

WEB AND SOFTWARE ENGINEERING THE FACEBOOK WAY - AN UNDERGRADUATE MINI PROJECT

Andrew K. Smith
University of Sunderland
Department of Computing, Engineering & Technology
David Goldman Informatics Centre, St Peters Campus, Sunderland
<http://nestor.sunderland.ac.uk/~cs0asm/home>

ABSTRACT

The following paper discusses the experiences of a seven week mini project run twice at the University of Sunderland, designed to teach applied web programming using the Facebook platform and makes the case for the use of Facebook as a suitable platform to teach web programming. First year undergraduate computing students used a broad range of technologies underpinned by a suitable software engineering methodology to design, develop and publish applications to the Facebook Application Directory.

Students are able to market and deploy applications easily and benefit from feedback from peers within a familiar, student friendly environment. There is an opportunity to engage with a wider audience and opportunities to develop and refine software development skills whilst consolidating their learning in a challenging project that is accessible to anyone with a Facebook account.

Keywords

Software development, software engineering, web programming, social network, programming, feedback

1. INTRODUCTION

Since 2009 the Department of Computing, Engineering and Technology at the University of Sunderland has offered all level one undergraduate computing students, the opportunity to complete an individual mini project. The module accounts for 40% of their final level one mark and was designed to offer some choice to students in an area they were interested in. (Devlin, 2011)

One of the five projects on offer in the last two years has been the development of a Facebook application. Students were required to demonstrate competency in browser based programming using a variety of technologies which leveraged the Facebook Application Programming Interface (API).

One of the overarching aims of the mini project was to provide students with an experience similar to that of the final year project early in the programme and pull together the various strands of teaching. Students arrive at the mini project with a grounding in computer science, having studied modules in programming, database design, software engineering, architectures, networking and systems administration. The challenge for any mini project is to use these new skills, provide a project of suitable difficulty and provide a framework to allow the student to explore new ideas, technologies and approaches, whilst remaining achievable for a level one student. (Connolly *et al*, 2008)

1.1 Why use Facebook?

Some of the principal motivations for designing and running the Facebook project were to provide students with an opportunity to practice key software development skills, develop applications for real users and engage in live user testing, whilst working to the requirements of the Facebook technical documentation.

It was also hoped that Facebook could be used as a suitable environment because of its popularity and strong brand within the student community (Mazman & Yasman 2010). It was felt that by using a familiar tool, something that students felt comfortable with using and combining it with a programming activity it could improve engagement, keep students motivated and enthuse development.

Finally, it was hoped that the environment could be used to build meaningful relationships between the developers and end users (Ellison *et al*, 2007) as well as motivating students to build better, high quality applications.

2. OPERATION

For the last two years the structure has remained the same, with students attending a series of formal lectures and hands-on labs which demonstrated the core elements of the Facebook API. Lectures were kept to a minimum and were delivered with live examples that could be downloaded as starting points for individual projects. Table 1 summarises the schedule and key deliverables of the project.

Wk	Lecture topics	Assessments
1	Overview of mini-project / PHP re-visited Facebook core concepts Developer documentation The Facebook Graph API Authentication	Mini Project Task 1 – (worth 5%) handed out (Research into Facebook app success – list 3 apps you consider to be successful – explain why – detail the technologies you think they use – submitted on Powerpoint)
2	API data and PHP complex data structures Making a Mashup Linking to a database in Facebook Review the top 3 applications on Facebook	FB Mini Project Task 1 – hand in FB Mini Project Task 2 – (worth 5%) hand out (App that lists your friends and shows their photo)
3	Social Plugins and Marketing your app (e.g. Quiz and wall posts) Developing your own ideas More on Social Plug-ins and Marketing your app	FB Mini Project Task 2 – hand in – DEMO TO TUTORS FB Mini Project Task 3 – hand –out worth 10% (Define the scope and functional requirements of your app, storyboards, define the technologies you will use, marketing plan)
4	Individual student meeting to review and assess Mini-Task 3	FB Mini Project Task 3 – Mini-Project reviews FB Mini Project Task 4 - Develop your Facebook App – worth 60%
5-6	No formal lectures - tutorials to help support development / mini-project progress	
7	FB Mini Project Task 4 hand-in FB Mini Project Task 5 Demo and Presentations – worth 20%	

