments on group of users can be specific to their different circumstances or environmental factors or social culture (Sakamoto , 2014), and meanings of colors might be differs from one society to another (Haller, 2011), for example the Japanese considers the purple color represents wisdom, but Americans considers wisdom is represented by brown color, in the other hand Eastern Europe will use blue as a color of wisdom, so based on these differences mentioned by Haller, and we did not consider the culture in our study so we think that we can’t rely on these results experiments.

So we think we need to match between emotions with colors, and emotions with ages so by transient we can assume that there is a match between the colors, and the ages based on emotions, and we think by this approach we might solve the problem of different colors meanings in different society.

According to the previous study of ages, each group mixing a combination of emotions, we will match these emotions to the colors emotions guide (Simon, 2016) to make a recommendation for colors selection in our framework.

3.2.1.1 Understanding colors emotions

There are many studies about the emotions bring to human mood by colors, and basically they use these standard in production and marketing field.

So if you want to target a group of users at specific age stage you want to show them the colors that motivates the current emotion they have, and meet their desires. Such as targeting adulthood users which general emotion is care and responsibility, you need to use colors such as blue, turquoise, or green. Because these colors motivate the security, trust, loyalty, and protection, but for sure these colors are not the only colors expresses these feelings, beside it has a negative feeling can be brought to the user by these colors.

The next figure 3.2-1 shows each color’s positive, and negative emotion, and we will relay on this reference to match between color’s emotions, and age’s emotions.

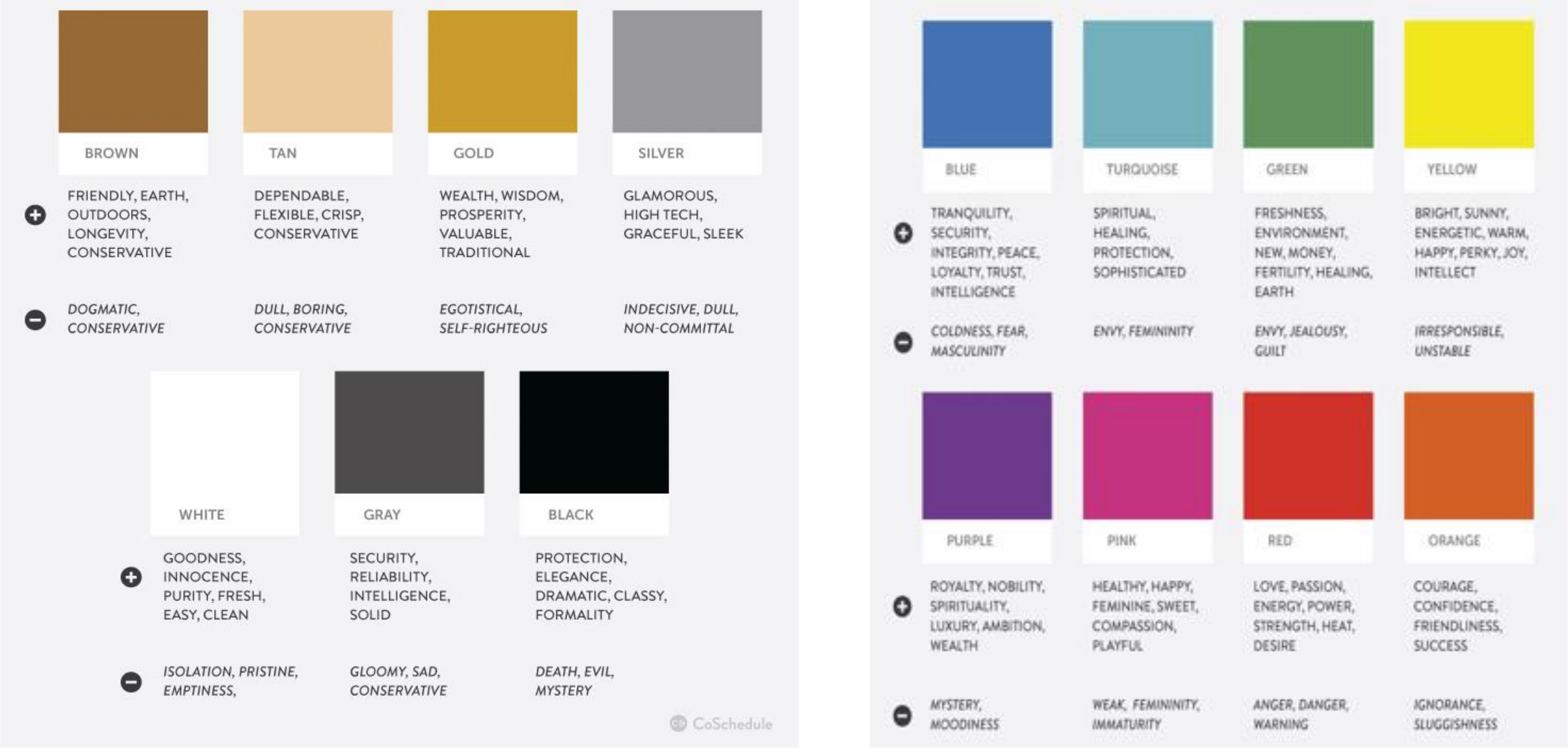

BROWN
TAN
GOLD
SILVER
FRIENDLY, EARTH, OUTDOORS, LONGEVITY, CONSERVATIVE
DEPENDABLE, FLEXIBLE, CRISP, CONSERVATIVE
WEALTH, WISDOM, PROSPERITY, VALUABLE, TRADITIONAL
GLAMOROUS, HIGH TECH, GRACEFUL, SLEEK
DOGMATIC, CONSERVATIVE
DULL, BORING, CONSERVATIVE
EGOTISTICAL, SELF-RIGHTEOUS
INDECISIVE, DULL, NON-COMMITTAL
WHITE
GRAY
BLACK
GOODNESS, INNOCENCE, PURITY, FRESH, EASY, CLEAN
SECURITY, RELIABILITY, INTELLIGENCE, SOLID
PROTECTION, ELEGANCE, DRAMATIC, CLASSY, FORMALITY
ISOLATION, PRISTINE, EMPTINESS,
GLOOMY, SAD, CONSERVATIVE
DEATH, EVIL, MYSTERY
CoSchedule
BLUE
TURQUOISE
GREEN
YELLOW
TRANQUILITY, SECURITY, INTEGRITY, PEACE, LOYALTY, TRUST, INTELLIGENCE
SPIRITUAL, HEALING, PROTECTION, SOPHISTICATED
FRESHNESS, ENVIRONMENT, NEW, MONEY, FERTILITY, HEALING, EARTH
BRIGHT, SUNNY, ENERGETIC, WARM, HAPPY, PERKY, JOY, INTELLECT
COLDNESS, FEAR, MASCULINITY
ENVY, FEMININITY
ENVY, JEALOUSY, GUILT
IRRESPONSIBLE, UNSTABLE
PURPLE
PINK
RED
ORANGE
ROYALTY, NOBILITY, SPIRITUALITY, LUXURY, AMBITION, WEALTH
HEALTHY, HAPPY, FEMININE, SWEET, COMPASSION, PLAYFUL
LOVE, PASSION, ENERGY, POWER, STRENGTH, HEAT, DESIRE
COURAGE, CONFIDENCE, FRIENDLINESS, SUCCESS
MYSTERY, MOODINESS
WEAK, FEMININITY, IMMATURITY
ANGER, DANGER, WARNING
IGNORANCE, SLUGGISHNESS

Figure 3.2-1 Emotions - Colors Guide (HAUFF, 2018)

3.2.1.2 Understanding Ages emotions

Erikson, classify ages to groups, and label each group with an emotion that is dominate the human at that age, the human will pass on several emotions in the one age, but there is a master or default emotion for the human at that age. So we want to map between the mater feel for a man with the appropriate colors collection possible for him.

1- **Preschool** - purpose: the appropriate colors for this attitude are: purple, pink, and turquoise
2- **School** - Competence: the appropriate colors for this attitude are: purple, and orange
3- **Adolescence** - Fidelity: the appropriate colors for this attitude are: purple, and orange,
4- **Early Adulthood**- Love: the appropriate colors for this attitude are: pink, red, orange and brown, since pink can happiness, sweet, compassion, red as express love, heart, desires, orange and brown can refer to friendliness.
5- **Adulthood** - Care: the appropriate colors for this attitude are: blue, turquoise, green, gray, and black
6- **Maturity** - Wisdom: the appropriate colors for this attitude are: white, gray, black, gold, brown, silver, and yellow

The previous groups show that one age can be mapped to many colors, and some colors can be used in different ages, this confusing may lead to instability of ages-colors mapping, for example the yellow color can express the brightness for Maturity group, joy and happy for preschool, and school ages, to solve this problem we need to exclude the mutual colors between ages such as the yellow, black, white, and orange. We will consider them as acceptable colors for these ages at all, and they will be changed to the nearest color for ages that does not contain these colors.

For example, if a designer did use the orange color for a view, the mature users may see it brown since orange and brown has the friendliness expression as in common, as shown in figure 3.2-2.

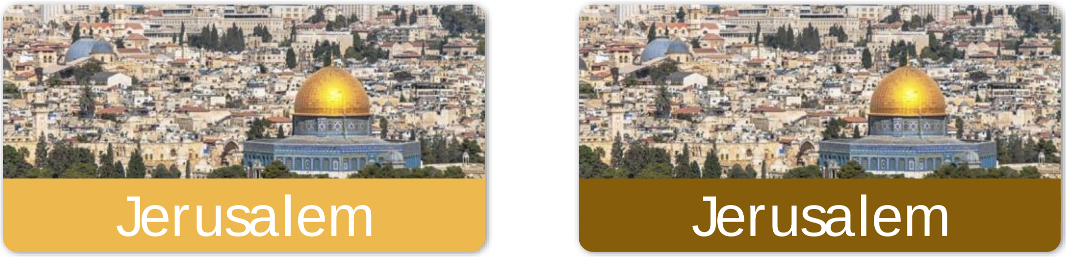


Figure 3.2-2 Color adaptation based on age

Figure 3.2-3, show the relation between Blue color with Turquoise, Black, Yellow, and White, there are relations between Turquoise with black, and blue, all these colors shares the expressions of security, the relation between blue and white expresses peace, and the relation between blue and yellow expresses intelligence, but there still some contradiction between these colors since, some of them express a negative emotion that contradicts with others emotions.

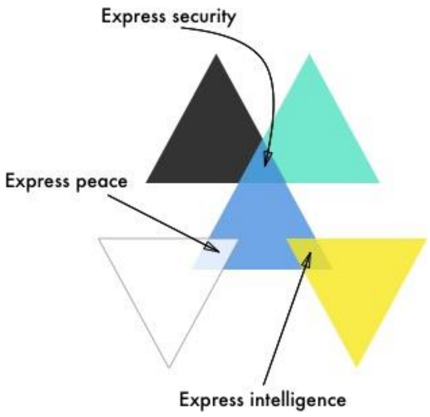


Figure 3.2-3 Colors emotions intersection

For example, Blue express trust as positive expression, and black express evil, trust and evil cannot be in the same side, also blue express coldness as negative expression, but yellow expresses warm as positive expression.

So we want to make an equation to identify the power between colors relation.

To identify the contradiction between colors expressions, one color should give a positive expression, and the other should give an inverse expression as negative, or positive, or one color gives an expression as positive, and the other gives the same expression as negative.

The relation between the two colors will be calculated using this equation.

Equation 3.2-1 $R = \sum \quad \text{mutual emotions} - \sum \quad \text{Contradicted emotions}$

R: Relation Power

Based on the previous analysis for the Blue color, we can see that there is a positive relation (mutual emotions) with the Black, White, and Yellow, but at the same time there is contradiction between Blue, with Black, and with Yellow so the summation of Blue-Black relation is Zero, and the summation of Blue-Yellow relation is Zero. The summation of Blue-White relation remains one, on the other hand the relation between Blue and Gray is valued by two since there is two mutual emotions, and the relation between Blue and Turquoise also valued by two, since there is contradiction on the negative emotion leads mutuality, so based on these values the Blue color can be shifted to Gray or Turquoise according to age factors.

Colors Ranges using HSB

The color range will be based on Hue value, since the other two values will identify the saturation, and brightness, they will not affect the value of the color.

Table 3.2-1, and 3.2-2 shows the ranges of each color in the hue spectrum, and we derived this range based on the basic HSV degrees 0/360 for Red, 60 for Yellow, 120 for Green, 180 for Cyan, 240 for Blue, 300 for pink. (learn., 2011)

Table 3.2-1 Colors' Hue Ranges

Color	Range	Color	Range	Color	Range
Red	0° - 30°	Green	70° - 150°	Purple	270° - 300°
Orange	30° - 60°	Turquoise	150° - 180°	Yellow	60° - 70°
Gold	40° - 50°	Blue	210° - 270°		
Pink	300° - 340°	Brown	20° - 40°		

The other colors can be generated using the saturation and brightness since they are shades, or tints for another previous colors.

Table 3.2-2 Special Colors HSB

Color	Hue	Saturation	Brightness
Black	Any	0%	≤ 25%
Sliver	Any	0%	98% ≥ B ≥ 95%
Gray	Any	0%	95% > B > 25%
Tan	34°	50% ≥ S ≥ 15%	88% ≥ B ≥ 65%
White	Any	0%	> 98%

Based on mapping the color to emotions, and the mapping between ages and emotions we need to map between colors and ages based on these emotions.

Table 3.2-3 Relation between ages' emotions and colors (HAUFF, 2018)

	emotions	Colors	Contradiction colors	Usable colors
1- Preschool	seeking for achievements as a life purpose	yellow, pink, , Purple, Red		Yellow Pink Purple Red
2- School	Seeking for achievements, and success over others.	Purple, Orange, Red	Orange: ignorance Red : Anger	Purple Orange Red
3- Adolescence	Fidelity	Purple, Blue, Orange, Gold	Orange: ignorance	Purple Blue Orange Gold
4- Early Adulthood	Love	Red, Pink, Orange, Brown, Blue	Orange: ignorance Brown : Dogmatic	Red Pink Orange Brown Blue
5- Adulthood	Care	Blue, Turquoise, green, gray, black, Tan	Black: Evil	Blue Turquoise Green Gray Black Tan
6- Maturity	Wisdom	White, gray, black, gold, brown, silver, yellow	yellow : irresponsible, joy, happy	White Gray Gold Brown Sliver Yellow

Table 3.2-3 at the last column – Usable Colors – shows the proposed colors to be used for each age and we call this set of colors by the attitude or emotion related to this age for example the set of {White, Gray, Gold, Brown, Sliver, and Yellow} witch contains the colors of the maturity age will be noted as wisdom colors since it is related to the wisdom attitude, you will find many colors are mutual between ages but as example of colors shifting, if the app contains a blue color, and the user is from class 1 (Preschool), the blue color is not an option for the preschool users, so the color should be shifted to one of the existed colors for this age (Yellow, Pink, Purple, Red), since there is a relation evaluated by one between blue and yellow the best choice here will be yellow for kids at this age. Another example could be choosing the yellow color

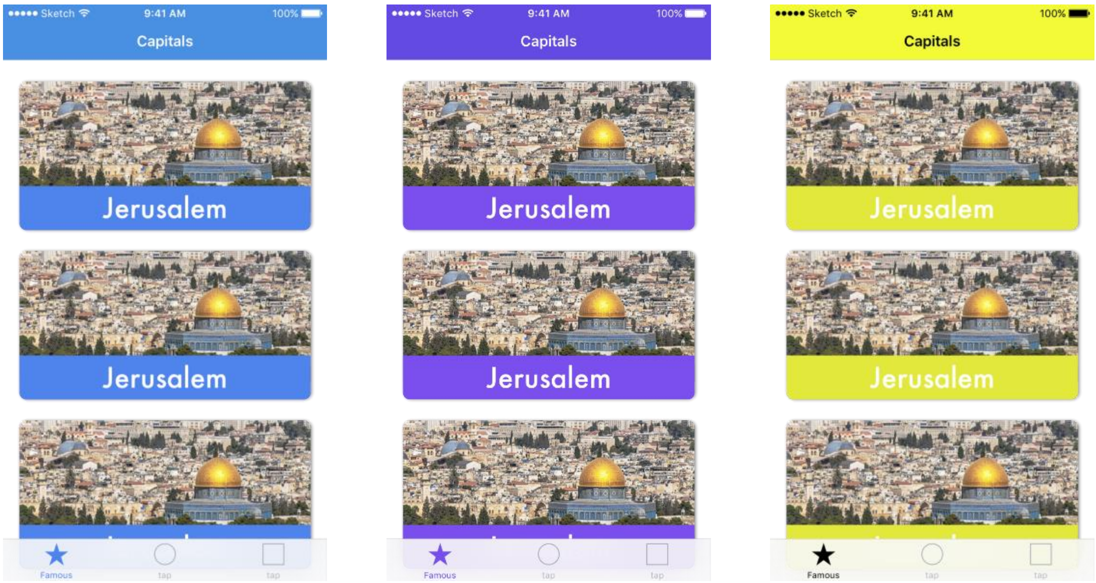
in the default design, and the user is an adulthood user, so we need to change the color to one of the adulthood group's colors (Blue, Turquoise, Green, Gray, Tan, Black), it is obvious that we shift blue to yellow in the previous example, so we will make the inverse now to shift the yellow to blue, but in fact we will first evaluate the relation between yellow and other colors to find out if there is a stronger relation than the one between yellow and blue, there is a relation between yellow and Gray it could be stronger than the relation between yellow and blue, since yellow express warm, and blue express coldness this contradiction will decrease the relation and make the gray over the blue in this shift. But after resolving the contradiction between gray and yellow it seems that yellow should be changed to Blue again.

Yellow - Blue Relation: Intelligence (+1), Warm vs. Cold (-1) => 0
Yellow - Gray Relation: Intelligence (+1), happy vs. Sad (-1), Reliability vs. Irresponsible (-1), Solid vs. Unstable (-1) => -2

Another perspective of colors hue changing could be the closest color to the original color, since sometime the color could be a part of the app or the logo identity, and it will not make sense to convert blue color to yellow in a logo that contains the blue as a part of the app owner identity, so we suggest to apply another formula based on hues differences between the two colors to find the closest color to the original color.

Let's take the blue and yellow again as example, the blue hue range between 210˚ and 270˚, consider our blue is 240˚, and the yellow color is between 60˚ to 70˚, consider our yellow is 65˚, shifting from 240˚ to 65˚ will be about moving 175˚, but if we check the emotions again for the preschool age we will find that purple color is the closest color to blue, and indeed there is common emotion between these two, and difference between blue and purple (285˚) is 45˚, by applying a threshold on the difference suppose (Θ = 75˚[1]), we will find that purple is more acceptable than yellow shift.

[1] The value of Θ derived from the average maximum difference between two neighbor colors (Blue and Turquoise)



(a) (b) (c)

Figure 3.2-4 Shifting blue to yellow and purple, (a) original design, (b) purple shift, (c) yellow shift

3.2.2 Impact of age on icons

Icons are symbolic representation for some text, or feature in the app, they used usually to save space, and as a decoration for the interface, but not all users can understand the meaning of the icons especially if it is not a common icon, also the icons takes the attention over text, so if the screen has too many icons, the user can't focus on the whole screen. About focus and concentration users abilities can be different but in general, too young, and too old users have a problem with focus and concentration (Pei-Luen & Jia-Wen, 2007), so we need to reduce the use of icons for elders, but for kids although they have problems with focus but the text problem can be greater for them, so kids need a meaningful icons that will not disturb user concentration, and elders may prefer text over icons (Al-Razgan, Al-Khalifa, Al-Shahrani, & AlAjmi, 2012).

Figure 3.2-6 shows two text fields one for email, and the other for password, containing an icon, with a placeholder.

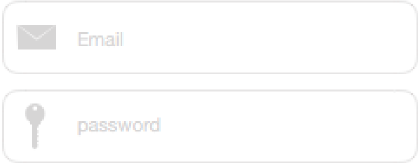


Figure 3.2-5 login form with icons and placeholder

In mobile screen sometimes you can't set the icon, and the text in at one place, so you might need to remove the text, or the icon to save space, here comes the age role, if the user is young or kid, the icon may be enough for him, but if he is an old man, he may not understand the icon, so we should present the text instead of the icon.

Another idea of icons adaptation based on user age can be the icon itself, some icons can be meaningful to use with kids, but it will look silly to young or elder user, and vice versa, will be meaningful for elder (Leung, McGrenere, & Graf, 2009), but kids will not like it or they may not understand its meaning, so designers need to create special icons for each age group, if possible.

To make icons suitable for some age you need to understand the shapes relation to the age which will be explained in the next section, because icons are collection of shapes grouped in order together.

