

# Teaching Facebook@NUS

Ben Leong  
Department of Computer Science  
School of Computing  
National University of Singapore

31 July 2008

## Abstract

We recently created and conducted a new course at NUS called *Software Engineering for Evolving Platforms* in Academic Year 2007/2008, Semester 2. In this report, we explain how the course came about, describe the teaching objectives we had hoped to achieve, how we conducted the class and how things turned out, and briefly discuss the lessons learnt. CS3216 seeks not so much to teach specific skills, but to ignite in the students a sense of passion, and to instill the confidence that they are capable of dreaming *and* achieving their dreams. While a number of students developed impressive applications, we do not claim any credit for their success. We believe that we merely provided these students with an environment conducive for them to learn what they needed to learn in order to do the right thing. While the final feedback for the class has generally been positive, there is room for improvement and some significant changes will be made to the course, i.e. the workload will be reviewed and reduced in future offerings.

## 1 Introduction

We recently created and conducted a new course at NUS called *Software Engineering for Evolving Platforms* in Academic Year 2007/2008, Semester 2. Within the School of Computing, we call it *Facebook Programming@NUS*, because Facebook, the new social networking platform, is used as the medium of instruction.

Unlike other courses at NUS, places for CS3216 were not allocated through the usual CORS bidding process. Interested students were required to submit a personal statement and places were allocated in a manner as to ensure that there was the diversity of talent required to build good Facebook applications. About 80+ applications were received and eventually 55 students were offered places for the initial offering of the course.

In this report, we explain how the course came about, describe the teaching objectives we had hoped to achieve, how we conducted the class and how things turned out. We also discuss the lessons learnt.

## 1.1 The Inspiration

Without doubt, the spirit of CS3216 was inspired by “the Last Lecture,” a talk given in September 2007 by the late CMU Professor Randy Pausch [9]. This talk was modeled after an ongoing series of lectures where professors are asked to give a hypothetical final lecture, where they would have to ask themselves “if they had but one last lecture to give before they died, what would it be?”

During his talk, Randy described this course he had created when he first moved to CMU called “Building Virtual Worlds,” where the students learnt to how to make short computer animation clips. If Randy had restricted the enrolment for the course to Computer Science majors, it might perhaps have been “just another course,” but he didn’t. Instead, Randy drew the 50 students for his first course from all departments across CMU.

The course consisted of five projects each lasting two weeks. The students were divided into teams of four for each project and between projects the teams were shuffled so that the students worked with different people each time. The rules for the course were simple: they could do anything they wanted, just no shooting violence and no pornography. As it turned out, the course turned out to be a phenomenal success.

There are two aspects of this course that I found most intriguing: (i) it managed to persuade non-Computer Science students to do Computer Science; and (ii) the students could do anything they wanted. Personally, I found these two ideas to be really cool. I think Computer Science is cool and we should help people from other faculties appreciate why Computer Science is cool.

Randy also spoke briefly about dreams in his talk – and it is my belief that as educators, one of our missions is to help students dream and achieve their dreams. The concept of letting students do whatever they wanted really appealed to me. Why should we want to tell students what to do? How do we encourage them to be creative if we give them assignments for which we already know the answers and to which they only have to come up with the matching answers?

As it turns out, while I am a Computer Scientist, I am not a graphics guy. “Building Virtual Worlds” is not up my alley and there was no way that I could possibly replicate it at NUS even if I wanted to. So “the Last Lecture” was a passing thought – or so I thought.

## 1.2 The Rise of the Social Networks

A sea change has been taking place in the software development landscape in recent years. In the past, a new major programming language is invented perhaps every decade or so. In recent times however, the pace of change has accelerated to a point whereby it is not reasonable for us to attempt to teach students everything that they need to know when they go out into the workforce. Also, many domain-specific languages are emerging quickly (php, Ruby on Rails, C# for webapps; Python for scripting/webapps), making it harder to catch up with the new developments even when one is focused on one of these big domains. We have also been speaking with folks from industry to understand the needs of the market and one core attribute that frequently comes up is the ability for fresh graduates to be able pick up new skills and abilities on-the-fly.