Table 1 - Mini project schedule and assessment deliverables (2010/11)

The introductory week started with the core Facebook concepts. These sessions introduced students to the building blocks of the Facebook platform and explored user profiles, links between users, the Facebook open graph and elements of data mining before reviewing authentication options and security. Students were then introduced to the developer administration area, which would be used to manage the build and given recommendations for further reading. (Figure 1)

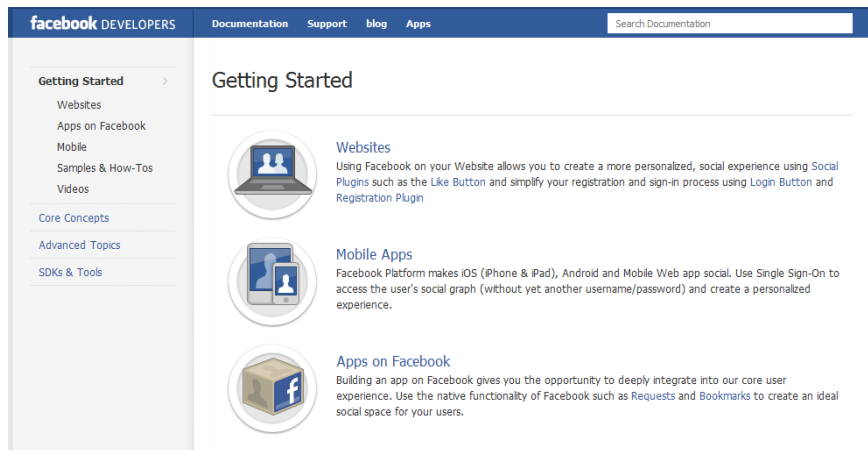


Figure 1 - Getting started documentation in Facebook wiki

During the second week, students were introduced to data exchange formats and reviewed models for working with Facebook data including eXtensible Markup Language (XML) and JavaScript Object Notation (JSON). Walk through examples and scenarios were illustrated and a break out activity was used to get participants thinking about why such data structures are required and how they're used in modern web applications. The session concluded with a review of database integration and storage options within the platform.

The final lecture sessions reviewed the important activities associated with marketing applications and the increasing use of social widgets, which provide an easy and efficient way of adding Facebook features to web pages, without writing bespoke code. Sessions also reviewed the diversity of applications on offer and explored the commonality between implementations, both from a technical and design perspective. Some practical sessions were delivered in which staff provided a simplified approach for the development of a web based application, using the students' previous experience of programming and software engineering. The aim here was to pull together all of the new material covered so far, and link this with the students previous experience of software engineering and core programming.

2.1 Assessment

The project was assessed by a portfolio comprising of four key tasks, each designed to demonstrate understanding and application of the technology. (Devlin, 2010) The first portfolio task required students to evaluate existing Facebook applications, reviewing the target audience, requirements, technology and how successful they were in the marketplace. This was submitted as a report and formal presentation.

The second task required the development of a simple Facebook application that displayed a list of users friends, by making relevant calls to the API. Marks were allocated for code, design, layout and use of API features.

The third portfolio submission required students to submit an initial proposal for their application; this consisted of a written specification, some prototype designs, information about the target audience and the technologies that would be used in the build. This was presented formally to staff, which gave feedback and confirmed suitability forming the terms of reference for the project. This in part was similar to the process final year students would undertake in the completion of the dissertation and final year computing project, to 'systematically and methodically develop a software artefact' and write-up clearly the steps involved.

This was the first time students were expected to design their own application and express this in a formal document. Whilst previous assessments provided realistic scenarios and client requirements, there was little opportunity to obtain 'real' and immediate feedback during the testing stages of the development life cycle. The Facebook framework by its very nature, afford the developer the opportunity to publish early builds of applications to friends, via the 'Wall'. End users were able to test applications quickly with little effort and provide immediate feedback via comments, messages and test polls.

Some students used Facebook status updates and set-up groups to generate ideas, discuss initial thoughts and get feedback before starting on their project. This was encouraging and provided some evidence of exploratory research, establishing user goals and requirements before diving into the code, which has been a challenge in previous assessments.