3.2.3 Impact of age on visibility and shape

As previously mentioned that decoration and icons can disturb the attention of the elder users so making the app more official and formal could be better for elders, and to add more decorations for kids will grape their attention to the app, although they are not focusing of some element.

For example, if an app contains too many star shapes, and polygons, the app will be annoying to the old wise users, but in contrast it will be exciting to children, and teenage girls.

As we assume that children prefer more polygons, elders prefer less polygons and sharp edges, we present an example illustration in figure 3.2-6 presenting the differences between different ages acceptable design.

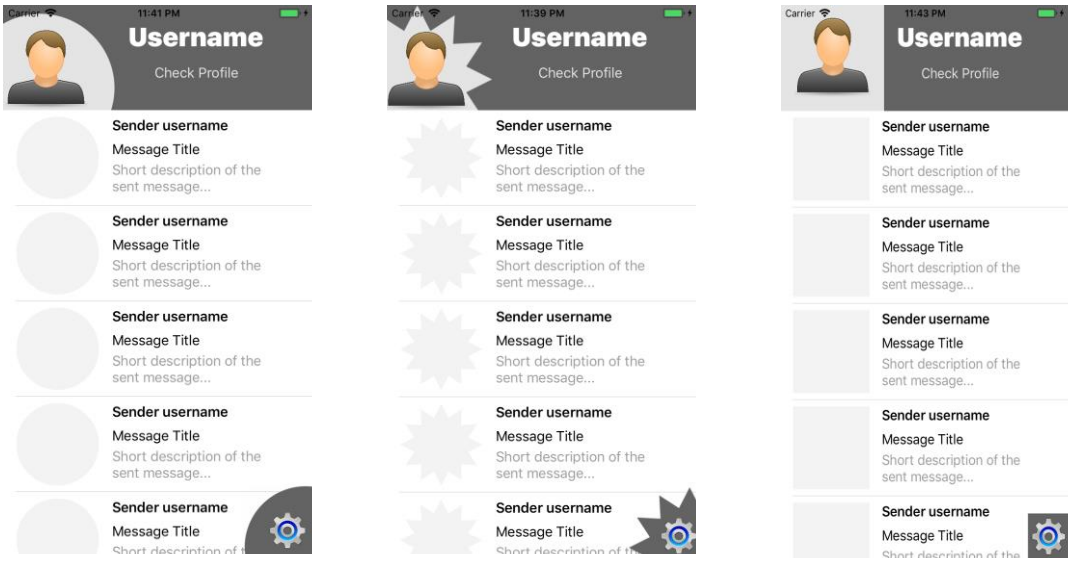


Figure 3.2-6 Different shapes for same interface, based on user age

3.2.4 Impact of age on position

The position of an interface item itself might not be affected independently by user age, but the screen as a whole (screen layout) can be affected by user age based on experimental results (Thinnukool & Kongchouy, 2015), Thinnukool categories the interfaces to two basic types, which are the static interface – that has no scrolling – and the dynamic interface – containing scrolling – and apply the on group of users with different ages, and genders, according to the results of Thinnukool experiment ages from 15 to 30 appreciate dynamic screens, but elders from age 30 to 50 appreciate static screens.

Based on Thinnukool experiment we suggest that developer need to allow the screen to be dynamic or static based on the age category.

Static interfaces will produce less flexibility, and visibility (Thinnukool & Kongchouy, 2015), so developer still need to use a few of dynamic content in some elements as shown in figure 3.2-7 (a) is a static interface and it shows only three icons from a list of icons, so developer might need to make icons getting smaller to show all possible items, or make this screen become dynamic, and this decision can be made based on the expected content

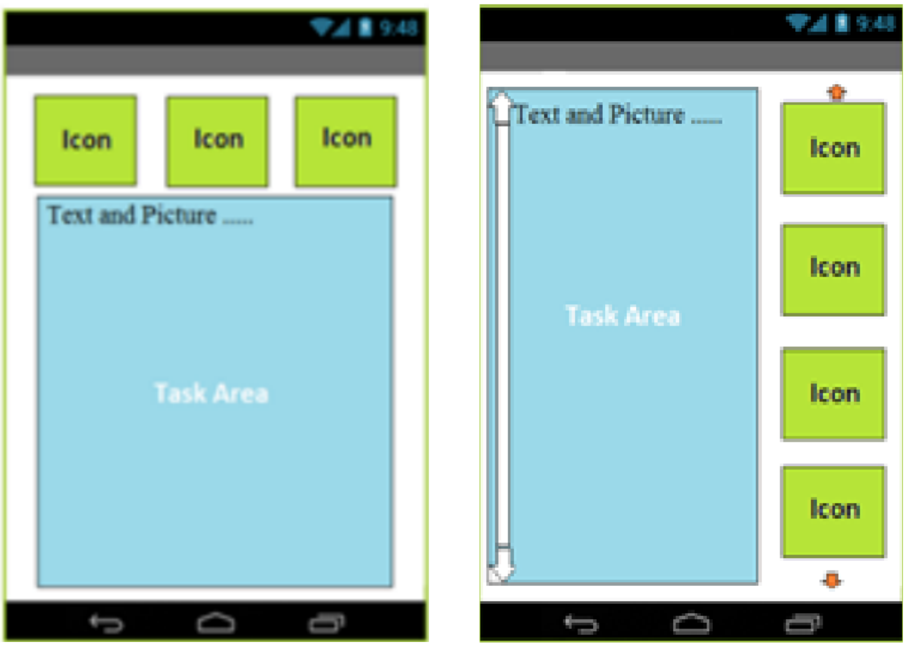


Figure 3.2-7 (a) Static Interface Design (b) Dynamic Interface Design (Thinnukool & Kongchouy, 2015)

3.2.5 Impact of age on types

The human age has no effect on the element type, since all users in all ages can use all element regardless their presentation mechanism. So we can consider that age will not be taken into account with the element type.

Table 3.2-4 Summary of Age-factor relation to interface elements

	Color	Visibility of shapes	Position	Types	Icons	
Age #1 (4-5)	Shinny Cheerful colors	Polygon shapes, and rounded shapes	Use Dynamic Screens	No effect	Mixing text with icons, or icons only	Define a set of icons for each age
Age #2 (5-12)						
Age #3 (13-19)						
Age #4 (20-39)	Vital colors	Sharp edges less lines, and rounded shapes				
Age #5 (40 - 64)	Care, and relax colors		User Static Screens			
Age #6 (64 - death)	Wisdom colors				Text rather icons	

3.3 Relation between Interface elements, and User skill / Experience

To understand the relation between user skills, and the interface elements, we need to define a few things, (**1)** First what events user made to be considered? (**2)** What actions should be taken based on these events? (**3)** On Which elements these actions should be applied? (**4)** And how these events and actions can categorize a person to be novice, medium or expert?

Users can make so many events while using an app starting from viewing a screen, touching a button, sliding a slider, scrolling view, or any other expected and unexpected events.
So based on this enormous number of events we want to classify these events first.

Users Events categorization

Users in general can make repeated events, from beginning to end, they might try something without finishing the task, they also may be fire an event for once, or not to fire it at all, and that is the cold start case.
Based on these expected behaviors we saw that event could be:
1- Repeated: The tasks, or event that the user usually does it, and always makes. Such as check his mail box every day twice, exploring the Sales sections, read details of news contains the tag ‘War’, or explore galleries related to Lacoste™ clothes.

2- Un-Finished: These events, or tasks, that never be completed, such as exploring the sales sections without read details of any product, or filling a form without submit.

3- Single: The tasks, or event that is fired once, when the user makes any compete action it registered as a single event, but when he repeats it again, and again, it become a repeated event.

4- Unknown: The system should previously know all features and service it provides to the user, when the use never accesses, or use a feature it will be unknown feature to him, so basically all events derive to achieve some task will be unknown till the user use it at least for one time

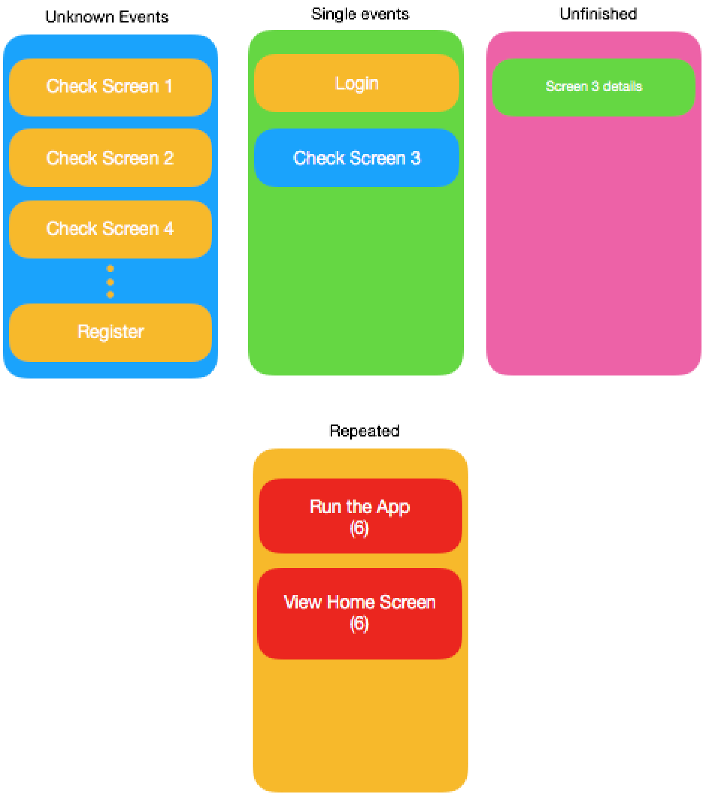


Figure 3.3-1 event categorization for app state

Figure 3.3-1, shows an example of an app instance state, which appear to be not used much, because the user did access the app several times, and keep it on the home screen, the user has a previous account since he did login without register, and he check on screen 3 once, without seeing any details of it.

This is how the knowledge about user skill and behavior should be translated.

Now the second question which is about the actions should be taken, for each group of event there should be actions to take, these actions related to the interface elements appearance or style, for example, the unknown events should notify the user about their existence in somehow, the repeated events should have shortcuts so the user need not to repeat the steps to achieve same task.

According to that analysis we need to classify the actions, and it will be as following:

1- Present and Hide : some elements might be ignored and hide from the screen because the user is not interested on them, and others should be presented clearly in some different shape or positions to notify the user of their existence, the action decision should be taken by the system based on the unknown event group, i.e. if the user did not use a feature ever, the app should take a decision after analyzing user behavior if the user not interested on this feature so it should be hidden somewhere, or the user needs this feature but did not know how to find it so the system need to clarify it for him.

2- Decrease / Increase appearance priority: according to single use event, the user may not need this feature anymore, or not interested on it, so it should decrease its appearance priority this means to make it down in the list, or the user use it for once, but there is a possibility to use it again, so it should be increased up to the list, so he can remember it and use it again.

3- Store previous steps: if the user did not finish a task the app should decide if the user is going back to achieve this task later, so it should keep his steps, and may be achieved some of them, or just forget it because the user not interested on this feature.

4- Default Resetting: some elements and screens meant to be the be defaults for the app, and for all users, and some apps allow the user to customize these defaults based on his interests, but here the app should reset the defaults for elements and screens based on user's behavior analysis.

Some of the previous actions could be functional actions, not interface adaptive, but in fact it could be both, for example the storing steps actions, could be functional if we say the user fill a form without submit, and when he back to this form, all his data is filled automatically this is a functional requirement, but as adaptive interface action, the app will present a not filled form, and showing a new button with 'Auto Fill' title, that will fill the form with the appropriate data.

The Third question was about elements, especially User Interface (UI) Elements, the most effective elements on the screen can be, Views, Labels, Text-Fields, Image-Views, Buttons in all buttons' cases, and Lists (scrolling screens), also we can consider the containing interface (e.g. UIViewController in iOS, and Activity in Android) as a UI Element.

But still there are many not basic elements can be found in a screen, such as progress bars, alerts, sliders, text views, maps, web views, ...etc.

Each of the previous actions we mentioned, has special meaning on each separate element, for example the present/hide action with Image-View, could means to show the image view, or hide it, or present a slider instead of image view, but in the View it could mean show elements inside the View such as extra buttons or labels, or hide essential elements from a view. It is worth to mention the navigations as adaptable elements two, such as the paging, navigation controllers, and tab bar. The animation at one element, or the transition between interfaces it also could be adaptable. And that will be our studies in the next sections.

The last question will be about categorizing a user’s into three main general known categories which are, the novice user, medium user, and expert user, to judge a user to be novice or expert or whatever, we cannot make a general judgment for the user, for example any computer user can be expert in gaming, but novice in searching on World Wide Web (WWW), and there is many researches were made about each field independently, based on that fact, we want to make judgment on the user to be in a group of actions.

To make that classification for the user we need to evaluate the user using one of the evaluation methods used in human computer interaction field, the evolution should test the usability, functionality, and stages of a design cycle, and by studying the existing methods of evaluation we see that Automated Analysis Evaluation (EVA) (Human-Computer Interaction: Design and Evaluation, 2015), it will be suitable to evaluate the user interaction with the design, since it based on the act and react of the user, we intend to group the actions so we can evaluate it as one feature, for example the user may need to use feature x, so he need to do several actions to finally use this feature, so the action evaluation will be on features not on actions, after we evaluate the time, and effort that a user may need to execute a feature, our framework tills that developer should find the best interface adaptation that can reduce this efforts and time.

3.3.1 Impact of skill on colors

Each interface element has different colors divided into three main types which are, the background, the foreground, and tint color. Some of the elements uses the three types, and some of them uses a group of these types, for example, image view, uses the tint and background colors only, but text fields use the three types. What could be the reflected effect applied on one of element colors based on action taken according to a user event?

To answer this complicated question, we need to establish a relation between colors, and skill, but apparently the relation between color and skill does not exists, since novice users, and experts will accept the app with its colors. And it is not correct to say blue color is suitable for novice but black is suitable for experts, and orange for medium users, such relation does not exist, and not proofed, so we will consider that user skill does not affect elements colors.

3.3.2 Impact of skill on icons

Icons can have different designs as we previously explained, icon could be detailed, flat, abstract, the level of abstraction may allow some users to understand the mean of icon, and others may not understand it, so we can consider that the level of abstraction can be matched with the skills and experience of the user.

If we take the history of icons during the computer development, we can see the development of icons, this development happens because current and new computer users become familiar with the icons and easy to understand the meaning of these icons

Figure 3.3-2 shows the development of my computer icon in windows 95, XP, 7, and 10, you can see that the level of details is reduced, in windows 95 there was a content in the screen, drive A, power light, and keyboard to tell the user that this icons means your computer, but later in XP, the keyboard disappeared, and content of the screen two, but we still see the computer box, with driver and power light, later in windows 7, the light disappeared, and the computer box is a little tiny item next to the screen, finally at windows 10, there is no existence for the box, the user now understand that this screen means your computer.

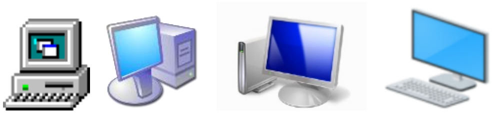


Figure 3.3-2 My-Computer icon development (captured from Microsoft Windows OS)

We can take advantage of these changes to reflect it on the adaptable interface for icons, according to the user skill, when the user is novice user we can show him a detailed icon for uncommon features, later the icons will change to level two which is less detailed, and finally the abstract icons can be used for the expert users.

If we consider the detailed icons to abstractions as a development of icons lifetime to be reflected on the development of user skills while using the app, we may face some cons, such as the user may become familiar with the detailed icons but lately when the icons changed to more abstract icon this may lead to confusing and become annoying feature rather than useful feature.

So we basically made these rules to protect the user from icons changes confusing, the icon design should contain a core symbols that should not changes at all, it can change colors or style but it should never have removed from the icon, medium and expert icons should not contain text or numbers, every meaning should be represented in symbols, symbols in medium icons can contains group of single symbol but it should be single in the expert icons.
As Example, here is a core symbol for “Transaction” icon

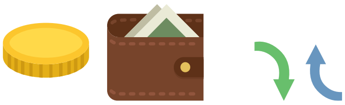


Figure 3.3-3 Core Elements that could be used in Transaction icon

Base on elements in figure 3.3-3, we want to make icons for the three skill levels, as shown in figure 3.3-4, to make an icon for novice we should add many details to the icon, you can see the dollar sign on the coins, the arrows are separated from other component so their direction will be noticed, and there is a content in the wallet, but for medium icon we did remove dollar sign and reduce the number of coins, and more interleaving between the arrows and other components, finally the expert user icon we make it as simple as possible, so one coin showed to present a group of coins, wallet is more abstract now.

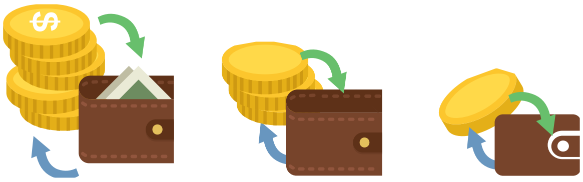


Figure 3.3-4 Transaction icon for different user skills (novice, medium, expert)

3.3.3 Impact of skills on visibility and shapes

It is clear that the visibility of decoration shapes and types of them is not a skill matter since the development of user skills will not change a circle to square or vice versa, only visibility of items can be related to the events mentioned before, the user can see items clearly, or not seeing an items due to his usage and events on the app.

The user events can affect the visibility of items, we declared at the beginning a group of actions related to these events, such as the present or absence of an element, or priority changes actions, based on the usually events, and the singularity events or other events. And the app should make appropriate decisions about the reaction should be taken.

To define the reaction should be taken by the app based on the user events, there is relations and dependencies should be defined by the developer, such as an element appearance based on group of element usage, but not in a functional way, i.e. the usage of these elements may lead to a clue that the use may be interested on such element, but there is no functional relation between them. Anyway the functional dependencies and non-functional dependencies remains on the developer side.

The relations between elements can be defined in two ways, first elements usage may lead to other elements usage, and second, elements usage may lead to not use other elements.

Also we define a threshold on the repeating of element usage, and rules can be set to define these relations.

For example, firing an event_1 more than 3 time, and event_2 more than 6 times this leads to bring element x to the top of a list.

$$Ev_{t=3}(1)\ AND\ Ev_{t=6}(2) \rightarrow Action\ (20)$$

So developer need to define the set of rules for elements relations, dependencies, and actions, in his system, based on the functional, non-functional, and aesthetic requirements of the app.

3.3.4 Impact of skill on position

Position of the item is a big considerable issue in the skill relation since it is the result of user events. Position of item should not be based on functional issues such as to show novice user’s tools less than expert user due to role or privilege, for example a accounting system may allow experts to achieve complex tasks but novice can achieve only simple tasks, but we are talking about interface changes due to user events.

The same principle used for the visibility of an element, could be used for the positions, so the developer should define these relations to define what reactions should be taken based on user events.

But the difference we face here, is the position may be strongly related to the usually events, more than other event, although there still relation between the element position and other events such as the singularity, un-finished, and unknown events.

Defining relations between elements

We will define some elements relations that might be used by the developer for repositioning based on user actions.

1- User may continually keep selecting an element, or navigating to a screen as a first step when run the app, this will change the defaults of the app, so the usual selected screen should become the first screen always, and the selected element, should be automatically pressed by the app itself each time.

2- User may continually keep selecting an element, or navigating to a screen but not as a first action or step, this should change the position of the elements to be the first visible element to the user, i.e. if the element is ordered in a list, these usual elements should be moved to the top of the list.

3- User may never select an element, but this element has relation to another one, that the user uses it in different frequencies

a. Always case (user always use the related element based on a threshold), in this case the app should predict that the user may need to use the original element since he uses the related element too much.

b. Sometimes case (user element usage is less than the always threshold), in this case the app should try to find more powerful evidence to ensure that the user may need the original element, such as seeking for another related element that the user uses much.

c. Rare case (user uses the element less than sometimes threshold), this case means the user may not be interested in this feature since he is not interested in the related features.

3.3.5 Impact of skill on types

User skills can affect the type of elements to be used since there is some elements can be used by experts, and others are for novice users, for example an element may ask the expert user to enter text, and allow the novice to select a preset text, or element might show a form in steps for novice, but expert may enter all data in the form in one shot.

These possibilities are not easy to judge, because each element has pros and cons, and it can't be generalized for experts, or novice.

So we want to find out what element types can be classified as expert elements, mediums elements, or novice elements.

First of all, we want to study each UI element in the mobile system, specially the iOS system.