It was towards the end of last year that Facebook applications (apps) started to become popular both locally and globally. And after being Super-Poked by several friends, the *Eureka*

moment came. It occurred to me that I had no idea how to write Facebook applications – but it was also quite clear that because most of the Facebook applications were quite simple, it couldn't possibly be too hard. The fact that Stanford had also started a Facebook programming course further convinced us that the a Facebook programming course might just be one avenue for us to attempt to teach the meta-skill of “learning how to learn.” With this backdrop, we eventually decided to start a new Facebook programming course after extensive discussions within the School.

On the surface, Facebook programming is merely web programming and it might seem odd that we would start a web programming course given we already have other courses teaching the “same” thing. Thus, it is perhaps appropriate to explain why we decided that Facebook is an ideal platform for students to learn how to assimilate and exploit new technologies. There are two key innovations pioneered by Facebook that sets it apart from traditional approaches for software development:

- **Open API for Social Network Applications.** User interactions are possible in traditional applications like discussion forums, and social networks in the form of Friendster and MySpace have been around for several years. The innovation in Facebook is an open API that allows user-created applications to exploit social networks to enable interactions between friends in new ways. A most notably example is that Facebook applications can exploit social networks to spread in a viral way.
- **Ultra-Agile Development Model with Intimate User-Developer Interactions.** Each Facebook application comes with a Facebook-hosted homepage complete with a discussion forum and review system. This allows users to post their reviews, comments and questions, and the developers to respond and act on the feedback from the users.

While similar forums do exist for some Open Source projects, this feature is fully integrated for all applications in Facebook and it allows online communities to be built around applications effortlessly. To be successful, the developers need not only be good programmers, but must now also know how to manage their users.

Finally, the Facebook platform while usable, was relative “unstable” and it was evolving even while we were conducting the class. In fact, the platform also had some known bugs. All in all, it was an ideal platform for students to learn how to cope with a new and *evolving* programming environment.

### 1.3 The Stanford Facebook Course

Stanford University conducted a similar Facebook programming course in Fall 2007 [3]. NUS is probably only the second school in the world to conduct such a course. However, while Facebook is the common platform used by both courses, the two courses are quite different because we have different needs.

The stated goal of the Stanford course is to “create, launch, and optimize web applications” and “focus on how metrics and user feedback can help developers and product managers improve their applications.” On the other hand, our goals are to have students (i) pick up new technology quickly; (ii) learn to work in inter-disciplinary teams; (iii) express their creativity; and (iv) pick up some industry experience and contacts (if possible).

The Stanford course focusses on persuading users to add and use the created applications, i.e. “capturing eyeballs.” In that course, the students create two applications. The first is graded on a “combination of the number of installs and daily active users (with a heavy weighting on active users)”, while the second “emphasizes user engagement and retention and is graded based on a combination of the number of visits, length of visit and/or page views”.

CS3216 is different, or perhaps less focussed. There are many paths to success in life — and this is a core belief for CS3216. Students in CS3216 are allowed to decide for themselves the application they believed was “cool” and build it. At the end of the semester, all they had to do was to persuade the instructors and evaluators of their projects to agree with them! There was no requirement for the students to attempt to create an application that attracts a maximal number of users.

In fact, we believe that it is neither fair nor desirable to grade students on the popularity of their apps. Technically, our students are definitely capable of reproducing some of the million-users apps within a week or two. However, it would be hard for such apps to gain traction given the fatigue and general decline in novelty of Facebook apps.

## **2 Teaching**

The CS3216 has two key teaching components: lectures/seminar and programming assignments. As the emphasis of CS3216 is on “learning by doing,” the latter is the more important component between the two. In addition, we also conducted a small number of workshops at the beginning of the semester.

### **2.1 Lectures & Workshops**

It was decided at the very beginning that we would not be teaching web application programming directly in CS3216. There are several reasons for this: (i) web programming (or at least the level required for Facebook programming) is relatively simple and it is something that good programmers should be expected to be able to pick up the skills relatively quickly; (ii) many of the students already had extensive web programming experience and (iii) there were other courses at NUS that would cover web programming and hence it didn’t seem appropriate for us to repeat the same content. The assumption that as long as we had the right composition of students the web programming elements would take care of itself turned out to be correct.