Participants were encouraged to release a final version of their application and 'mark it' for inclusion in the Facebook Application Directory several days before the final hand-in, with the aim of marketing the application and attempt to increase active users. Students were advised to provide meaningful information about their application on their application profile page, with high quality support documentation, graphics and respond to feedback promptly. A copy of the usage statistics were requested as part of the final report, which in some cases showed hundreds of active users, many of which were not connected to the student. (figure 2)

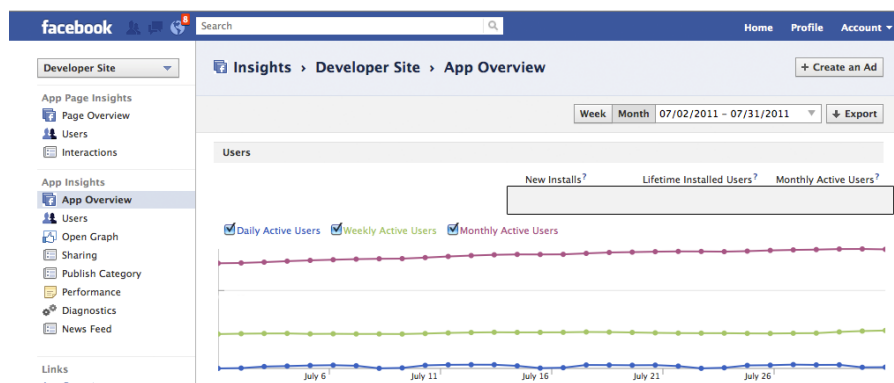


Figure 2 - Example of Dashboard screen, developer.facebook.com

Because of the evolving nature of the framework students were also required to submit a video screen capture of their application with audio or text subtitles that explained the main functionality. It was hoped this would help students preserve a copy of the application for their portfolios; personal development plans (PDPs) and follows the work from a previous exercise in screen-casting, to teach multimedia to level one students, which was well received. (Smith, 2010)

2.2 Feedback

The department is keen to ensure students receive timely and appropriate feedback during their studies, due in part to the success of previous projects where one-to-one feedback has occurred. Students were allocated time with a team member and staff operated help and feedback sessions throughout the project.

3. RESULTS

Work was of a high quality and demonstrated many of the features found in popular and successful Facebook applications. A mixture of game, entertainment and utility applications were submitted. These varied in programming complexity, core functionality, Facebook integration and the visual interface, with some students using un-styled HTML to create the front-end (figure 3) whilst others used Adobe Flash (figure 4) or a combination of HTML, CSS and Flash to achieve highly graphical experiences. (figure 5)

3.1 Platform integration

Students integrated their applications in a variety of ways, again with varying levels of complexity. Simpler applications made use of session options to check that a valid Facebook user was signed-in, in order to run, whilst others requested more permissions, such as the ability to post to the users wall (figure 6), or access the users profile information and photo library. Indeed, it is these advanced features and potential integration points that make Facebook a useful framework for a level one undergraduate computing student.

It was observed that students usually allocated full rights to an application during development, which in effect, gave the application full access to their entire account. Many students were surprised at what permissions

other applications they had installed, had to their data. Many students commented on this in their final submission.

Some students experimented with the API and wrote server-side scripts that mined information from friends and presented it in a more meaningful manner and its hoped that these types of applications can be refined into suitable final year projects.

3.2 Links with users

Status updates and messaging was used heavily to promote applications, but also used to communicate with end users and obtain real, immediate and useful feedback that assisted in the development. Getting feedback from users about their applications informally in this way was seen in a positive light. It was observed that students responded to the feedback given, reflected and often made changes to their applications because of it, although this wasn't always documented. Staff also noted the discussion around general themes including programming, web tools and error messages, which in part reflects some of the work undertaken by Lampe et al (2008) in the changing perceptions of Facebook and its long term use.