Interface Essentials

Bars (Navigation Bar, Search Bar, Status Bar, Tab Bar, and Toolbar): The bars are always some essential interface elements that can't be altered to another element, but instead if there is a group of elements inside these bars, these elements can be reordered, or change visibility options based on previous section. (Apple™, 2018)

Views, and Controls: are the playground we meant to study, since some of these elements can be alternative to each other's

Collections, and table views can sometimes consider to be alternative for each other, based on the user preferences, usually a customization button added in the interface to switch between collection and table presentation

Image View, can be matched with sliders in case of groups of images for the same screen

Controls

Pickers: are used to pick items from it, but there is many form of items selections list, such as dropdown lists, and full screen table view controller, we can alter between these elements based on user preferences and user events.

Refresh Content Control (RCC): to reload a table view we can simply pull the table from to top to refresh content, but what if the user did not know this action, so we can alter this element to another element that can has the same behavior such as adding a button on the top of table that used to refresh the content.

Segmented Controller: used to select between two to five element, or more, the pickers also are used for the same purpose so these elements may be alternative to each other's.

Switches: used to turn on, or off a feature, i.e. to choose between two elements, and this may lead to the same point we can alter it to segmented controller, or picker.

Based on the interface element essentials, we will assume that there is a group of items that can be switched from one type to another reflecting the user skill, we will try to find out the matching between skills and interface elements according to the abilities needed from the user to deal with an element such as the easy items to use that do not any previous experience from the user can be presented to novice users (Akiki, Bandara, & Yu, 2014).

Table 3.3-1 Elements' types changes

Element	Novice User	Medium User	Expert User
Bar Content	Show all buttons	Show all buttons	Group buttons and save space
Tables / Collections	Keep it as it is	Keep it as it is	Keep it as it is
Image View	Keep it image view as it is	Can set slider for images	Can set slider for images
Picker	User picker or drop-down list always	User picker or segmented in less than five options	Use picker and allows the user to enter text
RCC	Use visible button instead	Show hit and do not use buttons	Use RCC as it is.
Segmented Controller	User picker or drop-down list always	Use it if the options are less than five	Use picker, with the ability to write text
Switch	Do not use picker or segmented instead	Use switch as it is	Use switch as it is

3.3.6 User skill detection

To monitor user activities while working on an app there is many platforms support this idea, such as Google Analytics™, Flurry™, MixPanel™, Appsee™, each of which, has its own pros, and cons, but the main problem in all these tools is that all are web based, they can’t be used to store user events in a local storage to be used later by the app itself.

So basically we will not discuss the method, or the tool that can monitor the user actions, and extract them out, or the storage method, we are concerned with the use of this information so next we will clarify what information needed to be stored in the local storage.

The main action that is generated by the user as interaction with the screen is to touch the screen, and then the user may release his touch, or drag, these two actions called, Touch Up, and Touch Drag, and they can be generated by several gestures, such as tap, pinch, pan, and swipe.

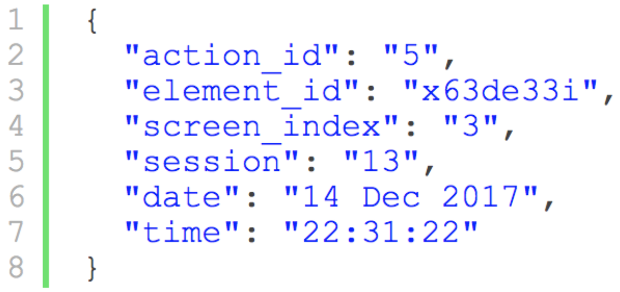
These actions can affect several interface elements, mainly the buttons, and the Scroll Controller which sometimes are included in a Table View, or Collection View.

We suggest that developer need to create a component in his app, that keep listening to the interactions, and touches, then it record these user actions in JSON format to be used later in decision making.
For example, when the touches an “Add to cart” button for some product in the sales section, the sequence of action will be stored as the following.

Actions: Tap Bar Item <Sales> ⇒ Tap Cell <IndexPath(0,3)> ⇒ Tap-Button <add to cart>

Time: 1 sec <Home Screen> ⇒ 3 sec <Sales Screen> ⇒ 22 sec <Details Screen>

JSON Format for an action



```
{
  "action_id": "5",
  "element_id": "x63de33i",
  "screen_index": "3",
  "session": "13",
  "date": "14 Dec 2017",
  "time": "22:31:22"
}
```

Figure 3.3-5 JSON format for an action

action_id: refers to the type of user action, such as tapping, dragging, or releasing ...etc.

element_id: is a unique id that is assigned from the developer to the adaptable interface element.

screen_index: the developer should assign an index for each screen

session: number of seconds that the user take on the screen before perform this action

date, and time: are the actual date and time when the action performed

The framework should get all actions related to an element and check what case applied to this element (usual, single, never, unknown), then it should apply the appropriate effect on the element based on the elements relations and dependences.

Events Detection Example

Consider an app always starts on the “recent” interface, showing a list of products, the user every time runs the app, he taps the “Featured” interface and move to the featured products’ list as shown in figure 3.3-7 (a)

This will record events as the following sequence

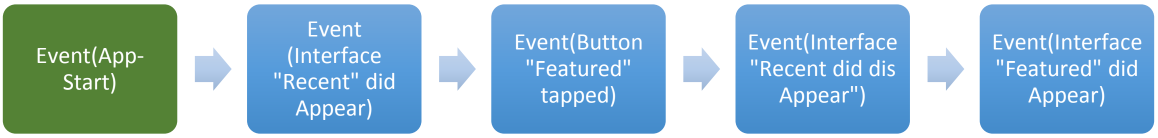

Event(App-Start)
Event (Interface "Recent" did Appear)
Event(Button "Featured" tapped)
Event(Interface "Recent did dis Appear")
Event(Interface "Featured" did Appear)

Figure 3.3-6 Events Recorder sequence

Each time the app runs it will check the reordered event and check if there is some events are equaled, the event are equaled if they have the same element, the same interface, and the same action type, but we should take in consideration that some events are not effective events, such as the app-start event is not an effective event, since it has no reflected action to the adaptability process, also the first interface appears is not effective because it is the default screen, but appearing of the "featured" interface is effective event.

We can say that the effective event is the one after button tapped, or to be more precise the appearance of interface that is after disappearance of another interface.

From figure 3.3-6 we notice that we need to record the sequence of events not just the events themselves to define the effective events, also here we can consider the sequence of event is another factor to the similarity of events.

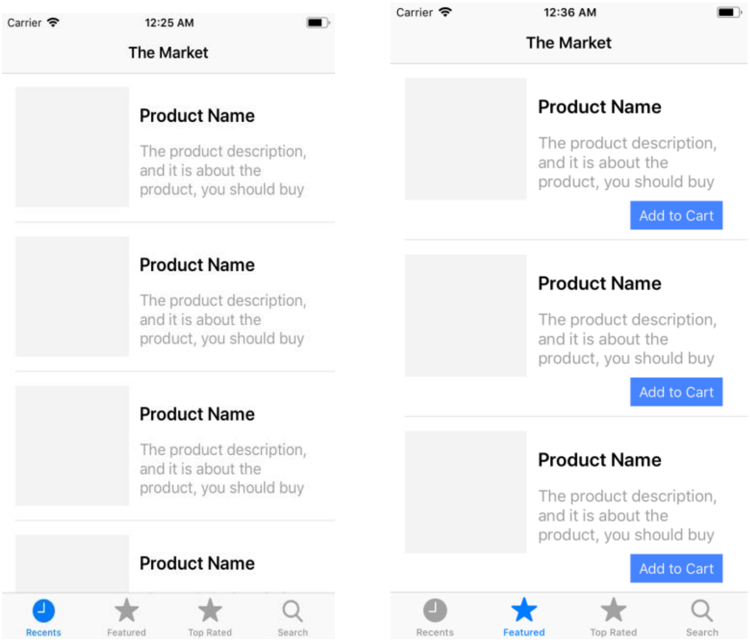

Carrier
12:25 AM
The Market
Product Name
The product description, and it is about the product, you should buy
Product Name
The product description, and it is about the product, you should buy
Product Name
The product description, and it is about the product, you should buy
Product Name
Recents
Featured
Top Rated
Search
Carrier
12:36 AM
The Market
Product Name
The product description, and it is about the product, you should buy
Add to Cart
Product Name
The product description, and it is about the product, you should buy
Add to Cart
Product Name
The product description, and it is about the product, you should buy
Add to Cart
Recents
Featured
Top Rated
Search

Figure 3.3-7 (a) App Main Screen, (b) Adjusted Feature interface

To decide that the event now is usual event and we need to take action based on this action, we should define a threshold for the usually events, it could be combination of repeating action, and time period, or just of these factor.

In our previous example image that the user did repeat this sequence more than the threshold value, the framework takes the decision to set the "featured" interface to be the first interface presented.

Adaptation may not stop on changing default setting, let's consider the example, to check out another type of possible adaptations.

If the user usually touches the button "Add to cart" inside the detailed screen of products in the featured screen interface, the suitable action to be applied at this time is, to bring the "Add to cart" button to be presented on the list, so the user can clicks add to cart before even see the detailed screen as show in figure 3.3-7 (b).

Table 3.3-2 Summary of User Skills - factor relation to interface elements

	Color	Visibility of shapes	Position (layout)	Types	Icons
Skill #1 Novice	No Effect	Based on usage patterns	Based on usage patterns	Clear, and need no remembering (selections), stepping	Detailed
Skill #2 Medium					Less Details
Skill #3 Expert				Need options remember, no need for steps	Abstract

3.4 Conflicting Effects of User Factors on Interface Elements

Sometimes the three user factors (gender, age, skill) may be crossed in an element's attribute, the effects on an attribute might contradict, or they may be having the same effect, so we need to identify the contradiction between effects based on these three user factors.

The Color attribute has been identified previously, there is two effects might be applied on a color, the first one based on gender and it manipulate the saturation and brightness of the color, and the other change based on age that manipulate the hue value of the color, so the effect of gender and age on color attribute has no contradiction.

The Visibility of shapes affected by two factors which are gender and age, the shape can be powerful for male, smooth and lovely for female, also it can be rounded corners for ages less than twenty years old, the females less than twenty years old will have no problems with the shapes corners or style, since they prefer the rounded corners over the sharp edges, but they will have a contradiction at ages greater than twenty, also males less than twenty years old will have a contradiction on shape's decoration, to solve this contradiction we need to set a priority value for the user factor, the priority used to set a factor effect over the other one.

The element position can be affected by two factors, the age, and the skill, the user age affects the element position in indirect way, and at this point we can leave this to the developer to define which age is suitable of an element using the priority of age constraint on the element position, the skill can affect the element position though the usage criteria as we discussed before, and here the developer also can solve this contradiction using the constraint priority.

Finally, the Icons style, we have the three factors affects the icons style. Gender will affect the icons style by setting masculine and feminine icons. Age can affect the icons by using icons, or not using icons, or use both icon and text, and Skill can affect the icons by abstracting the icon though three levels of experience, if we want to marge these conditions together. We need to create group of icons for each adaptable element, and define each icon for which age, gender, and skill, this matrix of icons, will generate 54 icons for each element as maximum.

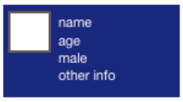
3.5 Framework Summary

Based on the previous sections be conclude that there is a relation between gender factor with colors, shapes, and icons, a relation between age factor with colors, shapes, elements' positions, and icons, and a relation between user skills factor with elements' positions, types, and icons as shown in table 3.5-1, and these relations can be summarized as shown in table 3.5-2

Table 3.5-1 Mapping between user-factors and adaptable element preferences

	Color	Visibility of shapes	Position (layout)	Types	Icons
User Gender	✔	✔			✔
User Age	✔	✔			✔
User Skill			✔	✔	✔

Table 03-5-2 Framework Summary

Factor	Value	Color	Shapes	Icons
User Gender	Male	Hue range (200˚ - 260˚) High saturation and low brightness 🟦 🟩 🟨	Straight lines, sharp edges, and bold borders 	Icons symbolize items from male perspective and environment. e.g. relax icon
	Female	Hue range (261˚ - 340˚) Low saturation and high brightness 🟪 🟩 🟨	Curves, and rounded corners and edges, smooth lines, and thin borders name age female other info	Icons from female environment, and symbolize items from female perspective. e.g. relax icon

Factor	Value	Color	Shapes	Position	Icons
User Age	Preschool (4-5)	Yellow, Pink, Purple, and Red	Much polylines, smooth edges and curves, and rounded shapes	Use Dynamic interface, that can position elements in a scrolling views	Mixing text with icons, using symbolic and semantic icons, and using abstract or voided icons
	School (5-12)	Purple, Orange, and Red			
	Adolescenc e (5-19)	Purple, Blue, Orange, and Gold			
	Early Adulthood (20-39)	Red, Pink, Orange,	Less lines, sharp edges, also rounded shapes can be used		

		Brown, and Blue			
	Adulthood (40-64)	Blue, Turquoise, Green, Gray, Black, and Tan		Use static interfaces that can show all possible options and items to the user so he don’t need to discover hidden options in a scroll	
	Maturity (64-*)	White, Gray, Gold, Brown, Sliver, and Yellow			Preferred to use text rather than icons. Icons should be more self-descriptive, containing metaphors for environment objects, and more details rather than abstraction.

Factor	Value [2]	Position	Type	Icons	
User Skill	Novice	Items can be repositioned based on its important to the user skills, and the user skills can	Use steps or paging, rather than scrolling, use radio buttons, rather than dropdown list (items needs no remembering abilities).	Use more detailed icons, that is self-descriptive.	**All icons should be derived from one**

[2] Novice, medium and expert users can be categorized based on usage pattern as we suggest, and according to number of features and actions the user made though the application.

		be defined by though the usage pattern. Developer can set constraints on each element to defined the position and the match skill.	Generally using items that shows all possible options the the user and make less confusion.	And Mixed with text labels	
	Medium			Use less detailed icons. You can get rid of text labels	
	Expert		Allow user to enter values manually rather that selection from a list or radio buttons. Paging, tutorial, and hints are not needed.	Icons can be abstract, with no descriptive text.	

Chapter 4
Implementation

Chapter 4

Implementation

We will start this chapter with defining the iOS app features, and then we will explain the structure and hierarchy of this app, later we will go deep in each component, and integrate the framework with it.

Our application will be a market app, with a main purpose showing products to the users, and each user has credits in the app, and a cart, the user can explore several types of products' lists, and in many ways, such as products by categories, by maximum price, or featured products, the user can explore the sales, and offers on products, he also can interact with the product board, so he can rate the product, add it to favorite list, or wish list, also the user can write a review on a product.

Test-bed App Features

The app should be able to get user gender and age, we will make it in the easy way which is via Facebook account, but the user still has the ability to register using another source of information such as a direct text field.

The app will automatically monitor user interactions with interface elements, for each element that the developer declares it as adaptable element, so it will store a map of user events with elements matching the suitable adaptable action should be taken for each case.

The implement app is a complete example of our theoretical framework that is discussed in the previous chapter, so developers can implement their own methods to apply adaptation on element based on what proposed in chapter three.

4.1 Test-bed app architecture

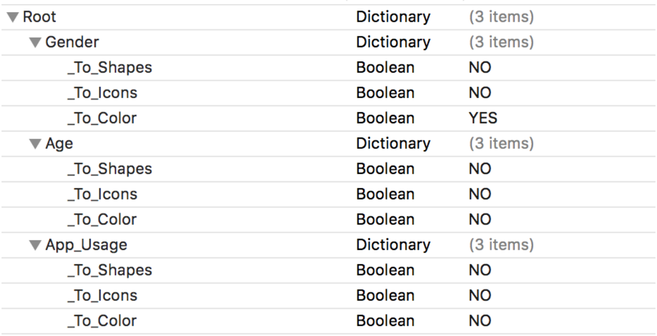
The basic element of adaptation implementation is the constraint which is called "AUIConstraint", the constraint has many properties to identify its role, it has the type which means is it a gender constraint, age, or skills, also it has the interface element that will be affected by this constraint, and it uses the superclass of all interface elements in iOS platform which is the "UIResponser", and a variable for the effect to be applied (AUIEffect).

The AUIEffect class is a super class for all possible effects that might applied to an element attributes, such as the color manipulation, shape, icon, position, and visibility.

Some of these constraint are global constraints which means that it will applied to all element on the interface unless the developer set the element as fixed element, “Fixed” option added to all interface element with a false value as default, the developer can set it to true so the global effect will not affect this element, global effects are defined to maintain the consistency and harmony between interface elements, for example the elements color, we can’t allow the app to adjust color for some elements rather than other, this might lead to mass in the construction of colors, and may produce new colors for some elements, so we might have more than three colors in one interface and this is not accepted in mobile interface design principles.

The developer can switch on and off the global constraint as he want, we add an “. plist” file to the app, containing these global constraints, which are the relations gender-color, gender-icons, gender-shapes, age-color, age-icons, age-shapes, skill-color, skill-icons, and skill-shapes, each of which is a Boolean type.

When the developer switches a global constraint on, a specific class we called the AUIEvaluator will start creating AUIConstraint for this relation without defining the UIResponder for this constraint, and when the interface (UIViewController) is loaded to the memory, it will clone all the global constraint for all elements contained in its view, then it will apply the effect for each element.



▼ Root	Dictionary	(3 items)
▼ Gender	Dictionary	(3 items)
_To_Shapes	Boolean	NO
_To_Icons	Boolean	NO
_To_Color	Boolean	YES
▼ Age	Dictionary	(3 items)
_To_Shapes	Boolean	NO
_To_Icons	Boolean	NO
_To_Color	Boolean	NO
▼ App_Usage	Dictionary	(3 items)
_To_Shapes	Boolean	NO
_To_Icons	Boolean	NO
_To_Color	Boolean	NO

Figure 4.1-1 Global Constrain plist file

The developer still has the ability to add constraints manually, by creating an object from AUIConstraint, with type, and effect, then bind it to any interface element he want. The constraint can’t be assigned to more than one interface element, but the interface element can have more than one AUIConstraint.

Another important class is defined in our app is the AUIUserModel which is a singleton used to get the user static information (gender, and age), and to capture the dynamic actions made by the user.

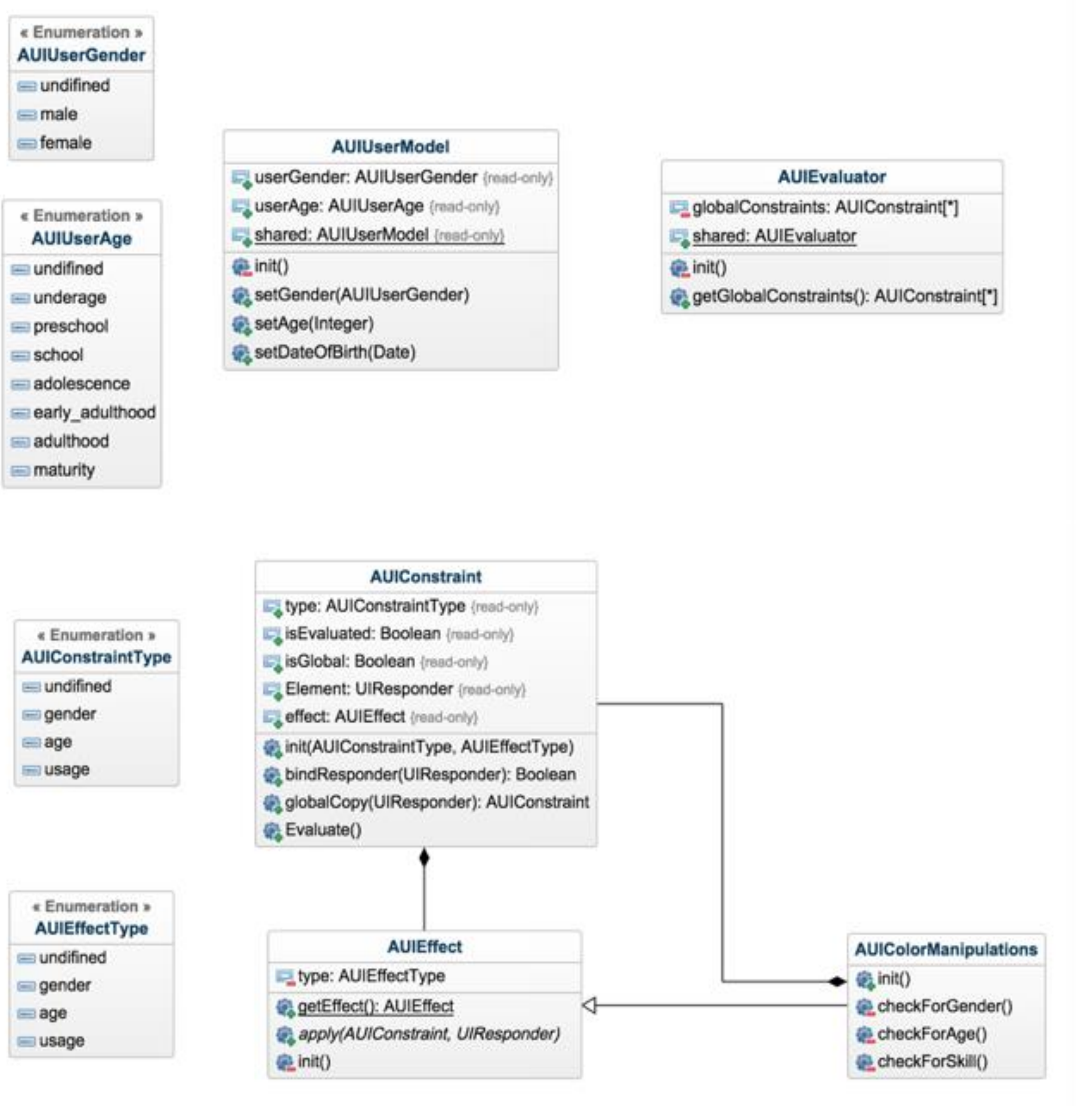


Figure 4.1-2 Core Classes, Class Diagram

The framework execution interleaved with the normal apps execution, so the developers do not need any further action to execute or call the framework effects. For each interface appearance there is a group of methods called by the system, which are the “ViewDidLoad”, “ViewWillAppear”, and “ViewDidAppear”, our framework actions lays in these three methods, so we execute our code while these methods called.