That said, we did organize several workshops at the beginning of the semester as crash courses for students who had less programming background to help them “level up.” Topics covered included working with UNIX, php/MySQL, html/CSS, Javascript and AJAX. These workshops were conducted by senior undergraduate students who were experts in these topics. Overall, the workshops were well-received, but attendance was relatively poor for two reasons: (i) many students were already well-versed in many of these topics and (ii) there were scheduling conflicts.

One of the difficulties associated with a cross-faculty module is that it is very difficult to find a good time for which conflicts with other classes are minimized. To this end, the classes for CS3216 were held on Monday evenings. This also turned out to be a good arrangement because of the numerous guest lecturers who were invited to share their experiences with the class. Since it was decided that the class time for CS3216 would not be spent teaching the

basics of web programming, it was instead used to expose students to three broad categories of topics.

The first category are topics that are relevant to web application development. These included principles of software engineering, user-centric web applications design, security issues, Ruby on Rails, provisioning for million-user web applications, and a seminar on developing applications on other social networking platforms, i.e. Google OpenSocial [5].

The second category of topics can loosely be classified as “entrepreneurial” activities. These activities typically involve inviting members of the industry to address the class. For example, we invited people who were interested to develop Facebook applications to tell the class about their business and the sorts of applications that they were interested in building. The goal was to allow students the opportunity to build apps in collaboration with the industry. This served two purposes: (i) it ensured that the final projects were more relevant to the real world and industry and (ii) it might allow the students to tap on resources not available within the school. Several partnerships came out of this exercise, including tie ups with MTV Asia and a social enterprise called Gift and Take.

Near the end of the course, we held a plenary session where seven local entrepreneurs were invited to come share their experiences and their views about entrepreneurship. The goal of this session was not so much to promote entrepreneurship among our students but to give them a more complete picture of what entrepreneurship truly entailed, including the risks and pitfalls. Media coverage of entrepreneurship has tended to be overly optimistic and exuberant. The risks and downsides of start ups are downplayed. In this session, the facts were made clear to the students so that they can better understand the options available to them after they graduate. One defining characteristic of entrepreneurs is passion and we were hopeful that some of that passion would also rub off on some of our students.

Finally, CS3216 is also about ideas. While technical competence and skills are critical for the success of Facebook applications and software development, it is important for students to learn how to come up with ideas and to critically evaluate them. To this end, the students were asked to evaluate existing Facebook apps and present their analysis in the form of a student-led seminar. We also had a session just before the term break where the students pitched their ideas for the final projects and critiqued each other’s ideas. For the final project, the students met with the TA and lecturer regularly to discuss not only the progress of the project, but also general ideas related to the project. The final project culminated in a poster session where the students had to “sell” their projects to a panel of evaluators and the SoC community.

The schedule of the classes is shown in Appendix A.

## 2.2 “Learning by Doing”

The programming assignments are the key to learning in the course. In total, we had three programming assignments and a final project.

The programming assignments for this class were carefully designed to equip students with the requisite skills and knowledge to develop and deploy their own Facebook applications. Students were free to choose to do the assignment and projects in a language of their choice, though only PHP and MySQL were supported by the teaching staff. We refer fondly to the three programming assignments as “The Trilogy”:

1. **A New Hello.** For this assignment, students worked in pairs to develop a very simple Facebook application that prints a text message. The purpose of this assignment was to

familiarise students with the Facebook platform and how to deploy a simple web application. It was also intended as an opportunity for peer-learning as we tried to pair the students with limited programming background with the experienced programmers.

2. **The Wall Strikes Back.** In this assignment, students worked in teams of three or four to develop a simple application that allows users to post messages on a Wall. The purpose of this assignment was to let students practise integrating a database into their web applications.
3. **Return of the Cow.** In this assignment, students worked in teams of three or four to develop a simple application that allows users to interact with other users. The exact form of interaction was left to the students. It did not have to be “throwing a cow”. The purpose of this assignment is to let students practise manipulating user data in their applications and to issue invitations.

Students were allowed choose their teammates for the assignments, subject to the constraint that they should not work with the same people twice. For the final project, the students were allowed form teams up to four persons in size without constraints and they could propose any app they wished.