Figure 3 - Example of un-styled HTML interface

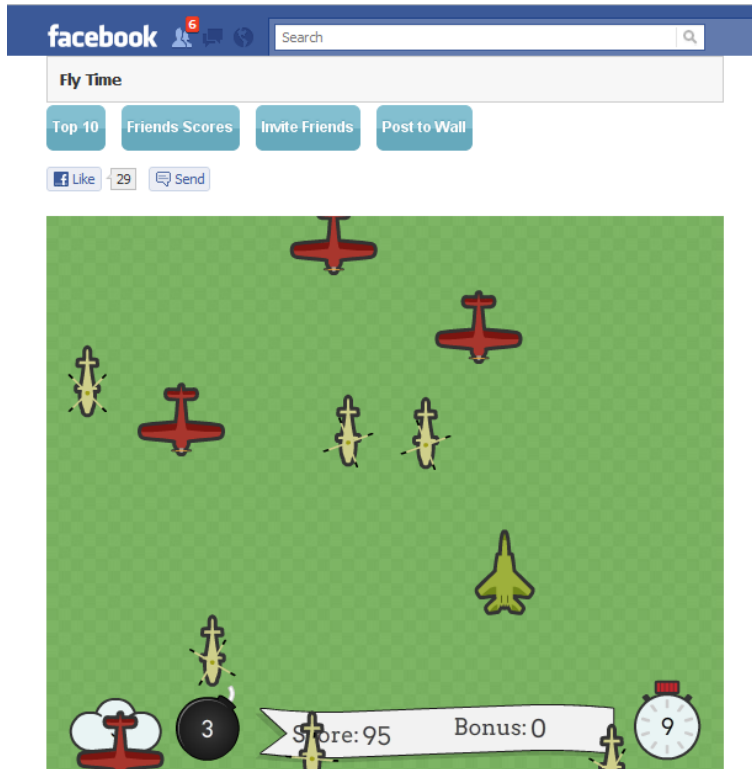


Figure 4 - Example Flash interface

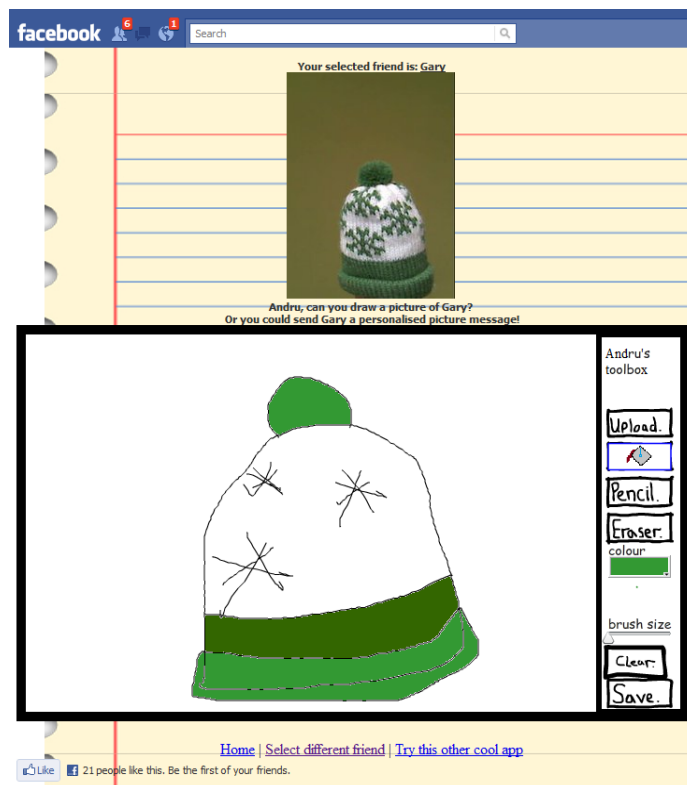


Figure 5 – HTML and Flash styled interface

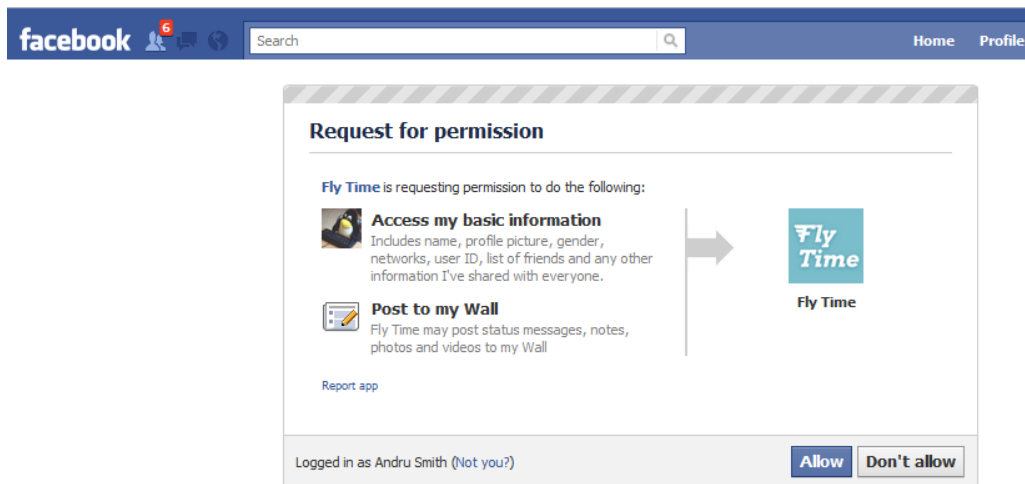


Figure 6 – User dialogue, request for permission

4. ANALYSIS OF RESULTS

4.1 Agility

Whilst one of the strengths of this mini project was that it wasn't restrictive and allowed the students flexibility in their choice of technology and direction, it was one of the greatest challenges for academic staff. With a great variety of projects, a range of potential technologies and a mixture of abilities, staff had to be able to respond in a flexible and supportive manner to the many challenges, some of which included programming in multiple languages, handling complex error messages and help in the use of third party APIs, some which were new to the staff. Both students and staff learned new things in the project.

4.2 Rate of change

The web is fast paced and over the last 2 years there have been significant changes in the Facebook platform, resulting in many of the textbooks and resources used by students in 2009/10 to become deprecated and unsupported. Indeed, many of the popular ways of creating interactive sites in 2009 using FaceBook Markup Language (FBML), a sort of hybrid tag based language similar to HTML have been deprecated with the move to using more widgets and an increased reliance on javascript, to support more intuitive and responsive applications without page refreshes. Whilst looking for tutorials and reviewing documentation it was clear that many students were unaware of the issues with using older code, often undocumented in the new architecture. For some this meant refactoring or re-engineering the code to do what they required, for others it meant simplifying their applications or making minor modifications to the design to meet the timescales.

Javascript for example, was demonstrated in lectures and students were encouraged to experiment, but were advised to leverage their existing skill base using PHP, HTML, Flash, AS3 and CSS in the first instance. The exposure to a new language and often technical documentation gave the students a new experience of what it was like to develop an application using emerging technologies and approaches, something the mini project aimed to achieve when it was created.