Fiugre 4.1-3 shows the app methods in white boxes, and some of these methods are called when the app starts, and others are called for each interface loaded and appears.

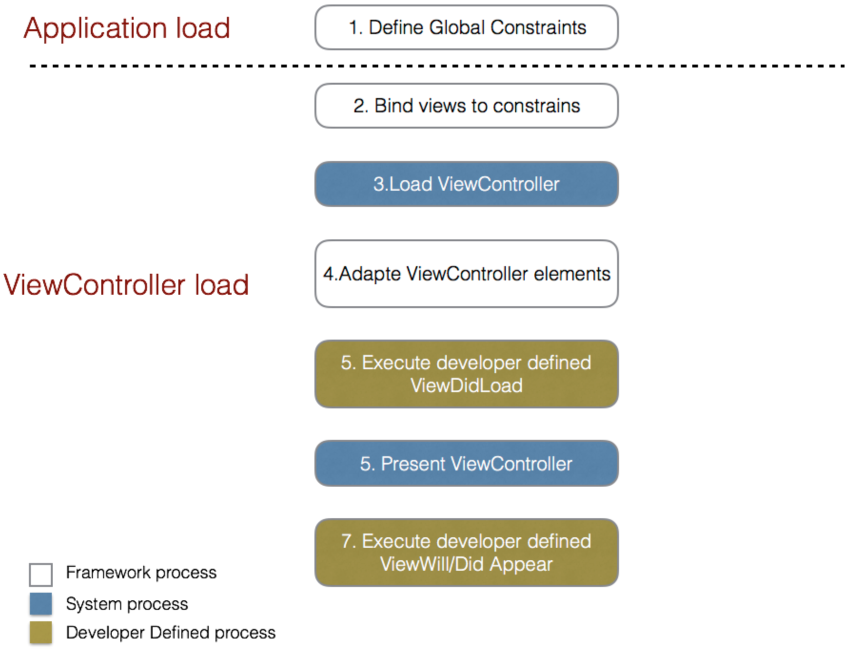

Application load
1. Define Global Constraints
2. Bind views to constrains
3.Load ViewController
ViewController load
4.Adapte ViewController elements
5. Execute developer defined ViewDidLoad
5. Present ViewController
7. Execute developer defined ViewWill/Did Appear
Framework process
System process
Developer Defined process

Figure 4.1-3 Steps of App launch with the framework

4.2 App main classes

4.2.1 AUIEffect Class

This class represent the type of the effect that need to be applied on an element, so the effect class check the type of constraint and check the element to be adjusted, then it adjusts its target attribute based on the user model info.

The change of element attributes actually applied by the element class itself, so we make an extension for each element that contains a modify methods to adapt the color, shape, icons, and etc., but the effect class is the one responsible to call the appropriate method for the element

AUIColorManipulations

This is a sub class of the AUIEffect, and it implements the apply function to check the target element adaptability against the user model, and call out the methods that should adjust the colors of the element.

As mentioned there is a group of extensions for the original interface element, which are, UIView, UILabel, UITextField, UIButton, UIImageView, UINavigationBar, UIToolBar, UITabBar, UIBarButtonItem, UITabBarItem, UISwitch, UISegmentedController, UIDatePicker, UIPickerView, UITableView, UITableViewCell, UICollectionView, UICollectionViewCell, and UITextView.

In the color manipulations class all the extensions handle the color cases such as checking the background color state, is it suitable for this constraint type (e.g. gender), what changes should be made on this element, and so on.

AUIIconStyleManipulations

Icons manipulation can only have applied to UIImageViews since they are the only component that deal with images, this class checks the UIImageViews that has icons effect constraints binded to gender, age, or skills.

The designer should define each adaptable image, and the developer needs to name it using a special naming convention we suggest in our app.

The app tags the Gender constraint with "hash tag symbol "(#), the Age constraint with "asterisk symbol" (*), and the skill with "percentage symbol" (%), so the image to be bound to male user, it should has a name prefix "#m", and "#f" for females, the user can name the image with a combination of these symbols so he can match an image to a specific type of user.

For example, an icon names “user” found in the assets folder, in run time the app will check the icon with the three factors, so if the user is a female, the app searching for image named “user” and contains “#f” the image name can be found in several cases, if it is combined with other factor.

Table 4.2-1 Icons naming with respect to user factors

Image Name	Target Gender	Target Age	Target Skill
user	Cold Start	Cold Start	Cold Start
user#m	Male users	Any	Any
user#f	Female users	Any	Any
user*1	Any	Preschool	Any
user*2	Any	School	Any
user*6	Any	Maturity	Any
user%1	Any	Any	Novice
user%2	Any	Any	Medium
user%3	Any	Any	Expert
Combinations could be like this			
user#f*3%2	Female users	Adolescence	Medium
user%1#m	Male users	Any	Novice
user*6#m	Male Users	Maturity	Any

The developer need to assign a string value for the image refers to the image name, so we add a new variable in UIImageView extension called name, which is the string of the image name that the app will search for it in the assets.

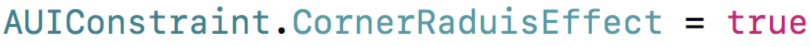
AUIShapeManipulations

Shape manipulations has the same idea of icons manipulations, the user need to define images in the assets folder using the same naming convention for age, gender, and skill symbols. But the difference that is the shape can affect UIViews as layer mask, as will as you can add shape constraint to UIButton, and to UIImageView. Another difference that is there is no image for the cold start state, because normally we use the rectangle as cold start state.

The shape manipulations can be done away from images, based on the corner radius of rectangular shapes, so the developer can set Corner Radius Effect (CRE), or Layer Mask Effect (LME).

Setting Corner Radius Effect

When the user called this effect it will cancel the other one (LME), in the AUIConstraint class there is a Boolean flag points to using the corner radius effect, and it is true by default, if the developer set it to false, it will stop the effect at all, unless the user assigns an image to the Layer Mask Effect.

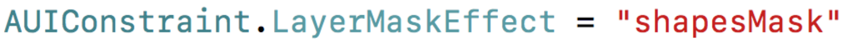


```
AUIConstraint.CornerRaduisEffect = true
```

Figure 4.2-1 Support automatic corners

Setting Layer Mask Effect

When the user called this effect it will cancel the other one (CRE), in the AUIConstraint class there is a string points to the name of the image that will be used as layer mask, and it is null by default, if the developer assign string to it, it will set the CornerRaduisEffect to false.



```
AUIConstraint.LayerMaskEffect = "shapesMask"
```

Figure 4.2-2 Supports views' masks

AUIPositionManipulations

This class will not deal with all elements at the same time, as in the global constraints concept, this class's apply method is called only when there is a position constraint bind to an element in the interface.

Based on our previous study in chapter three, there is two factors can affect the position of item, which are the age, and skills, the age can affect the position by setting the weight for the standard user age, but it is important to define the weight for elements that are related to each other in one list, stack, or table.

On the other hand, the adaptation based on user skill, and usage pattern need to define the dependences between elements and relation, so here this class will find the element, and the related element, then evaluate the element to check if it needs repositioning.

Reposition adaptation should respect another existing adaptation which is also based on constraint, and that the the Auto-Layout that is exists in the XCode IDE.

AUIElementTypeManipulations

The type of element adaptation based only on the usage pattern and user sills, so the system will be able to create alternative elements and re-represent data models based on the original element.

Also this adaptation need to respects the Auto-Layout adaptation.

4.3 Extensions on Original Classes

Usually it is not possible to adjust the original class, but swift, and objective-C, allows us to do such things, so we made many extensions to support our purpose.

Extension UIResponder

The UIResponder is the super class for all interface element we mentioned previously, so anything added to the UIResponder extension will be reachable from all interface elements.

Contains the array of AUIConstraints, the function to bind a constraint with Responder, and the modify function which need to be overridden in the subclasses of the UIResponder.

Extensions to NSObject

The NSObject is the parent for all classes in the swift, and if we add a function for the NSObject it will be defined for all classes in the swift, so we define the swizzle method in the NSObject, and this method is responsible for manipulating an original method, for example if we want to add a print instruction in predefined method in a class but we cannot just open the source code for the class and add the line, also we don't want to make a subclass and override the method so we can add our line and call the super one. All we need it to write our line in separate function and swizzle the original function with this function, so the compiler will execute or function before executing the original function.

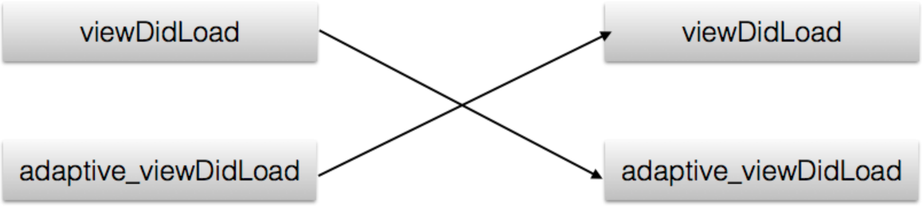


Figure 4.3-1 Methods Swizzle

Extension UIViewController

The UIViewController is the simplest state of the interface which basically contains a main view (UIView), and the developer add any elements he want to this main view, so we want to inform any UIViewController to check on the adaptations when it start, so we need to add code this time not a new function, or variable, but we need to add code to an existing function which is ViewDidLoad, and this can be done using the swizzling methods.

Extensions to UI Elements (UIToolbar, UINavigationItem, UIButton …etc.)

Each UI Element need an extension for several many response, it needs it to adjust colors, shapes, and to detect user interaction.

In our app we almost implement an extension for each type of UI Elements.

Extensions to UIColor

The color class also need to be modified so we can add function to check if the color is gray scaled, check if it clear color (alpha = 0), to get the complementary, or any other related colors as mentioned in the previous chapter, to adjust the color shade, or tint, and to shift the color hue as we defined in chapter three.

4.4 Detecting User Actions

We create a class for events as we mentioned before containing the statuses for the event which are: Unknown event, Single event, unfinished event, and Repeating event. We called it **AUIEvent**.

For each element that can cause event we make a trigger or listener to this element, for example there is a trigger for viewing an interface, and another one for hiding the interface, also there is a trigger for buttons touched, and views scrolled, developer had to define the relation between elements such as a relation between a button and Image, the button open the image in full screen mode, or the relation between a button and slider, the button brings the next image, or previous image in the slider, so the app can define what is the suitable action to take according the user event on an element related to other elements.

Monitored Event Class

In our app now we are monitoring three events only which are the appearing of a ViewController, disappearing of the ViewController, and TouchUpInside for a UIResponder, and record these events in a .plist file, this file will be loaded in the next launch of the app, and all events will be analyzed and grouped, then at each interface starts, we will check the related events to this interface and the app will decide to trigger the action or not, and what action should be triggered.

The monitoring, detection, storage, loading, and analysis processes are implemented on different classes, with different data models.

RecoredEvent Class

This class represent the data model for the event, it contains all the event information such as the date, time, event id, screen index, event type, interface element id, and other possible information.

AUIMointor Class

This class responsible of receiving events from elements, and we mentioned that we cover three possible events, this class will create a RecoredEvent object for the received event, then store it in array of events.

When the app sent to background, or terminated, this class has to store all gathered information about events in a local plist file.

Also when the app launched, this class will restore the plist file that holds all events information, then pass this list to the EventAnalysis class, and it will resume adding new events to the restored list.

EventsAnalysis Class

First it will group the event in a EventCollection so each repeated event will be set in an array inside the EventCollection.

Also it will create a list of EventTrack objects, which hold the sequence of events.

This class is responsible also of analyzing the screens, so it can get all events related to one interface, or one element, and decide if the events worth to trigger an action or not, and find out what action to be triggered.

EventCollection Class

The is a data model class which is receives an array of events, then group these events based on the equality between events' screen_index, element_id, and action_id

Also it can return list of repeated events, single events, or unfinished events.

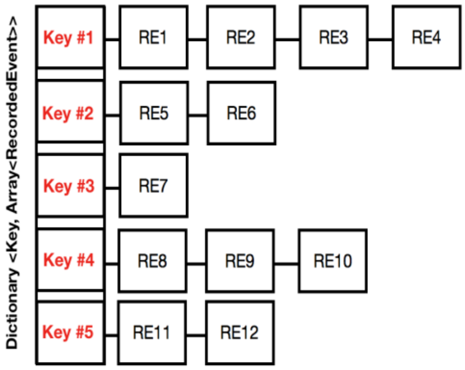


Figure 4.4-1 (a)Chaining recorded events, (b) find chains similarities

EventTrack Class

This is also a data model class, its main purpose is to find the sequence of events and create a list of this sequence.

The Tracks might not be completely equal but they can be match at some percentage.

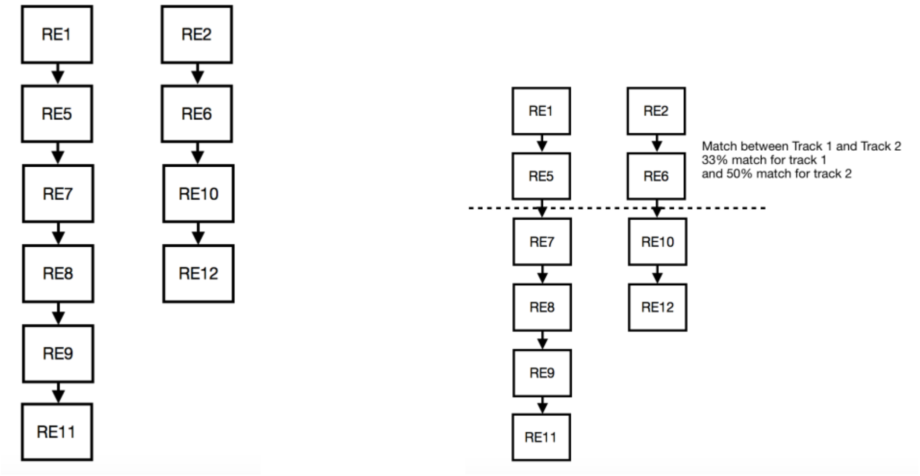


Figure 4.4-2 (a) Chaining recorded events, (b) find chains similarities

App Events

1- View Loaded: Normally no event recorded here, just check if there is an action to be triggered based on previous stored events

2- View Appeared: Record “Show” event for the current ViewController.

3- View Disappeared: Record “Disappear” event for the current ViewController.

4- Button touched: Record “Touch” event for the button.

5- Image touched (view full image): No event will recorded here, since there is no interaction Between UIImage and user UIControlState, if the user wants to make the image interactive, he should add a button on the top of it, so the event will be recorded as UIButton event.

6- Cell did select: Record “Touch” event for the cell with special parameters, such as the data source should be passed to the triggered ViewController or not, and of course the IndexPath of the cell.

7- App started: This is a special event should be stored it will have.

8- App Terminated: This is a special event should be stored it will have.

9- Segment selected: a “Touch” event will be recorded with special parameter points to the selected segment

10- Switch On / Switch Off: a “Touch” event will be recorded with special parameter points to the switch state

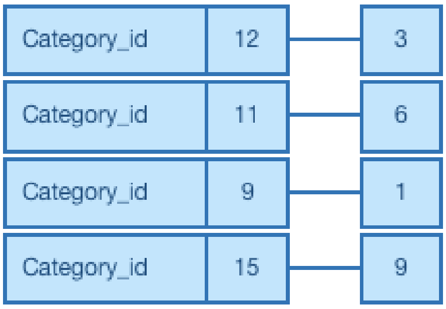
These are not all the expected events that can be fired by the user, there is another group of events we are will not implement it in our app

1- View scrolled down
2- View scrolled up
3- Cell delete button swiped
4- Cell delete button touched
5- Text copied
6- Text pasted
7- Text field selected
8- Text entered
9- Search text entered
10- Search button touched
11- Search canceled
12- Alert Button chosen
13- Action Sheet Button chosen
14- Picker show
15- Picker hide

The app should monitor the data models in runtime also, so when an object moved from one interface to another based on user interaction, or not, this data model should be stored and all the properties and attributes should be registered, the app then will find the matches between the stored models by comparing the attribute name and value, and count the occurrence of each attribute for each model type, this will lead us to the interests of the user and based on these interest we can take appropriate actions.

For example, the user might check on different products in our app, and the app will store the prosperities of these products, such as the product_name, product_id, product_category_id … etc. In our app we give the developer the ability to define what attribute should be monitored so we can avoid storing unrequired data.

If the user did select many products from different screens all these products models are monitored on the category id key, so there will be mutuality in some of these category ids.



Category_id	12	3
Category_id	11	6
Category_id	9	1
Category_id	15	9

Figure 4.4-3 Model Properties Monitoring

In figure 4.4-3 there is a group of categories' ids stored by monitoring products models, the first column represent the key monitored, the second column shows the value of this key, and the last columns shows the number of occurrence for this key, and it is clear that the user here interested in product from category 15 more than other, this can be useful in many ways, the normal way we know is to adapt the flow of data and recommendations as in the recommender systems.

We can benefit from this records on the aesthetic level too, the developer can link the repeating of the key with adaptation action such as changing default configurations, changing element type, or changing element position.

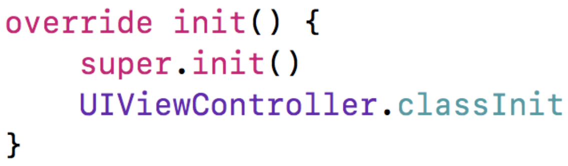
In our example there is a scrollable tab of categories and it selects the "Random" tab by default in the cold start case, but after many usages the system changes the default selection of these tabs to the highest appearing category.

Also we make adaptation on the cells of the products from these categories, so we present the product description, and product rate on the cell that displaying a product from this category.

The developer can predict any other changes can apply to the interface based on this key occurrence, or other keys too.

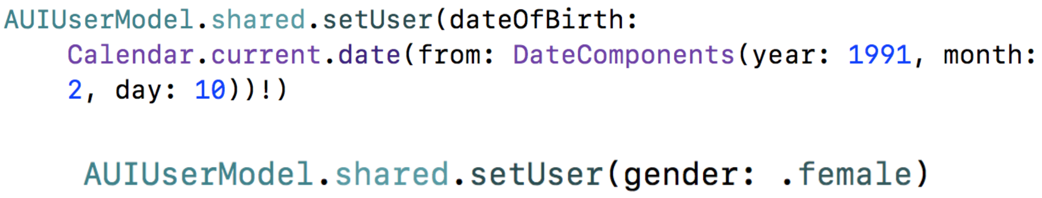
4.5 App Calling Framework

To use call the function responsible of executing the framework, the developer need to do two important things after he adds the code of the implemented framework function, first he need to call the Swizzling function in the AppDelegate class so the new code will be added to all UIViewController.viewDidLoad functions, and he needs to provide the implemented framework specially the UserModel with the source of the static information.



```
override init() {
    super.init()
    UIViewController.classInit
}
```

Figure 4.5-1 Initialize the framework



```
AUIUserModel.shared.setUser(dateOfBirth:
    Calendar.current.date(from: DateComponents(year: 1991, month:
    2, day: 10))!)

AUIUserModel.shared.setUser(gender: .female)
```

Figure 4.5-2 Set User static factors

The developer also need to feed the implemented framework with the end user date of birth, or current age, and the user gender, and refresh the layout and content so it should meet the current user's needs.