To ensure that the teams did not go off track or choose projects that were too trivial, each team was required to submit a project proposal by the midterm break and the lecturer met with each team before they started work to discuss their project. Then for the duration of the project, the teams would meet with the Teaching Assistant every fortnight to report their progress and to discuss their latest ideas.

## 3 Evaluation

We have to be realistic that when designing a new class, it is not always possible to achieve our intended goals – and to some extent, the process of conducting the new class is also a lesson for the teaching staff in itself. In this section, we evaluate the conduct of CS3216 and discuss the lessons learnt.

### 3.1 Learning Outcomes

We conducted a survey after the midterm break to understand how students perceived the course and to seek suggestions on how the course could be improved. A total of forty-four students responded. A summary of the survey responses is given at Appendix B. The results were quite surprising because the reaction of the students to the course was quite polarized. About 70% of the respondents thought course was good and perhaps the best course they had ever taken at NUS, while 30% felt that the course was mostly painful. None of the respondents chose the middle ground, i.e. that the course “was just like other modules at NUS.” The students either liked the course or they didn’t. Our view is that teaching is not a popularity contest and doing well in such surveys cannot and should not be a goal. Nevertheless, we have to come to terms with the fact that the students’ perceptions of a course do matter. It would be hard to persuade students to learn in a course that they thought was a waste of their time.

We believe that there are several possible reasons for the adverse reactions from the 30% of the respondents who did not like the course: (i) some students did not anticipate the high

workload generated by the course and were not prepared to spend quite a much time on the course as some of their coursemates; (ii) some students did not anticipate the level of competitiveness in the course arising from the quality of students that we had in the class; and (iii) some students had the impression that CS3216 was going to be a step-by-step course on how to program Facebook applications. It turned out that they were simply “thrown into the deep end” and expected to learn most of the things by themselves. Of course, adequate guidance was provided, but that wasn’t quite enough for some of the students.

All in all, our response to these reactions is that the way that CS3216 is structured and designed actually works for some students and there is no reason to be overly concerned about comments of the students who did not like the course. Since CS3216 is an elective, the most reasonable approach moving forward to reduce such adverse reactions is probably for us to ensure that students who sign up for the class know what to expect. It is common for students to consult their friends before taking a class and since some 50 students have taken the class, they would probably be able to provide their friends with the information required to make an informed decision. That said, we also concur that there is a need to moderate the workload for the class in future offerings.

We also attempted to evaluate the learning outcomes by asking the students what they think they learnt in the course. Given the diverse backgrounds of the students enrolled in the course, it came as no surprise that different students would have learnt different things. It was not a goal for the course to ensure that every student who completes the course would be a qualified Facebook application programmer. Such a goal is impractical. The students from the Business and Arts Faculties who were not programmers cannot reasonably be expected to pick up the requisite programming skills within one semester. Nevertheless, all students would be exposed to the process of creating Facebook apps and the non-programmer students would have learnt something about the software development process.

That said, it clear from the feedback and quality of the applications developed that many students managed to pick up the requisite skills required to develop Facebook applications. However, what is an important feature of this course and this is something that is appreciated by many students is that the course has also facilitated a meeting of minds. There were for example Business students who took the class to get to know more programmers because they had intentions of doing some technology-related start-ups and signed up for the class to get to know good programmers. This is quite enterprising and commendable.

The most tangible assessment of the course is perhaps in the “eating of the pudding,” i.e. the quality of the applications developed by the students. In this regard, we are pleased to report that the quality of the apps developed has been high. In fact, we are of the opinion that the quality of the apps developed by our students compare favourably with those developed by students at Stanford [3]. In terms of the application popularity our students did not fare quite as well – but for good reasons. For one, it is much more difficult to propagate apps under the current environment. Facebook has clamped down on the number of invitations that apps can issue daily and there’s been a sense of “app fatigue” among Facebook users. Next, There is a huge number of apps currently available and the competition for eyeballs is intense. As at May 2008, there were almost 26,000 apps, compared probably a few hundred at the beginning of the year when the Stanford class ended. Finally, we made it clear to the students that attracting users was not core requirement for success in CS3216 and they should not be excessively concerned about attracting eyeballs. Instead they should focus on what they thought would be worth doing and to do it well.