Another example of change was seen in the first presentation of the project in 2009/10. During the second week of teaching and during the tutorial sessions, the Facebook conference was taking place in San-Francisco during which the new Facebook API was announced, this would become the recommended way to develop applications. At launch the documentation was brief and students were faced with a partially working 'new platform'. Older approaches were deprecated and the 'testing console', an area used to try out new fragments of code was removed from the developer site. Students had to find new ways of testing and building their applications and staff had to find new ways of teaching it and explaining why it had happened.

Academics wanting to write code once and regurgitate should steer away from such practices as the development of a Facebook application, or indeed any other multimedia application that relies on the use of a rapidly changing third party API such as Facebook demonstrate. Much work was required by academic staff to refresh the notes and provide examples for the 2010/11 presentation of the project using the new Facebook API. This gave the opportunity to teach about source management and release issues not normally discussed in a mini project at this level.

4.3 Familiarity

Student familiarity with the Facebook platform as users, has helped bridge the development gap often seen in other areas of traditional software development. Where students not only need to become familiar with the various programming languages, technical documentation and plethora of computing jargon they also need to become familiar with the development platform as a whole, learning to use new tools and software often with unintuitive interfaces. As experienced Facebook users, students found no major problems getting to grips with the developer interface and found the process intuitive.

Another motivation for using the Facebook platform as a suitable mini project, was the almost universal use by students to communicate with friends. This is not surprising as many studies have shown the growth of the platform, particularly in the further and higher education sector since its development in 2001. (Mazman et al, 2010) Of the year as a whole, over 95% had a Facebook account and of the mini project participants over the two years, only one student did not have an account prior to starting development.

In all cases, students that had Facebook accounts, were familiar with the concept of Facebook applications and the core elements of the platform, albeit from a users perspective. Features such as wall posts, messaging, pokes, status updates, geo-tagging etc didn't require detailed explanation. This was seen to be advantageous, as not only did students feel comfortable about the technology, they were emotionally positive towards it. Informal feedback seemed to suggest students desire to create something 'cool' that could be accessed by friends was a big motivator in their development efforts.