As we mentioned that the app keep tracking the user events and record them locally, also classify these events, the developer may monitor a specific object, so the app will store the objects locally and compare its attributes with previous stored objects, for example if the user keep checking products that contains an attribute "category_id" with value "3", the app will find the common attribute in these products which is the category id, and count the appearance of this attribute to understand what products that the user interested in, this will be summarized in special object for the developer so he can take any related action to this product.

The app contains a customized UIView that is used by the framework, to move some UI element from position to another, and from screen to another, such as the case of the "Add to cart” button that is exists in the product details interface, the app set this button in the products list for all products that from the category id = 3.

4.6 Limitations of the app

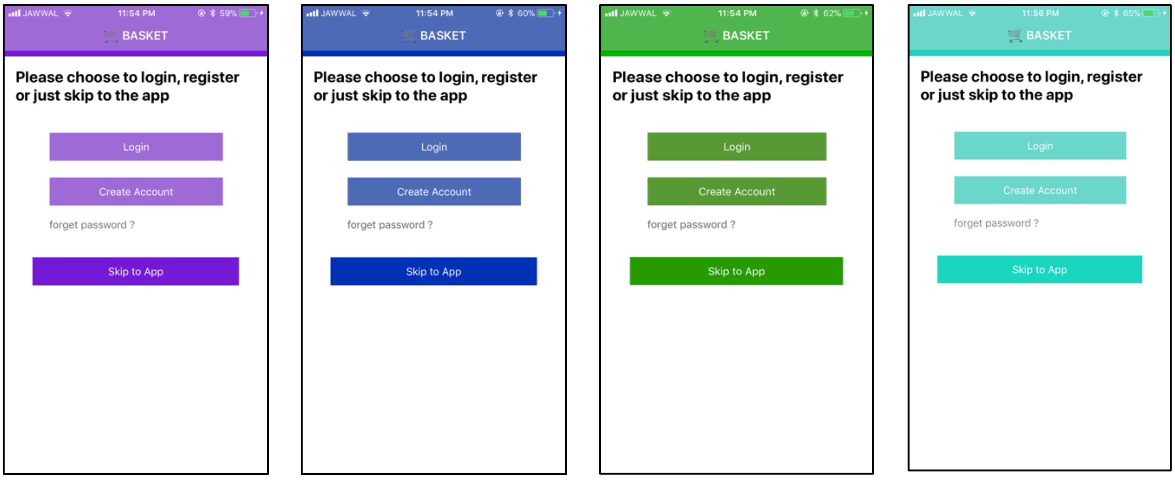
1. The app cannot always match the user gender when changing colors, such as when there is a colored text on a background, we will try to match the background color with user gender, and get the contrast color for the text, and the contrast color may not match the user gender.

2. Unable to make a generic module for the implemented framework, because the adaptation process need to be carefully set, and need very deep studies for each type of apps, since it is not easy to make adaptation and satisfy the users at the same time.

4.7 Application Interface

The App interface contains several screens, starting from landing screen, and branch to the login, register, cart, products, and other screens.

The App first starts on the landing screen which shows the user the options of login, register and skip to the app, usually this is the default screen in the apps that supports the login and registration features, if the user wont login nor register and he clicks on skip to app button, the event will registered and if he keep doing the same step each time he open the app, later the app will start directly by skipping the landing screen, for the first time the app will show hint to the user telling him how he can go back to the landing screen.



Please choose to login, register or just skip to the app

Please choose to login, register or just skip to the app

Please choose to login, register or just skip to the app

Please choose to login, register or just skip to the app

(a) (b) (c) (d)

Figure 4.7-1 App Landing screen in with adaptations, (a) original colors, (b) young male, (c) old woman, (d) little girl

Chapter 5
Testing and evaluation

Chapter 5

Testing and Evaluation

We need to check the satisfaction of the users about the adaptable apps, the frameworks convert the layout of the app to different style, so we need to measure user's satisfaction about two things, which are, the new style that supposed to match their tastes, and the dynamic changing of the app while using the app.

So first we need to make test-bed app (See Appendix A), as normal app, without any adaptation and let a group of users use this app for a period of time about one week, then we will ask them to use the adaptable app for another week, they need to fill a form at the end, and in between the two experiments, these forms will be a survey (See Appendixes B, and C) measuring their acceptance for the new design, and the dynamic changing mechanism.

5.1 Test-bed Application

The adaptation will cover the colors, icons, some element positions, it will consider the gender, age, and user skills, but the elements types' adaptation will not be covered since the categorization of users based on design pattern to the main three categories may take more time that is allowed for the experiment

The App's design in general was examined by "Sanabil TV" which is a local studio for design and motion graphics, to evaluate the design by senior designers and artists, to address the strength and weakness of the design.

Sanabil TV studio's report about the app(See Appendix E), was about the primary design for the original app, they find out the wrong and not suitable design issues, they also gives us recommendations and suggestions to make the app better in its design, they don't have any interference with the adaptation effects, they only judge the non-adaptive app, in the other hand, we take all these recommendations into account and applied these changes before the first experiment started, except the one that is related to the app color since the color we use is a part of our experiment.

5.2 Users Sample Space

We choose a group of iPhone users with different ages and genders with cooperation with Gaza Sky Geeks / Mercy Corps organization (See Appendix D), the group consists of ten members, there is three females, and seven males, also the ages were vary from 18 years old to 40 years old. Another group of children was selected in the same organization the ages were vary from 12 years old to 17 years old it was four members, two are males, and two are females. About the skills, generally the application measures the user skills based on the user usage Pattern, so by default the skill is in a cold start, after many usage times the app will determine each user's skill.

5.2.1 First Experiment

The first experiment will be using the app without any adaptation, and we will get feedback from the users about the app as normal app, to measure their satisfaction about it. Also we need to understand what will make the app better in their point of view.

To measure their satisfaction, we need them to answer a survey made especially for this purpose. The survey asks the users about the design of element such as the colors of views, buttons, and labels, the shapes of views the sharp edges and rounded edges, also it asks about the position of specific elements, and the method of performing some takes, these point that we ask about in this survey expected to be adaptable in the next experiment and might change due to the user static and dynamic factors.

To ensure that users did use the app as they claim in the survey, we did review our database server, and monitor their interactions, we found that all users did create account and the log table shows many app launching for all users, also we notice that users did make orders and purchase items, that was for the first group (Early Adulthood), the second group was monitored directly, we gave the app for the kids, and let them use it for two hours, and then we ask them to fill the survey.

Table 5.2-1 Early Adulthood Group for Non-Adaptive App

User	1	2	3	4	5	6	7	8	9	10					
Gender	M	M	F	F	M	M	M	F	M	M	Scale				
Age	27	38	25	34	40	33	23	28	19	18	Very Bad	Bad	Acceptable	Good	Very Good
Q 1	3	2	4	3	1	3	2	4	2	3	10%	30%	40%	20%	0%
Q 2	3	3	5	4	3	4	3	5	4	4	0%	0%	40%	40%	20%
Q 3	2	2	3	3	1	2	3	3	2	1	20%	40%	40%	0%	0%
Q 4	4	5	5	4	4	4	5	4	3	4	0%	0%	10%	60%	30%
Q 5	3	3	4	4	3	4	5	5	4	4	0%	0%	30%	50%	20%
Q 6	3	4	5	4	3	4	5	5	4	5	0%	0%	20%	40%	40%
Q 7	3	4	2	3	4	3	3	2	3	4	0%	20%	50%	30%	0%
Q 8	3	2	3	2	1	1	3	2	2	1	30%	40%	30%	0%	0%
Q 9	1	1	2	1	1	1	2	3	1	2	60%	30%	10%	0%	0%
Q 10	3	3	2	3	2	3	4	2	3	4	0%	30%	50%	20%	0%
Q 11	3	3	2	3	2	3	4	3	3	3	0%	20%	70%	10%	0%
Q 12	3	2	4	3	3	2	3	3	3	4	0%	20%	60%	20%	0%
Q 13	3	4	5	4	3	4	4	5	4	3	0%	0%	30%	50%	20%
Q 14	2	3	4	3	3	3	5	5	3	4	0%	10%	50%	20%	20%
Q 15	2	1	2	2	1	1	3	3	1	2	40%	40%	20%	0%	0%
Q 16	4	3	4	4	5	4	5	5	3	4	0%	0%	20%	50%	30%
Q 17	4	4	5	4	4	3	4	5	5	5	0%	0%	10%	50%	40%
Q 18	3	4	5	4	2	3	3	4	5	4	0%	10%	30%	40%	20%
Q 19	4	4	5	4	4	3	3	4	4	5	0%	0%	20%	60%	20%
											8.4%	15.3%	33.2%	29.5%	13.7%

Based on the experiment we notice that in general we notice that 33.2% of users answers was in the acceptable range which is predicted because the app we designed with respect to the guidelines of UI/UX principles provided by Apple™, and examined by seniors, and we can predict that published apps on the stores right now might fall on the same range, and user can consider it to be acceptable to good.

Table 5.2-2 show the results for the first group by considering the scale rate from 1 to 5, 1 for the worst impression user takes about the app’s design, and 5 for the best impression user takes about the app’s design, and it also show the average for each question based on user gender, the questions are groups relative to UI element characteristic, so in table 5.2-2 we present the average of user's satisfaction based on the UI element characteristic.

Table 5.2-2 Early Adulthood Group for Non-Adaptive App

User	1	2	3	4	5	6	7	8	9	10				
Gender	M	M	F	F	M	M	M	F	M	M				
Age	27	38	25	34	40	33	23	28	19	18	Total	Avg.	M.Avg.	F.Avg.
Q 1	3	2	4	3	1	3	2	4	2	3	27	54.0%	2.3	3.7
Q 2	3	3	5	4	3	4	3	5	4	4	38	76.0%	3.4	4.7
Q 3	2	2	3	3	1	2	3	3	2	1	22	44.0%	1.9	3.0
Q 4	4	5	5	4	4	4	5	4	3	4	42	84.0%	4.1	4.3
Q 5	3	3	4	4	3	4	5	5	4	4	39	78.0%	3.7	4.3
Q 6	3	4	5	4	3	4	5	5	4	5	42	84.0%	4.0	4.7
Q 7	3	4	2	3	4	3	3	2	3	4	31	62.0%	3.4	2.3
Q 8	3	2	3	2	1	1	3	2	2	1	20	40.0%	1.9	2.3
Q 9	1	1	2	1	1	1	2	3	1	2	15	30.0%	1.3	2.0
Q 10	3	3	2	3	2	3	4	2	3	4	29	58.0%	3.1	2.3
Q 11	3	3	2	3	2	3	4	3	3	3	29	58.0%	3.0	2.7
Q 12	3	2	4	3	3	2	3	3	3	4	30	60.0%	2.9	3.3
Q 13	3	4	5	4	3	4	4	5	4	3	39	78.0%	3.6	4.7
Q 14	2	3	4	3	3	3	5	5	3	4	35	70.0%	3.3	4.0
Q 15	2	1	2	2	1	1	3	3	1	2	18	36.0%	1.6	2.3
Q 16	4	3	4	4	5	4	5	5	3	4	41	82.0%	4.0	4.3
Q 17	4	4	5	4	4	3	4	5	5	5	43	86.0%	4.1	4.7
Q 18	3	4	5	4	2	3	3	4	5	4	37	74.0%	3.4	4.3
Q 19	4	4	5	4	4	3	3	4	4	5	40	80.0%	3.9	4.3

Table 5.2-3 Group (1) Average of UI elements characteristics

Early Adulthood 18 - 40	Male Users	Female Users
Satisfaction Average	61.95%	71.93%
Colors	58.6%	78.3%
Design	44.8%	60.0%
Icons and Style	69.1%	65.3%
Positions	64.3%	76.7%

Table 5.2-3 shows the results for the results for the second group as they fill the survey in the first experiment, and the averages based on gender, and the table 5.2-4 shows the results based on the UI elements characteristics.

Table 5.2-4 Group (2) results for first experiment

User	1	2	3	4				
Gender	F	M	F	M				
Age	17	13	12	16	Total	Avg.	M.Avg.	F.Avg.
Q 1	5	4	5	3	17	85.0%	3.5	5.0
Q 2	5	5	5	4	19	95.0%	4.5	5.0
Q 3	4	5	4	3	16	80.0%	4.0	4.0
Q 4	5	5	5	5	20	100.0%	5.0	5.0
Q 5	4	3	2	3	12	60.0%	3.0	3.0
Q 6	4	3	3	3	13	65.0%	3.0	3.5
Q 7	2	2	1	3	8	40.0%	2.5	1.5
Q 8	3	3	2	4	12	60.0%	3.5	2.5
Q 9	2	1	2	2	7	35.0%	1.5	2.0
Q 10	4	5	3	3	15	75.0%	4.0	3.5
Q 11	4	4	3	2	13	65.0%	3.0	3.5
Q 12	4	4	5	4	17	85.0%	4.0	4.5
Q 13	3	4	4	4	15	75.0%	4.0	3.5
Q 14	3	3	4	2	12	60.0%	2.5	3.5
Q 15	2	1	2	2	7	35.0%	1.5	2.0
Q 16	3	2	1	3	9	45.0%	2.5	2.0
Q 17	4	3	4	4	15	75.0%	3.5	4.0
Q 18	3	4	3	4	14	70.0%	4.0	3.0
Q 19	4	3	3	5	15	75.0%	4.0	3.5

Table 5.2-5 Group (2) Average of UI elements characteristics

12-18	Male Users	Female Users
Satisfaction	66.8%	67.9%
Colors	85.0%	95.0%
Design	60.0%	53.3%
Icons and Style	62.0%	60.0%
Positions	60.0%	63.3%

5.2.2 Experiment results discussion

After extracting the surveys and analyze them we found out that in the first group overall satisfaction is about 64.9%. The users' satisfaction for males can reach 61.9%, and 71.9% for females. About colors only the male users were 58.6% satisfied, and female users were 78.3% satisfied, and this might be because the app is in purple which is classified as a feminine color. The general design satisfaction for male was approximately 44.8%, and 60% for female users. About the icons both males and females in the same rate which is 69.1%, 65.3% respectively. Finally, the items position the male users were 64.3% satisfied, and 76.7% is the females' satisfaction average.

From figure 5.2-1 we can notice that the original app design satisfying female users more than male users in almost all directions, except for the icons and style, since the icons are more muscular and we use the shape edges in the original design which is considered preferable by male more than female as explained in chapter three.

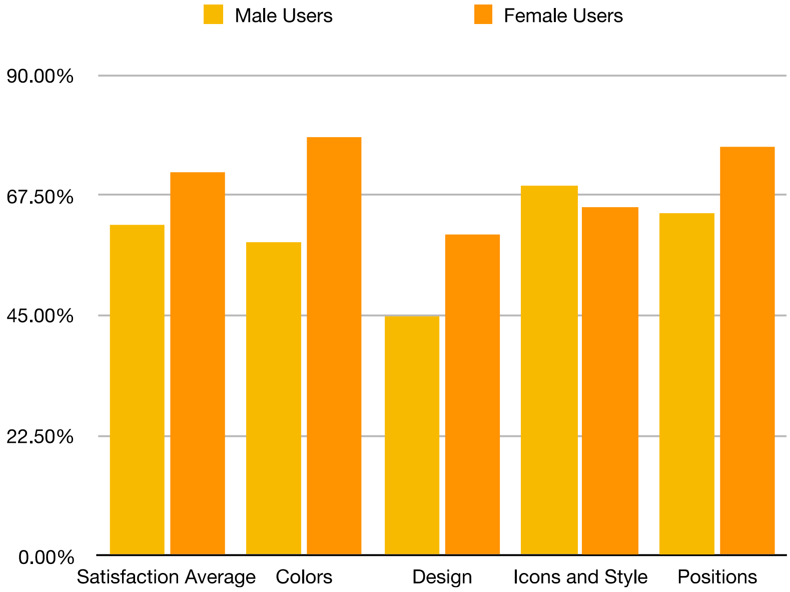


Figure 5.2-1 Users satisfactions for group 1 (Early Adulthood)

The second group which was the adolescence users show not so different results, since the overall satisfaction was around 67.4%, and male users overall satisfaction was about 66.8%, and 67.9% for female users. But there was a huge difference in colors appeal for kids, because male kids give 85% satisfaction, and 95% satisfaction for girls. The overall design was similar to the adults' results giving 60% for male users, and 53% for female. About icons, the results was 62% for male, and 60% for female users. Finally, the acceptance for elements positions was 60% for male users, and 63% for female users.

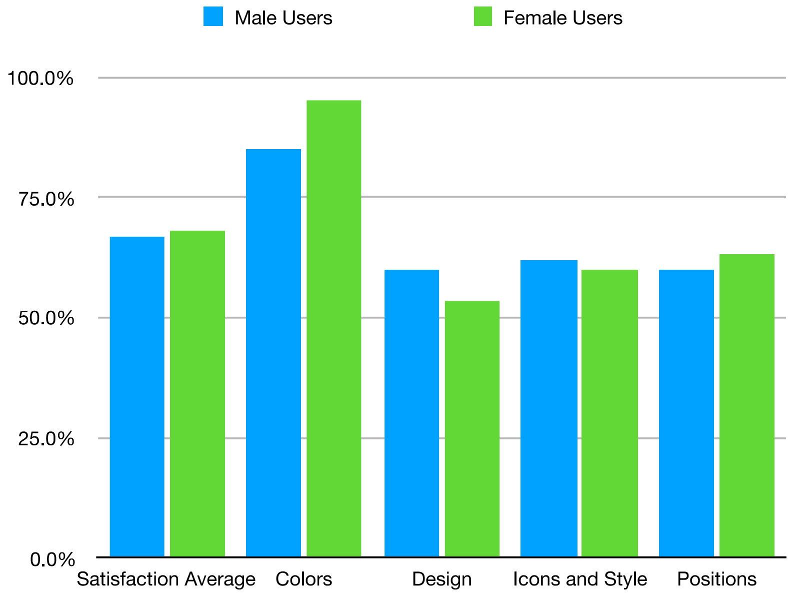


Figure 5.2-2 Users satisfactions for group 2 (Adolescence)

From figure 5.2-2 we can notice that the columns are closed enough to the columns in figure 5.2-1 if we consider the relation between male, and female appeal to the App regardless the actual values, we can tell the female did like the app and become satisfied more than male users because we choose the feminine color from the beginning, but in both groups the icons and styles for male was higher than female because we did use a Sharpe edges design.

To conclude the first experiment, the overall average for both groups (66.94%, 67.4%) was 67.1%. Based on this percentage the app in its current design and colors, would satisfy about 67% of users in both genders, if we tried to generalize this rate we might think it is good at some level, but we with adaptation we expect that we can increase this value to get more users to be satisfied with the app design. And this percentage seems to be good also because users in the sample space are familiar with the marketing apps, and the non-adaptive apps, so we expect to reach higher number of users in adaptive app, since it is a new idea and should satisfy more users.

5.2.3 Second Experiment

Now we will apply the second experiment on the same groups using the adaptive app this time, and they will answer another survey related to the adaptation idea, and the adaptive app itself. This time we expect to get higher percentage of satisfaction since it should meet the desires for different users. We tried to control the experiment by the same circumstances in the first experiment, but this time users already have accounts in the system, and the system knows their genders and ages, so it should react to these two factors directly, and while they use the app, the app will start monitoring their actions, record events, and enhance interfaces based on their actions.

The second survey was divided into three categories, the first one was about the general idea of adaptive apps, and it was evaluated separately from the other two, since it does not represent users' satisfaction, and also it is not in the same scale used as in the other categories and the previous survey which is a quintuple scales from 1 to 5, one for very bad, and five for very good, the first category here uses a triple scale from 0 to 1. Zero for no, a half for maybe, and the one is for yes option.

Table 5.2-5 shows the results of the second survey for the first group, the questions here are divided into three main categories, which are the adaptive idea questions , these questions are the asking about the general idea of adaptation after the users understand the meaning of adaptive interface, the second group is the general changing question, and these questions asking for the changes that noticed by the user, and it is related to the static user model data, the last group asking about the changes that noticed during app usage, and it is related to the dynamic user model.