4.4 Quality

Tutors reported genuine interest from students on the success of their applications and the positive feedback from seeing their work accessed by others and kudos from peers. The usage statistics gave the student developer a raw metric for how successful their application was and who had been using it. The initial aim was to get a few dozen users, but many students aspired to attract more installations and considered ways to attract more users. Like professional developers and companies creating web applications, they started to ask what they could do to improve, was the quality sufficient, did it meet the requirements and how could they improve the marketing. In the same way that a hit counter was often used in the early days of web development to show how popular a site was, the usage counter in the Facebook dashboard gave the users something to aim for, albeit a rather blunt statistic.

4.5 Feedback

Many of the end-users of applications were not computing students, unfamiliar with the development framework and programming. This was seen as a positive and provided the opportunity to obtain feedback from different groups of users and not just university friends. Students were surprised at how quickly even the simplest applications were spread, following a status update.

4.6 Student perspective

It was felt by the students that the project was worthwhile and sufficiently challenging for a level one mini project. Students have commented on the flexibility the platform offers and the approach taken by the department in front-loading the lectures with support tutorials to help in the development stages. Whilst issues relating to lack of documentation during the build cause some problems, it was seen in a positive light and students gained valuable real world experience of working with a changing and emerging platform. Indeed some students have since approached staff about monetising their applications and are currently exploring advertising and micro payment options within the platform.

5. CONCLUSION

The student feedback has been positive and the project can be considered a success in terms of meeting its original objectives. The extra effort required to leverage the Facebook API was worthwhile and enabled undergraduate computing students to demonstrate their creativity, analytical and programming skills to deliver

complex web applications in their first year. We believe as a mini project it succeeded in challenging students and consolidated the level one as a whole, whilst ensuring students were motivated and enthusiastic about the possibilities open to them in future years.

The project utilised the social and community strengths of Facebook to obtain feedback from users and facilitated a way of releasing updates and software early to users. The project addressed a topic that is being actively debated in many institutions about how problem based, rather than topic based approaches to teaching and learning are relevant to the computing science curriculum.

In terms of future work, there are multiple facets of the academic experience, web engineering, user testing and feedback that resonate and we're seeking to collect data on these for future projects and publications.

The department intends to run the project again in the 2011/12 academic year where there will no doubt be new challenges and exciting opportunities for both staff and students.

6. REFERENCES

Connolly, C., Murphy, E., Moore, S., (2008), *Programming Anxiety Amongst Computing Students—A Key in the Retention Debate?*, Volume 52 (1), IEEE Transactions on Education, Department of Computing, Dundalk, Institute of Technology, Dundalk, pp52-56.

Devlin, S. L, (2011), *You Can't Solve A Jigsaw With Only One Piece – Why Cross-Topic Lta Is Key To Student Engagement*, In *11th Annual Conference of the Subject Centre for Information and Computer Sciences*, edited by Higher Education Academy ICS Subject Centre, Co. Antrim. UK.

Ellison, N. B., Steinfield, C., & Lampe, C. (2007), *The benefits of Facebook "friends:" Social capital and college students' use of online social network sites*, Journal of Computer-Mediated Communication, 12(4), article 1, Indiana, USA.

Kirschner, P. A, Karpinskib, A., C, (2010), *Facebook and academic performance*, Computers in Human Behavior, Volume 26 (6), pp1237-1245.

Lampe, C., Ellison, N. B., Steinfield, C., (2008), *Changes in use and perception of Facebook*, In Proceedings of the 2008 ACM conference on Computer supported cooperative work (CSCW '08), pp721-730, ACM, New York, NY, USA,.

Mazman, S. G., Usluela, Y. K., (2010), *Modeling educational usage of Facebook*, Computers & Education, Volume 55 (2), pp444-453.

Smith, A. K, (2010), *Exploring the use of screen capture software and multimedia to include diverse groups of learners*, Poster session Ref No.42 In *3 Rivers Consortium Conference*, 3Rivers Consortium, University of Sunderland.

Copyright of ITALICS: Innovations in Teaching & Learning in Information & Computer Sciences is the property of Higher Education Academy's Subject Centre for Information & Computer Sciences (HEA-ICS) and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.