Table 5.2-6 Early Adulthood group for second experiment

	User	1	2	3	4	5	6	7	8	9	10				
	Gender	M	M	F	F	M	M	M	F	M	M				
	Age	27	38	25	34	40	33	23	28	19	18				
Group	Question											Total	Avg.	M.Avg	F.Avg
Adaptive Idea	Q 1	0.5	0	1	0.5	0	1	1	1	1	0.5	6.5	65.0%	0.6	0.8
	Q 2	0.5	1	1	0.5	1	0.5	1	0	0.5	0.5	6.5	65.0%	0.7	0.5
	Q 3	1	0.5	1	1	1	1	0.5	0	0.5	1	7.5	75.0%	0.8	0.7
	Q 4	0	0	0.5	0.5	0	0.5	1	0	0	0	2.5	25.0%	0.2	0.3
	Q 5	1	1	1	0.5	1	1	0	1	0.5	0.5	7.5	75.0%	0.7	0.8
	Q 6	0.5	0.5	0.5	1	1	1	1	0.5	1	0.5	7.5	75.0%	0.8	0.7
General Changes	Q 1	4	4	5	4	4	3	5	3	4	4	40	80.0%	4.0	4.0
	Q 2	3	2	3	3	1	2	3	4	3	3	27	54.0%	2.4	3.3
	Q 3	3	4	2	3	3	4	5	4	5	4	37	74.0%	4.0	3.0
	Q 4	4	4	5	4	4	2	4	5	5	5	42	84.0%	4.0	4.7
	Q 5	3	3	4	3	4	4	5	4	4	3	37	74.0%	3.7	3.7
Technical level	Q 1.1	5	4	5	4	5	4	4	3	5	4	43	86.0%	4.4	4.0
	Q 1.2	4	3	3	4	3	2	4	3	5	5	36	72.0%	3.7	3.3
	Q 1.3	5	4	5	5	4	4	5	4	3	4	43	86.0%	4.1	4.7
	Q 1.4	4	4	3	3	5	4	3	5	4	5	40	80.0%	4.1	3.7
	Q 1.5	4	2	3	2	3	2	4	3	4	4	31	62.0%	3.3	2.7
	Q 2.1	5	4	5	5	4	5	4	5	5	5	47	94.0%	4.6	5.0
	Q 2.2	4	4	5	4	3	4	5	5	5	5	44	88.0%	4.3	4.7
	Q 2.3	3	4	4	5	5	4	5	3	3	2	38	76.0%	3.7	4.0
	Q 2.4	2	1	2	3	1	3	2	4	3	3	24	48.0%	2.1	3.0

The tables, 5.2-6 and 5.2-7 are presenting the averages for the first group based in the questions groups (Idea, general, and technical) also it shows the average of users' satisfaction, and is shows the averages of UI elements characteristics acceptance by users.

Table 5.2-7 Group (1) Averages based on questions groups

Early Adulthood 18 - 40	Male Users	Female Users
Satisfaction Average	75.10%	76.67%
Adaptive Idea Avg.	63.1%	63.9%
General Changes Avg.	72.6%	74.7%
Technical Changes Avg.	76.5%	76.5%

Table 5.2-8 Group (1) Average of UI elements characteristics

Early Adulthood 18 - 40	Male Users	Female Users
Colors	81.7%	78.7%
Style and Icons	64.8%	71.1%
Positions	86.7%	95.6%

Table 5.2-8 shows the results of the second survey for the second group, and the tables 5.2-9, and 5.2-10 are presenting the averages for the second group based in the questions groups (Idea, general, and technical) also it shows the average of users' satisfaction, and is shows the averages of UI elements characteristics acceptance by users.

Table 5.2-9 Group (2) for second experiment

	User	1	2	3	4				
	Gender	F	M	F	M				
	Age	17	13	12	16				
Group	Question					Total	Avg.	M.Avg	F.Avg
Adaptive Idea	Q 1	1	0.5	1	1	3.5	87.5%	0.8	1.0
	Q 2	0.5	1	1	0.5	3	75.0%	0.8	0.8
	Q 3	1	0.5	1	0.5	3	75.0%	0.8	0.8
	Q 4	0.5	1	1	1	3.5	87.5%	0.8	1.0
	Q 5	1	1	1	1	4	100.0%	1.0	1.0
	Q 6	0.5	1	1	1	3.5	87.5%	0.8	1.0
General Changes	Q 1	4	4	2	5	15	75.0%	4.0	3.5
	Q 2	5	2	5	4	16	80.0%	3.5	4.5
	Q 3	3	3	3	3	12	60.0%	3.0	3.0
	Q 4	4	5	4	2	15	75.0%	4.5	3.0
	Q 5	5	2	3	5	15	75.0%	3.5	4.0
Technical level	Q 1.1	5	4	5	5	19	95.0%	4.5	5.0
	Q 1.2	3	4	5	2	14	70.0%	3.5	3.5
	Q 1.3	3	2	4	4	13	65.0%	2.5	4.0
	Q 1.4	5	5	4	3	17	85.0%	5.0	3.5
	Q 1.5	5	4	5	5	19	95.0%	4.5	5.0
	Q 2.1	5	4	4	5	18	90.0%	4.5	4.5
	Q 2.2	5	4	2	3	14	70.0%	4.5	2.5
	Q 2.3	5	5	3	5	18	90.0%	5.0	4.0
	Q 2.4	4	3	4	2	13	65.0%	3.5	3.0

Table 5.2-10 Group (2) Averages based on questions groups

12-18	Male Users	Female Users
Satisfaction Average	80.00%	75.71%
Adaptive Idea Avg.	79.2%	91.7%
General Changes Avg.	74.0%	72.0%
Technical Changes Avg.	83.3%	83.3%

Table 5.2-11 Group (2) Average of UI elements characteristics

12-18	Male Users	Female Users
Colors	78.0%	78.0%
Style and Icons	83.3%	83.3%
Positions	76.7%	73.3%

Table 5.2-12 All users results for the "Adaptive idea" questions group

	YES	NO	MAYBE
Question 1	8	2	4
Question 2	6	1	7
Question 3	8	1	5
Question 4	4	6	4
Question 5	10	1	3
Question 6	8	0	6

5.2.4 Experiment results discussion

Results of the second experiment was better as expected than results from the first experiment, the difference was not two large between the overall satisfaction especially in the first group (Early adulthood), it was about 9% difference.

The overall satisfaction for both male and female users in the first group was about 75.6% which is higher than the percent we get from the first experiment and this leads to adaptation will increase satisfaction, not just this percentage that lead to this theory, but if we check all the factors we measure we will notice that all values are higher than the values from first experiment, but we previously knew that adaptation on user interface will improve users satisfaction from previous studies and experiments. Now we want to prove the effectiveness of our proposed framework.

Our framework focuses on changes of colors, the users' satisfaction we gain from the first experiment about colors 58.6% for male, and 78.3% for female, as average we get 68.5% colors satisfaction, but after we applied our framework the males' satisfaction about color increased to 81.7%, and females' satisfaction stays in the range of 78%, this can stability in the female side can be explained because the original colors for the app was feminine colors, and the changes on these colors will not so different from the original color, also the users are still comparing the new colors with the original colors, so male users see huge difference between the two colors, and that was cause the large change in the males' results.

Another factor we tried to measure is the style and icons, the original satisfaction was 69%, and 65% for males, and females respectively, after the adaptive app installed and used the percentages changes to 64.8%, and 71.1% for male and females respectively. The stability in the male side can be explained in the same way as we explained the colors, since the app uses masculine shapes and icons as default, so we did not apply and noticeable changes for male users on the shapes and style, on the other hand female jumps from 65 to around 71 and that because they now more satisfied.

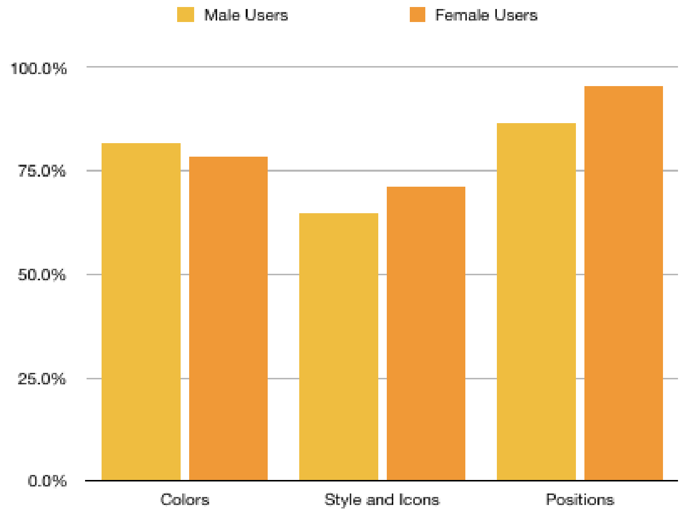


Figure 5.2-3 Users satisfactions for group 1 (Early Adulthood) one new interface attributes

Finally the elements positions, the first experiment gives us 64.3% and 76.7% as satisfaction on elements positions, but the second experiment shows us a large change in the satisfactions, it becomes 86.7% and 95.6% for males, and females respectively. This change leading to the effectiveness of our proposed framework of elements positions changing.

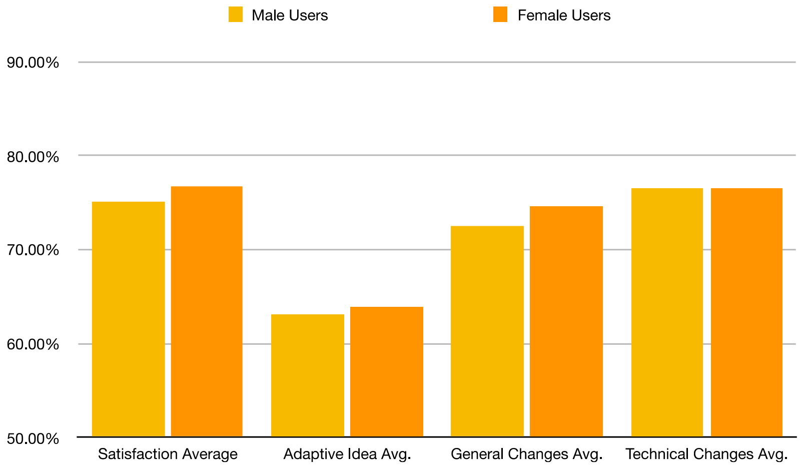


Figure 5.2-4 Users satisfactions for group 1 (Early Adulthood) on changes

Figure 5.2-4 shows that the satisfaction average increased over the 70s, also the changing process itself is considered to be satisfying since the total average is 73.6% for both male and female, and when we ask the users of specific changes the average of satisfaction of these things was 76.5%.

The Adolescence group of users overall satisfaction rate jumps from 67.4% to 77.9%, about 10 degrees difference appears as a result of app adaptation. In General male users were more satisfied than female users in this group, and that was because when they compare the adaptive app with the original app, male users can see wide difference, but female may not.

The Figure 5.2-5 shows that the overall satisfaction for male users reach 80%, and 75.1% for female users, and general changes average for both groups was closed and around 73%. The specific items and attributes changes was about 83.3% for both groups.

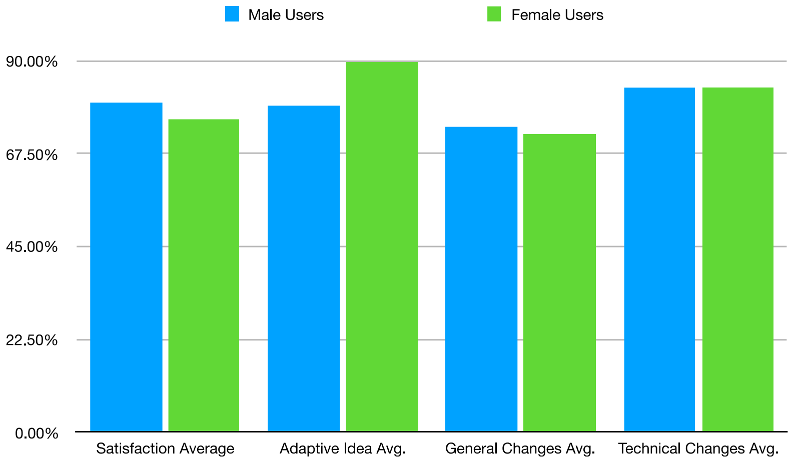


Figure 5.2-5 Users satisfactions for group2 (Adolescence) on changes

On the other hand, when we check on users' satisfaction about interface attributes (Colors, Icons, and Positions) we notice that the style and icons was the most effected with changes since it jumps from 62.0% and 60% to 83.3% for both genders, this change meaning that the new icons used for this age was suitable and satisfying than the original one, and this leads to the importance of icons and style for this age and younger ages.

Changes in colors show odd results, since it decreased from over 85% to about 78%, and this may lead to little failure in the chosen colors for this age, from figure 5.2-5 and figure 5.2-6 we can notice that users did enjoy the change of colors, and thinks this is satisfying, but they did not actually like the result color.

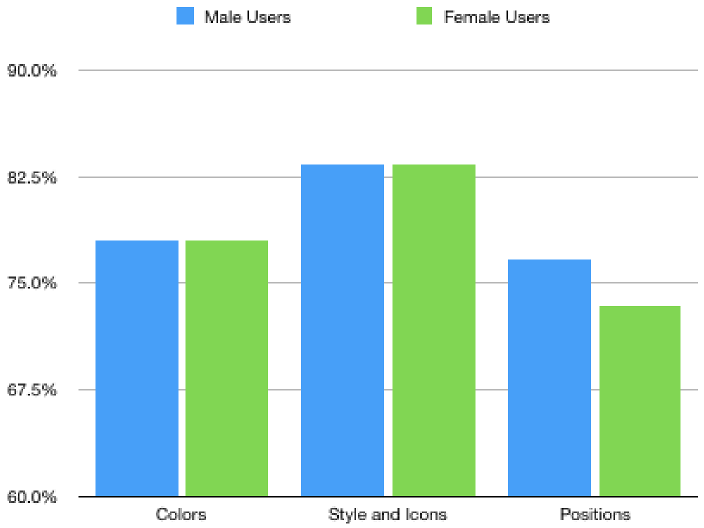


Figure 5.2-6 Users satisfactions for group2 (Adolescence) on new interface attributes

Finally, and as shown in figure 5.2-6 the position changes satisfaction was about 75% and this is another failure in the framework since it decreased from 91.1% for items position to 75%, the reason for that was that this group of users felled confused when items change its position, and this might be caused by many possible factors such as their psychology, or it is really annoying, or they just don’t like it.

Our survey includes a group of questions asks the user about the idea of adaptation, The novelty of the idea considered by 65% buy the early adulthood users as novel idea, and about 75% of the users will use apps that is using adaptable techniques, this is similar to the overall satisfaction average we get from these users.

Chapter 6
Conclusion and Future work

Chapter 6

Conclusions and Future Work

In this chapter we are going to discuss the experiments and results in more details, also we are going to clarify the differences between the final conclusion and the proposed framework.

6.1 Conclusions

The framework we proposed was a theoretical guidelines for designers and developers to make their apps adaptable and responsive to different user factors, these factors was the user gender, age, and skills, and they will affect different elements attributes, which are the color, icons, shapes, elements' positions, and types, also we take into account other adaptable issues such as the app interfaces sequence.

The experiments show a very close results to the expected results, we notice in the first group that the satisfaction grows for both genders in all examined attributes, the color changes affects the male users more than the female since the original color used was a feminine color. On the other hand, the shapes and icons more masculine in the original design so they affect the female user while adaptation more the male users. The other group shows odd results in for some attributes and that's mean the suggest adaptation for ages less than 18 maybe not accurate, and might need further check, or it was just because of the sample space we use the quantity of users we examine.

But in general the framework guidelines can survive in ages larger than 18, and as the experiments show it can be surely correct till the age of 40. We expect that if we enlarge the scale of experiments' sample space, we might have more accurate results specially for ages less than 18, also we believe that if the app was completely designed by UI/UX experts and the icons chosen carefully by experts that also will support the results we get, and might increase the distance between the original user satisfaction and the satisfaction after adaptation.

In our surveys we asked the users about the idea of adaptive app is self, we notice that about 43% of users thinks it is a novel idea, in fact the adaptation starts at eighties, but according to sample space, and they are not related directly to the Information Technology field, they are just normal users. They did not enjoy the app with an average about 43%, this might be happening because the app is not in reality use, and they did not use it for a purpose, the enjoy ability considered to be one of the satisfaction factors, and that could be a reason for the low distance between the two satisfaction averages.

6.1.1 Pros and cons of this framework

The advantages of the framework can be summarized in a few points:

1. It can be used as a guidance for targeted apps to make the aesthetics elements matches the users need and wishes.
2. The app will become easier to use each time, this will make the updates on the app, and the adding of new feature easy to reach since the app will previously know if the user is interested in these new features or not.
3. Adaptation encourages users to use the app more than normal apps, since they are truly want the app to know their personality.

And the disadvantages also can be set as following:

1. Overdo of adaptation on interface elements, may lead to many different clones of the same app with different interfaces, this will cause lack of support, and user assistance.
2. The adaptation for some attribute may not match developer, or designer desires.
3. Some adaptation might be not acceptable for the end user, and he want to undo it, this will add more work on the developer side.

6.2 Future Work

We intend to enhance the experiment in the future work, so we need to get a new design from experts that matches all the design standers for a mobile platform, and we need to release the app as real app in the store to expand the size of participate users group. Also we need to figure out a new technique to measure users' satisfaction other than the surveys such as instance analyzing for user interaction with interface.

Another assumed feature that can be added to make the adaptability more efficient is that we can detect user age, and gender automatically not just by linking previous accounts of the user, but we can capture the user photo directly by the camera, and using face recognition techniques we can possibly know the gender and the age group of the user.

Also it worthy to mention that we could integrate some of the other related work with our framework so we can add more user factors, such as the culture, and the educational level, we can benefit from user characteristics, such as movies genre that user likes, habits, books, music, and others.

We believe adaptation cannot stop at this point, and it is larger than this small and narrow research field. But we hope that we complete this research to cover more over factors, and attributes, to finally form a full usable framework for designers and developers

References

Akiki, P. A., Bandara, A. K., & Yu, Y. (2014). *Cedar Studio: An IDE Supporting Adaptive Model-Driven.* Milton Keynes, United Kingdom: Computing Department, The Open University.

Alexdmr. (2008). *Design by example of adaptive User Interfaces* . Retrieved 2017, from YouTube: https://www.youtube.com/watch?v=QFlxj4Obs7g

ALFRED, K. (2001). Generic User Modeling Systems. *User Modeling and User-Adapted Interaction* , *11* (1), 49-63.

Al-Razgan, M. S., Al-Khalifa, H. S., Al-Shahrani, M. D., & AlAjmi, H. H. (2012). Touch-Based Mobile Phone Interface Guidelines and Design Recommendations for Elderly People: A Survey of the Literature. *Proceedings of the 19th international conference on Neural Information Processing* . Doha, Qatar.

Arhippainen, L., & Tahti, M. (2003). Empirical Evaluation of User Experiance in Two Adaptive Mobile Application Prototypes. *Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia.*

Brajnik, & Tasso. (1994). A shell for developing non-monotonic user modeling systems. *International Journal of Human-Computer Studies* , *40*, 31-62.

Brusilovsky, P. (2007). In P. Brusilovsky, *The Adaptive Web: Methods and Strategies of Web Personalization.*

Carson, N. (2017, April 11). *5 ways to use shape psychology in logo design*. Retrieved 2018, from Creative Bloq: https://www.creativebloq.com/features/5-ways-to-use-shape-psychology-in-logo-design

Christie, M. (2017, September 11). *The psychology of logo shapes : A designer's guide*. Retrieved 2018, from Creative Bloq: https://www.creativebloq.com/logo-design/psychology-logo-shapes-8133918

Chrysoula, G., Anastasios, P., & Dimitrois, Z. (2012). The importance of mobile interface icons on user interaction. *International Journal of Computer Science and Applications* , 92-107.

Color & Vision Matters. (n.d.). Retrieved 2018, from Color Matters: https://www.colormatters.com/color-and-vision/color-and-vision-matters

Demeure, A., Meskens, J., Luyten, K., & Coninx, K. (2009). Design by Example of Graphical User Interfaces Adapting to Available Screen Size. *Computer-Aided Design of User Interfaces VI.* London: Springer, London.

Designing a user interface in 5 steps. (2014). Retrieved 2017, from Designn Blog: http://www.blog.designn.org/udara/designing-user-interface-5-steps/

Deuschel, T., & Scully, T. (2016). On the Importance of Spatial Perception for Design of Adaptive User Interface. *IEEE 10th International Conference on Self-Adaptive and Self-Organization System (SASO).*

Dogtiev, A. (2018, April 20). *App Download and Usage Statistics 2017*. (Businessof Apps) Retrieved April 21, 2018, from http://www.businessofapps.com/data/app-statistics/

Ehlert, P. (2003). *INTELLIGENT USER INTERFACES Introduction and survey.*

Erikson, E. (1963). Psychosocial development stages.

Feuerstack, S., Blumendorf, M., Schwartze, V., & Albayrak , S. (2008). Model-based layout generation. *Conference: Proceedings of the working conference on Advanced.* Berlin.

Flat Icons. (n.d.). Retrieved from Flat Icons: https://www.flaticon.com/

Frassanito , P., & Pettorini, B. (2008). *Pink and blue: the color of gender*. Rome, Italy: Pediatric Neurosurgery, Catholic University.

Gajos, K. a. (2006). Exploring the Design Space for Adaptive Graphical User Interfaces. *Microsoft, Advanced Visual Interfaces* .

Gajos, K., & Weld, D. S. (2004). SUPPLE: automatically generating user interfaces. *Proceedings of the 9th international conference on Intelligent user interfaces, 04*, pp. 93-100. Portugal.

Gardner, E. (2009, Mar. 15). *A Math-Based Approach to Color Theory Using Hue, Saturation, and Brightness (HSB)*. Retrieved 2017, from Ethan Gardner.

Google Inc. (n.d.). *Design - Metrial Desgin*. Retrieved 11 16, 2018, from Material Design: https://material.io/design/

Hall, W., & Hothi, J. (1998). *An Evaluation of Adapted Hypermedia Techniques Using Static User Modelling.* UK: Southampton University, Electronics and Computer Science.

Haller, K. (2011, July 29). *colour & design surgery…how does culture influence colour*. Retrieved Nov 2018, from KarenHaller: http://karenhaller.co.uk/blog/colour-design-surgery-how-does-culture-influence-colour/

Hallock, J. (2003, March). *Colour Assignment By Joe Hallock.* Retrieved 11 2018, from Colour Assignment: http://www.joehallock.com/edu/COM498/index.html

Hassenzahl, M. (2005). The Thing and I, Understanding the relation between user and product. In M. Hassenzahl, *Funology* (Vol. 3, pp. 31-34).

HAUFF, A. (2018, Aug 29). *The Know It All Guide To Color Psychology In Marketing*. Retrieved Sep 2018, from CoSchedule: https://coschedule.com/blog/color-psychology-marketing/

Hervás, R., & Bravo, J. (2011). Towards the ubiquitous visualization: Adaptive user-interfaces based on the Semantic Web. *Interacting with Computers , 23* (1), 40-56.

Holzinger, A., Geier, M., & Germanakos, P. (2012). On the Development of Smart Adaptive User Interfaces for Mobile e-Business Aplications. *Data Communication networking*, (pp. 205-214).

Hsin-Hsi, L., Yang-Cheng, l., Chung-Hsing, Y., & Chein-Hung, W. (n.d.). User-Oriented Design for Optimal Combination on a Product Design.

Hulbert, A., & Ling, Y. (2007). Biological components of sex differences in colour preference. *Current Biology , 17* (16).

Kurosu, & Masaaki (Eds.). (2015). Human-Computer Interaction: Design and Evaluation. *17th International Conference, HCI International 2015.* Los Angeles.

Icons - Style - Metrial Design. (n.d.). (inc., Google) Retrieved 2017, from Material Design: https://material.io/guidelines/style/icons.html

Jiangfan, F., & Yanhong, L. (2015). Intelligent Context-Aware and Adaptive Interface for Mobile LBS. *Computational Intelligence and Neuroscience* .

Kobsa, A. (2000). *Generic User Modeling Systems.* California: Department of Information and Computer Science, University of California.

Langley, P. (1991). User Modeling in Adaptive Interfaces. *Seventh International Conference on User Modeling.* Heidelberg.

learn. (2011, July). *HUE, VALUE, SATURATION | learn*. Retrieved Nov 2018, from learn.: http://learn.leighcotnoir.com/artspeak/elements-color/hue-value-saturation/

Lee, W.-Y., Gong, S.-M., & Leung, C.-Y. (2009). *Is Color Preference Affected by Age Difference.* Tatung University, Department of Industrial Design. Taiwan: Tatung University.

Lidwell, W., Holden, k., & Butler, J. (2003). *Universal Principle of Design.* USA: Rockport Publishers, Inc.

Lopez, W., Merlino, J., & Rodriguez-Bocca, P. (2017). Vector representation of internet domain names using a word embedding technique. *Computer Conference (CLEI) XLIII.* Latin American.

MCGREGOR, L. (2016, June 24). *The Basic Properties of Color*. Retrieved July 2018, from The Beat: https://www.premiumbeat.com/blog/basic-properties-color/

Metrovic, N., & Mena, E. (2002). Adaptive User Interface for Mobile Devices. In N. Metrovic, & E. Mena, *Interactive System : Design Specification, and Verification* (pp. 29-43). Berlin: Springer Berline, Heidelberg.

Miao, R. (2013). *Colour Preference.*

Moss, G. (2003). The Implications of the Male and Female Design Aesthetic for Public Services. *The Innovation Journal: The Public Sector Innovation Journal , 8*, 11.

Passig, D., & Levin, H. (2001). The Interaction between Gender. Age, and Multimedia Interface Design. 241-250.

Pei-Luen, P., & Jia-Wen, H. (2007). Interaction Devices and Web Design for Novice Older Users. *Educational Gerontology* , 19-40.

Reinecke, K. (2010). *Culturally adaptive user interface.* Thesis, University of Zurich, Zurich - Switzerland.

Sakamoto , K. (2014). Cultural Influence to the Color Preference According to Product Category. *INTERNATIONAL CONFERENCE ON KANSEI ENGINEERING AND EMOTION RESEARCH.* Japan: Kyoto Institute of Technology.

Simon. (2016). *Psychology Of Color In Logo Design*. Retrieved 2017, from The Logo Company: https://thelogocompany.net/blog/infographics/psychology-color-logo-design/

Theophanis, T., & Schraefel, m. (2005). An Empirical Assessment of Adaptation Techniques.

User Modeling. (2016, December 9). Retrieved Jun. 5, 2017, from Wikipedia: https://en.wikipedia.org/wiki/User_modeling

Wang, H., Hung, S., & Liao, C. (2007). A survey of icon taxonomy used in the interface design. *Preceeding of the 14th European conference on Cognitaive ergonomics* (pp. 203-206). London: ECCE '07 .

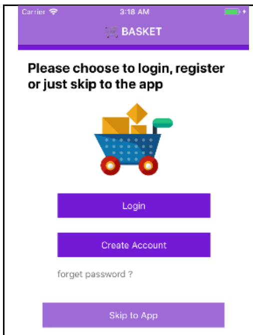
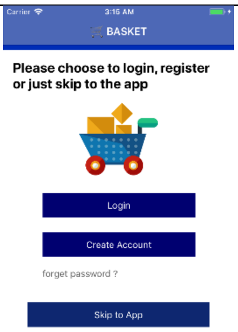
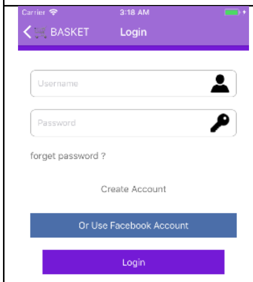
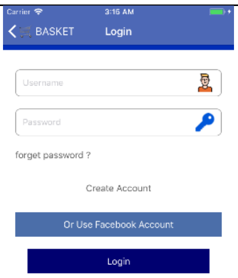
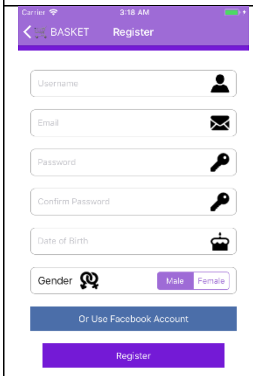
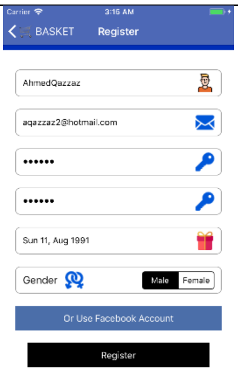
Weiss, E. (1994). *Making Computers People-Literate.* San Francisco: Jossey-Bass Inc.

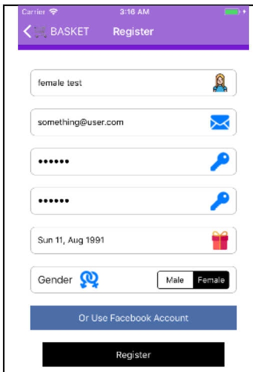
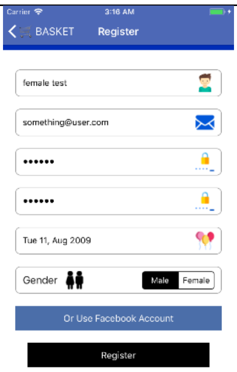
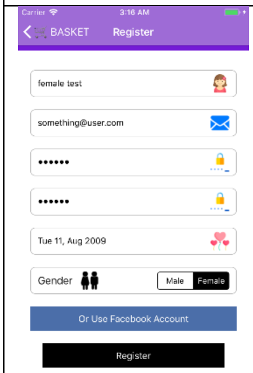
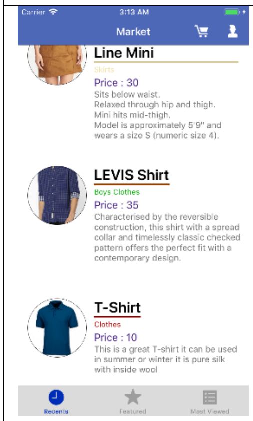
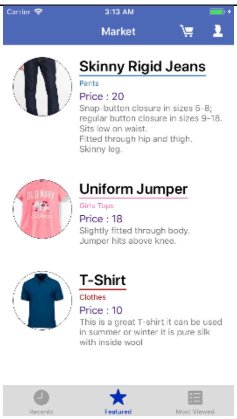
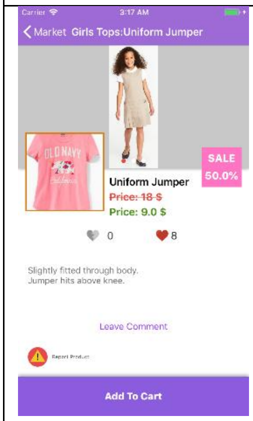
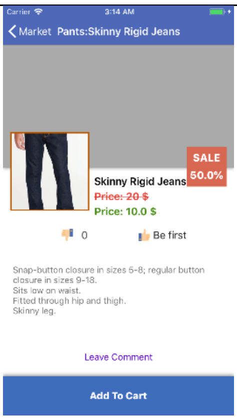
Wesson, J. L., Singh, A., & van Tonder, B. (2009). Can Adaptive Interfaces Improve the Usability of Mobile Applications. *Nelson Mandela Metropolitan University* .

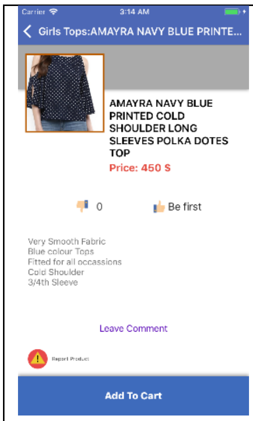
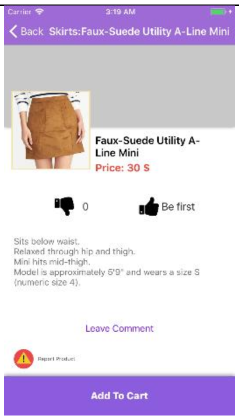
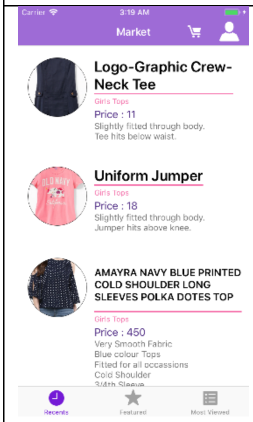
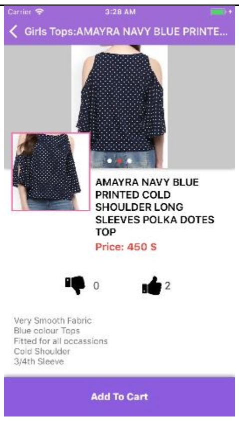
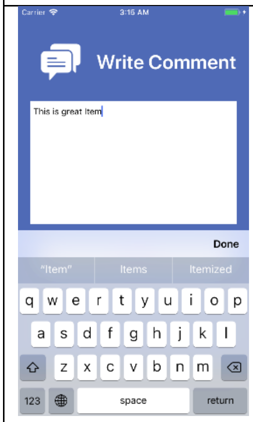
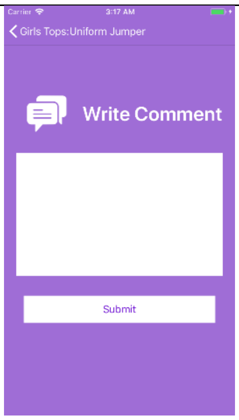
Work, S. (2011). *True Colors – Breakdown of Color Preferences by Gender*. Retrieved 2017, from Kissmetrics Blog: https://blog.kissmetrics.com/gender-and-color/

Appendix A « Basket App » User Interfaces

App Icone

 Carrier 3:18 AM BASKET Please choose to login, register or just skip to the app Login Create Account forget password ? Skip to App	Main Interface for female user	 Carrier 3:15 AM BASKET Please choose to login, register or just skip to the app Login Create Account forget password ? Skip to App	Main interface for male user
 Carrier 3:18 AM BASKET Login Username Password forget password ? Create Account Or Use Facebook Account Login	Login interface for female user with unknown age	 Carrier 3:15 AM BASKET Login Username Password forget password ? Create Account Or Use Facebook Account Login	Login interface for male user with 27 years old
 Carrier 3:18 AM BASKET Register Username Email Password Confirm Password Date of Birth Gender Male Female Or Use Facebook Account Register	Register interface for female user with unknown age	 Carrier 3:15 AM BASKET Register AhmedQazzaz aqazzaz2@hotmail.com •••••• •••••• Sun 11, Aug 1991 Gender Male Female Or Use Facebook Account Register	Register interface for male user with 27 years old

 Carrier 3:16 AM BASKET Register female test something@user.com •••••• •••••• Sun 11, Aug 1991 Gender Male Female Or Use Facebook Account Register	Register interface for female user with 27 years old	 Carrier 3:16 AM BASKET Register female test something@user.com •••••• •••••• Tue 11, Aug 2009 Gender Male Female Or Use Facebook Account Register	Register interface for male user with 9 years old
 Carrier 3:16 AM BASKET Register female test something@user.com •••••• •••••• Tue 11, Aug 2009 Gender Male Female Or Use Facebook Account Register	Register interface for female user with 9 years old		
 Carrier 3:13 AM Market Line Mini Skirts Price : 30 Sits below waist. Relaxed through hip and thigh. Mini hits mid-thigh. Model is approximately 5'9" and wears a size S (numeric size 4). LEVIS Shirt Boys Clothes Price : 35 Characterised by the reversible construction, this shirt with a spread collar and timelessly classic checked pattern offers the perfect fit with a contemporary design. T-Shirt Clothes Price : 10 This is a great T-shirt it can be used in summer or winter it is pure silk with inside wool Recents Featured Most Viewed	Recent products for male user with 27 years old	 Carrier 3:13 AM Market Skinny Rigid Jeans Pants Price : 20 Snap-button closure in sizes 5-8; regular button closure in sizes 9-18. Sits low on waist. Fitted through hip and thigh. Skinny leg. Uniform Jumper Girls Tops Price : 18 Slightly fitted through body. Jumper hits above knee. T-Shirt Clothes Price : 10 This is a great T-shirt it can be used in summer or winter it is pure silk with inside wool Recents Featured Most Viewed	Favorits screen for male user with 27 years old
 Carrier 3:17 AM Market Girls Tops:Uniform Jumper SALE 50.0% Uniform Jumper ~~Price: 18 $~~ Price: 9.0 $ 0 8 Slightly fitted through body. Jumper hits above knee. Leave Comment Report Product Add To Cart	Product Details interface for female	 Carrier 3:14 AM Market Pants:Skinny Rigid Jeans SALE 50.0% Skinny Rigid Jeans ~~Price: 20 $~~ Price: 10.0 $ 0 Be first Snap-button closure in sizes 5-8; regular button closure in sizes 9-18. Sits low on waist. Fitted through hip and thigh. Skinny leg. Leave Comment Add To Cart	Product Deatials interface for male

 Carrier 3:14 AM Girls Tops:AMAYRA NAVY BLUE PRINTE... AMAYRA NAVY BLUE PRINTED COLD SHOULDER LONG SLEEVES POLKA DOTES TOP Price: 450 $ 0 Be first Very Smooth Fabric Blue colour Tops Fitted for all occassions Cold Shoulder 3/4th Sleeve Leave Comment Report Product Add To Cart	Product Details (without sale) interface for male user	 Carrier 3:19 AM Back Skirts:Faux-Suede Utility A-Line Mini Faux-Suede Utility A-Line Mini Price: 30 $ 0 Be first Sits below waist. Relaxed through hip and thigh. Mini hits mid-thigh. Model is approximately 5'9" and wears a size S (numeric size 4). Leave Comment Report Product Add To Cart	Product Details (without sale) interface for female user
 Carrier 3:19 AM Market Logo-Graphic Crew-Neck Tee Girls Tops Price : 11 Slightly fitted through body. Tee hits below waist. Uniform Jumper Girls Tops Price : 18 Slightly fitted through body. Jumper hits above knee. AMAYRA NAVY BLUE PRINTED COLD SHOULDER LONG SLEEVES POLKA DOTES TOP Girls Tops Price : 450 Very Smooth Fabric Blue colour Tops Fitted for all occassions Cold Shoulder 3/4th Sleeve Recents Featured Most Viewed	Recent Product interface for unknown user model	 Carrier 3:28 AM Girls Tops:AMAYRA NAVY BLUE PRINTE... AMAYRA NAVY BLUE PRINTED COLD SHOULDER LONG SLEEVES POLKA DOTES TOP Price: 450 $ 0 2 Very Smooth Fabric Blue colour Tops Fitted for all occassions Cold Shoulder 3/4th Sleeve Add To Cart	Product Details interface for unknown user model
 Carrier 3:15 AM Write Comment This is great Item Done "Item" Items Itemized q w e r t y u i o p a s d f g h j k l z x c v b n m 123 space return	Comment interface for male user	 Carrier 3:17 AM Girls Tops:Uniform Jumper Write Comment Submit	Comment interface for female user

Appendix B Questionnaire Forms (English)

Islamic University – Gaza

Deanery of Graduate Studies

Faculty of Information Technology

Thesis name : Framework for Adapting Mobile Apps User Interfaces

Researcher : Ahmed A. Qazzaz

Criteria

User Age : ______

User Gender : ☐ Male ☐ Female

What Information did you tell the app about yourself : ☐ Age ☐ Gender

How long you usually use the app : ____ hours / day , for ____ days

This questionnaire is meant to measure the users' acceptance and satisfaction about the actions taken by the framework to match their desires and needs while they are using the app.

This questionnaire will be delivered only to the users who did use the app in both cases (Non-Adaptive, and Adaptive), for at least two weeks at each case.

The test bed app "Basket" is an e-commerce app created only for study purposes. The app allows users to check groups of products categorized by several factors. The user can create an account, and interact with sellers and other users through the app.

Non-Adaptive App Case:

	Question	Very Bad	Bad	Acceptable	Good	Very Good
1	Colors of the app in general					
2	Colors of the Interface background					
3	Colors of the buttons					
4	Colors of labels text					
5	Style of app icons					
6	Size of fonts and buttons					
7	Curved vs. Squared corners of cards					
8	Design of the products list screen					
9	Design of the product details screen					
10	Icons used in the user profile interface					
11	Icons of the registration screen					
12	Location of product reviews					
13	Design of the “Submit Review” screen					
14	Position of “Rate”, and “Like” buttons					
15	Position of the “report product” button					
16	Position of the “add to cart” button					
17	Position of the “profile” button					
18	Position of “cart” button					
19	Showing the Recent product as default screen					

Adaptive App Case:

The Adaptive Apps idea

	Question	Yes	No	Maybe
1	The idea is novel			
2	Adaptation of styles is useful			
3	Adaptation is effective and produce satisfaction			
4	Did enjoy using the app			
5	Will use adaptive apps if exists			
6	The adaptive app meet the needs and desires			

Relative to the non-adaptive case, to what extent are you satisfied with how the app:

	Question	Very Bad	Bad	Accep table	Good	Very Good
1	Changed the color					
2	Changed the icons					
3	Changed elements positions					
4	Changed style of elements					
5	Changed on presentation sequence					

Technical Adaptive App Questions

Answer these question after login, or register

	Question	Very Bad	Bad	Accept able	Good	Very Good
1	Colors of the app in general					
2	Colors of the Interface background					
3	Colors of the buttons					
4	Colors of labels text					
5	Style of app icons					

	Question	Very Bad	Bad	Accep table	Good	Very Good
1	The Addition of "Add to Cart" button to the " Products List " screen					
2	The Addition of "Like" and "Rate" buttons to the "Products List" screen					
3	Changing the default screen when app run, based on your most viewed screen					
4	Presetting the quantity for product in the cast based on previous actions					

Appendix C Questionnaire Forms (Arabic)

الجامعة الإسلامية - غزة
عمادة الدراسات العليا
كلية تكنولوجيا المعلومات

اسم الاطروحة : ملاءمة واجهات المستخدم في تطبيقات الأجهزة المحمولة

اسم الباحث : أحمد عدنان القزاز

معايير البحث

العمر : ـــــــ

النوع : □ ذكر □ أنثى

المعلومات التي قمت بتغذيتها للتطبيق : □ العمر □ النوع

المدة التي كنت تستخدم بها التطبيق : ــــــ ساعة باليوم / ــــــ أيام

هذا الاستبيان يهدف إلى قياس مدى تقبل ورضى المستخدمين للتغيرات التي يحدثها التطبيق لملائمة متطلباتهم ورغباتهم أثناء استخدام التطبيق

سيم توزيع هذا الاستبيان على المستخدمين الذين استخدموا كلا التطبيقين (العادي، التطبيق الذي يتلائم مع المستخدم)، لمدة اسبوعين لكل واحد منهما

تطبيق الإختبار "سلة" هو عبارة عن تطبيق متجر إلكتروني، صمم خصيصا لأغراض الدراسة. وهو مبني على فكرة عرض المنتجات المصنفة بناء على عدة معايير. يسمح التطبيق للمستخدم بإنشاء حساب خاص به على المتجر، ويمكنه من خلاله التفاعل مع المستخدمين الآخرين والبائعين.

التطبيق العادي (الذي لا يتغير بتغير المستخدمين)

	السؤال	سيء جدا	سيء	مقبول	جيد	جيد جدا
1	ألوان التطبيق بشكل عام					
2	ألوان الخلفيات للواجهات في التطبيق					
3	ألوان الأزرار في التطبيق					
4	لون النص المستخدم في التطبيق					
5	نمط الأيقونات المستخدمة في التطبيق					
6	حجم الخط، وحجم الأزرار في التطبيق					
7	الحواف الدائرية مقابل الحواف الحادة					
8	تصميم واجهة ” قائمة المنتجات ”					
9	تصميم واجهة تفاصيل المنتج					
10	الأيقونات المستخدمة في واجهة ” الملف الشخص“					
11	الأيقونات المستخدمة في واجهة ”التسجيل“					
12	موقع عرض ”تعليقات المستخدمين“ على المنتجات					
13	تصميم واجهة ”إرسال تعليق“					
14	موقع أزرار الإعجاب، والتقييم					
15	موقع زر ”الإبلاغ عن منتج سيء“					
16	موقع زر“ اضف إلى سلة المشتريات”					
17	موقع زر ” الملف الشخصي“					
18	موقع زر ” سلة المشتريات ”					
19	عرض واجهة ” أحدث المنتجات“ دائما عند بدء التطبيق					

التطبيق الذي يتلاءم مع حالات المستخدمين

فكرة ملاءمة التطبيق

	السؤال	نعم	لا	ربما
1	الفكرة ابداعية			
2	التغيرات على الشكل مفيدة			
3	التغيرات تؤدي إلى رضا أكثر			
4	استمتعت في استخدام التطبيق			
5	سوف استخدم التطبيقات المتلائمة في حال تواجدها			
6	التطيقات المتلائمة، تلائم رغباتي ومتطلباتي			

مقارنةً بالتطبيق العادي، إلى أي مدى راضٍ عن التغيرات التي أحدثها التطبيق من ناحية :

	السؤال	سيء جدا	سيء	مقبول	جيد	جيد جدا
1	التغير بالأوان					
2	التغير بالأيقونات					
3	التغير على أماكن العناصر					
4	التغير على أنماط العناصر					
5	التغير في تسلسل سير وعرض التطبيق					

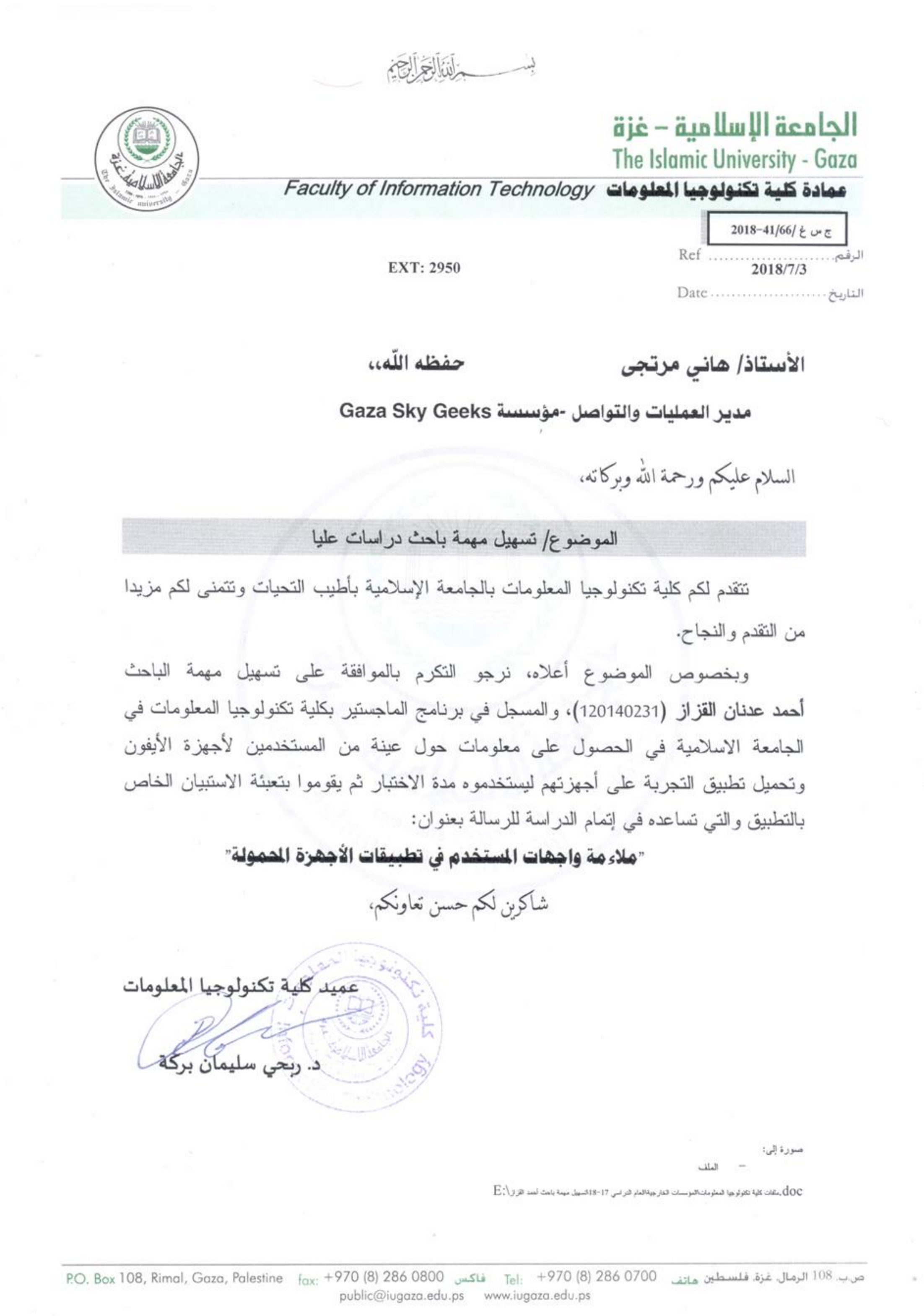
أسئلة تقنية حول التطبيق المتلائم

قم بالإجابة على الأسئلة بعد ان تقوم بتسجيل الدخول أو انشاء حساب جديد

	السؤال	سيء جدا	سيء	مقبول	جيد	جيد جدا
1	ألوان التطبيق بشكل عام					
2	ألوان الخلفيات للواجهات في التطبيق					
3	ألوان الأزرار في التطبيق					
4	لون النص المستخدم في التطبيق					
5	نمط الأيقونات المستخدمة في التطبيق					

	السؤال	سيء جدا	سيء	مقبول	جيد	جيد جدا
1	إضافة زر ”اضف إلى المفضلة“ إلى واجهة ” قائمة المنتجات”					
2	إضافة أزرار ”الإعجاب“ و“التقيم“ على واجهة ”قائمة المنتجات“					
3	تغير الواجهة الرئيسية التي تظهر عند بدء التطبيق، بالواجهة الأكثر استخداما من قبل المتسخدم					
4	وضع قيمة مسبقة لكمية المنتجات في سلة المشتريات بناء على الكميات التي حددها المستخدم في مرات سابقة					

Appendix D Framework Official Communication Letter



بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza
Faculty of Information Technology عمادة كلية تكنولوجيا المعلومات

ج س غ /66/41-2018

الرقم........................ Ref

2018/7/3

التاريخ........................ Date

EXT: 2950

الأستاذ/ هاني مرتجى حفظه الله،،

مدير العمليات والتواصل -مؤسسة Gaza Sky Geeks

السلام عليكم ورحمة الله وبركاته،

الموضوع/ تسهيل مهمة باحث دراسات عليا

تتقدم لكم كلية تكنولوجيا المعلومات بالجامعة الإسلامية بأطيب التحيات وتتمنى لكم مزيدا من التقدم والنجاح.

وبخصوص الموضوع أعلاه، نرجو التكرم بالموافقة على تسهيل مهمة الباحث **أحمد عدنان القزاز** (120140231)، والمسجل في برنامج الماجستير بكلية تكنولوجيا المعلومات في الجامعة الاسلامية في الحصول على معلومات حول عينة من المستخدمين لأجهزة الأيفون وتحميل تطبيق التجربة على أجهزتهم ليستخدموه مدة الاختبار ثم يقوموا بتعبئة الاستبيان الخاص بالتطبيق والتي تساعده في إتمام الدراسة للرسالة بعنوان:

"ملاءمة واجهات المستخدم في تطبيقات الأجهزة المحمولة"

شاكرين لكم حسن تعاونكم،

عميد كلية تكنولوجيا المعلومات

د. ربحي سليمان بركة

كلية تكنولوجيا المعلومات
Faculty of Information Technology
الجامعة الإسلامية غزة

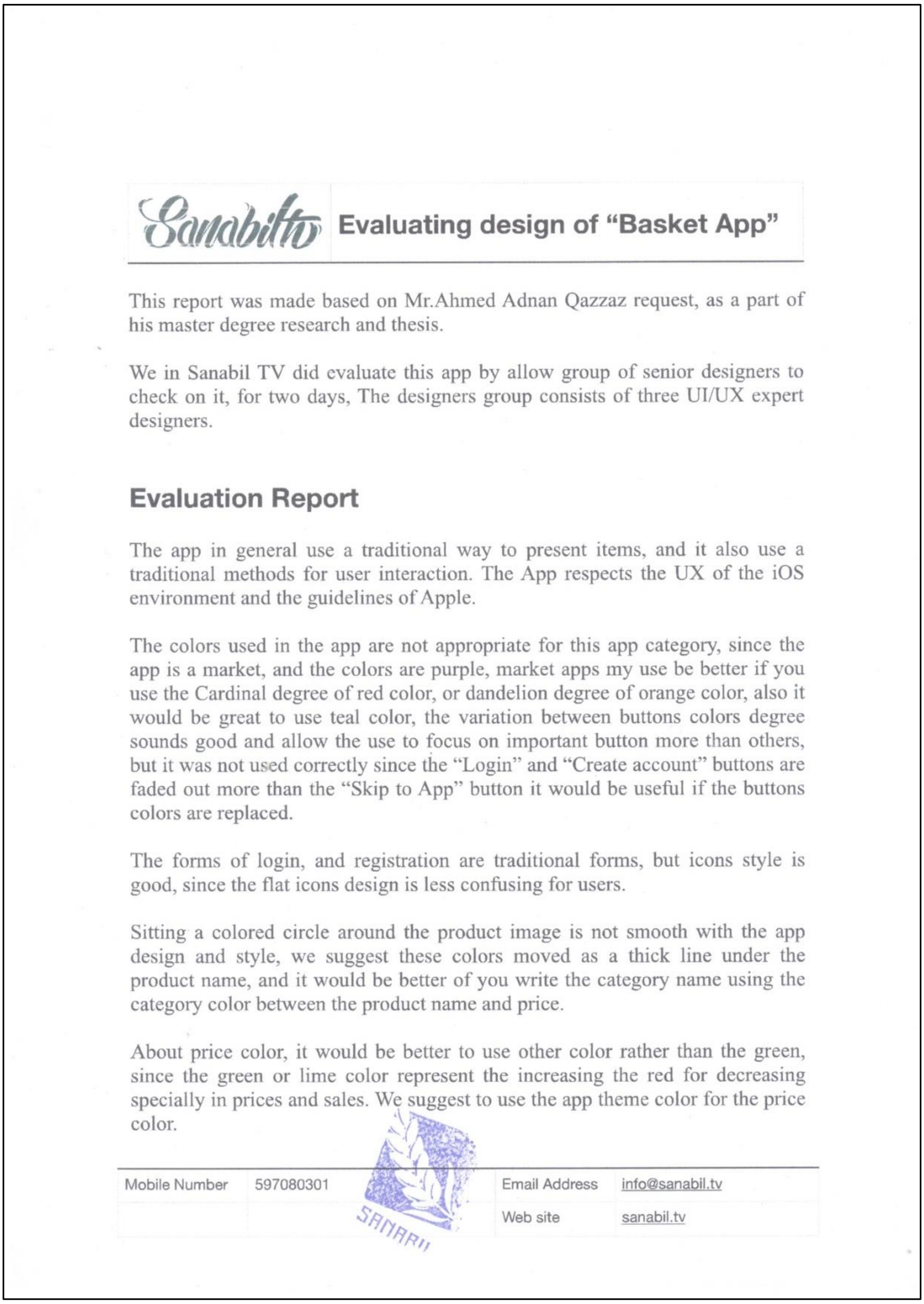
صورة إلى:

- الملف

E:\ملفات كلية تكنولوجيا المعلومات\المؤسسات الخارجية\العام الدراسي 17-18\تسهيل مهمة باحث أحمد القزاز.doc

ص.ب 108 الرمال، غزة، فلسطين هاتف Tel: +970 (8) 286 0700 فاكس fax: +970 (8) 286 0800 P.O. Box 108, Rimal, Gaza, Palestine
public@iugaza.edu.ps www.iugaza.edu.ps

Appendix E Sanabil TV Report



Evaluating design of “Basket App”

This report was made based on Mr.Ahmed Adnan Qazzaz request, as a part of his master degree research and thesis.

We in Sanabil TV did evaluate this app by allow group of senior designers to check on it, for two days, The designers group consists of three UI/UX expert designers.

Evaluation Report

The app in general use a traditional way to present items, and it also use a traditional methods for user interaction. The App respects the UX of the iOS environment and the guidelines of Apple.

The colors used in the app are not appropriate for this app category, since the app is a market, and the colors are purple, market apps my use be better if you use the Cardinal degree of red color, or dandelion degree of orange color, also it would be great to use teal color, the variation between buttons colors degree sounds good and allow the use to focus on important button more than others, but it was not used correctly since the “Login” and “Create account” buttons are faded out more than the “Skip to App” button it would be useful if the buttons colors are replaced.

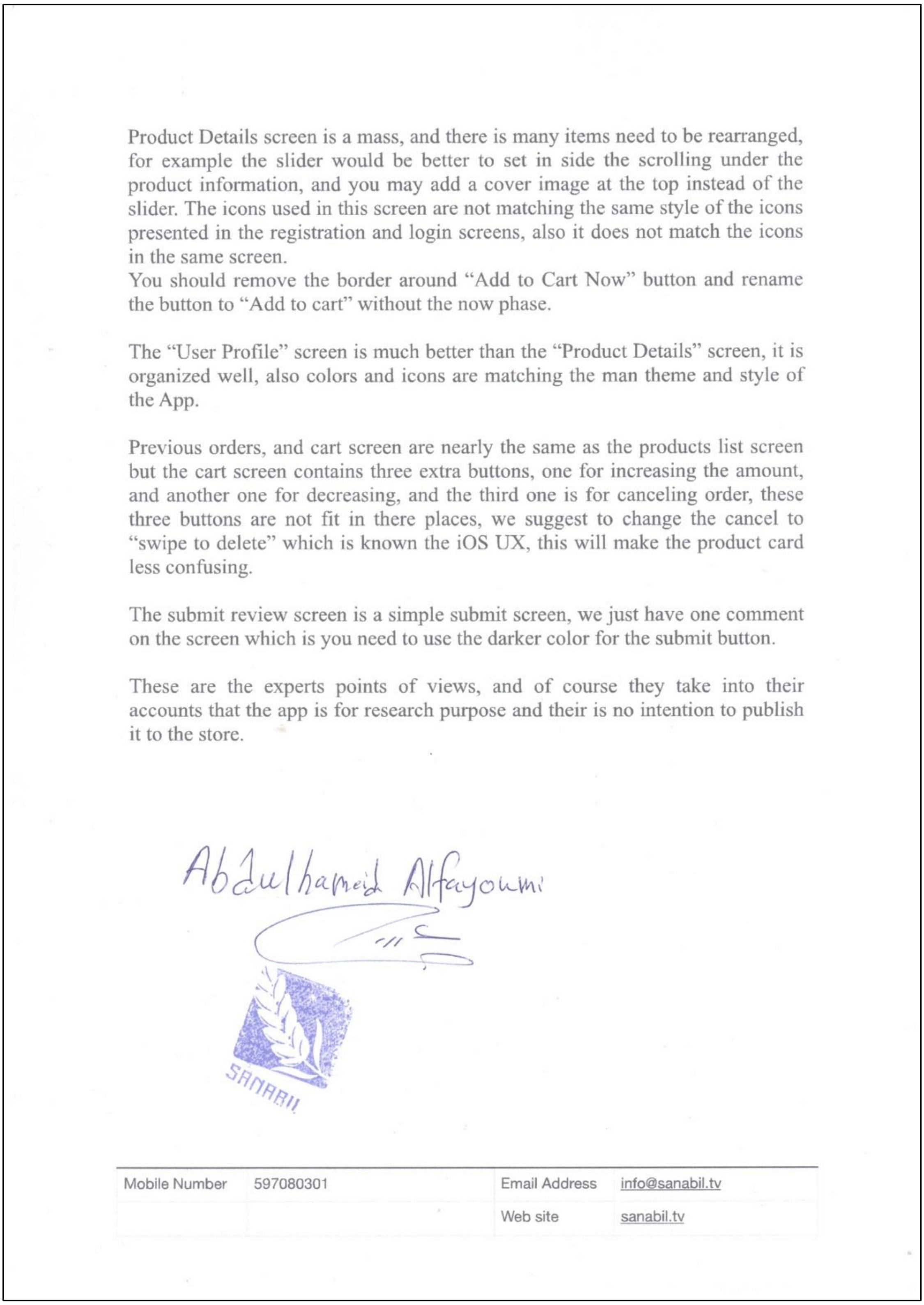
The forms of login, and registration are traditional forms, but icons style is good, since the flat icons design is less confusing for users.

Sitting a colored circle around the product image is not smooth with the app design and style, we suggest these colors moved as a thick line under the product name, and it would be better of you write the category name using the category color between the product name and price.

About price color, it would be better to use other color rather than the green, since the green or lime color represent the increasing the red for decreasing specially in prices and sales. We suggest to use the app theme color for the price color.

SANABIL

Mobile Number	597080301	Email Address	info@sanabil.tv
		Web site	sanabil.tv



Product Details screen is a mass, and there is many items need to be rearranged, for example the slider would be better to set in side the scrolling under the product information, and you may add a cover image at the top instead of the slider. The icons used in this screen are not matching the same style of the icons presented in the registration and login screens, also it does not match the icons in the same screen.
You should remove the border around "Add to Cart Now" button and rename the button to "Add to cart" without the now phase.

The "User Profile" screen is much better than the "Product Details" screen, it is organized well, also colors and icons are matching the man theme and style of the App.

Previous orders, and cart screen are nearly the same as the products list screen but the cart screen contains three extra buttons, one for increasing the amount, and another one for decreasing, and the third one is for canceling order, these three buttons are not fit in there places, we suggest to change the cancel to "swipe to delete" which is known the iOS UX, this will make the product card less confusing.

The submit review screen is a simple submit screen, we just have one comment on the screen which is you need to use the darker color for the submit button.

These are the experts points of views, and of course they take into their accounts that the app is for research purpose and their is no intention to publish it to the store.

Abdulhamid Alfayoumi

SANABIL

Mobile Number	597080301	Email Address	info@sanabil.tv
		Web site	sanabil.tv