Some of our students started to receive offers of freelance work after they finished the second assignment. In particular, the team that developed Prosperity Garden [8], a Chinese New Year app, stands out as a team fully capable of developing high-quality apps extremely fast. The following is an overview of some of the more noteworthy apps that were developed by the students as final projects:

- FarmWars [1]. This is a brilliantly executed Facebook game that started as a class assignment and further developed as a final project. The game managed to attract 9,000 players within three months and the user base has continued to grow. It would not be surprising for the game to have 50,000 players by the end of the year if the rate of growth is maintained (and if the students continue to maintain their app). Much credit must go to the team for not only building a good game, but for successfully building an online community around their game.
- My Music [4]. This is an application that allows users to play tunes on a virtual piano and send the tunes to their friends. The tunes can also be embedded in webpages and downloaded as ring tones for mobile phones. The students linked up with Musicpedia, the online music search website, to allow their users to search for the scores and names of tunes that they can play, but for which the names are not known. Once the scores of such songs are found online, their scores can be downloaded for playback on the virtual piano.
- MTV Cribs [6]. Two of the students worked out a collaboration with MTV to develop an app called MTV Cribs. The app allows users to interact with each other and chat using online avatars. The app is quite similar to Google Lively — and credit must be given to the students for coming up with the concept independently. The students were granted internships with MTV at the conclusion of the class to further develop the app during the June vacations for an official launch scheduled in August 2008.
- Get Help! [7]. From their success with Prosperity Garden [8], a group of students were approached by a Silicon Valley startup called Discoverio to develop a Facebook app that is aligned with their goals and as a form of market survey.
- Tankie [10]. This is a Java-based tank game that is technically challenging to deploy unlike traditional php or Flash applications. In addition, this game is really an initial proof-of-concept deployment for a research project on peer-to-peer networking. Facebook is used as a platform for collecting user data for the prototype system.

The end-of-semester module feedback for the course is attached in Appendix C. CS3216 has received relatively good feedback compared to other modules at the same level. It is curious to note that while an overwhelming number of students complained about the workload, about 50% of the respondents gave the course the highest “Excellent” rating.

### **3.2 Lessons Learnt**

CS3216 is an experimental course and we did not know what to expect when we started. It is certainly very different from traditional lecture/tutorial/exam-based courses and at the same time quite dissimilar from traditional software engineering courses. The teaching of CS3216



was truly a learning experience in itself. The following is a summary of some of the key lessons learnt in the process:

1. **Work around the schedules for the other classes.** One of the things that we did right in CS3216 was to schedule most of the assignments early in the semester. Assignments for most classes tend to be due either the week before the midterm or on the last week of the semester. Midterms tend to be held the week immediately following the midterm and students are mostly unable to spare the time to work on projects and assignments.
2. **It is important to understand how students think.** We had planned for the students to do mutual code reviews for their second and third assignments. The intention was for students to exchange their source codes and to review and comment on the source code for the assignments done by other teams. It turned out that most of the reviews were written by the non-programmers because it was felt that they were not able to contribute quite as much to the actual programming and so should do more writing. Unfortunately, this completely defeated the purpose of the exercise as the non-programmers were not in a position to critique code meaningfully.

Another “*mistake*” we made was not enforcing the preliminary deployment of the final project strictly. As a result, most of the groups procrastinated and only started working furiously on their projects during the last two weeks. This compromised the quality of some of the projects. More importantly, this meant that some of the group’s did not get the chance to experience the community-building and interactions with users that is a core feature in building successful Facebook applications.

3. **Do Less. Learn More.** The workload for the course turned out to be heavier than expected this past semester because of two main reasons:
  - Being a new class and we had concerns that the module would be perceived as “fluffy” and lacking in technical depth. This concern has turned out to be completely unfounded. Students in general perceive CS3216 to be as technically challenging as CS3214 and CS3215, albeit in a somewhat different context.
  - The quality of the students was very high and the standard of the “Hello World” exercise was extremely high and it sparked an “*arms race*” as students tried to out-do each other. Prima facie it was not exactly something bad, but it certainly caused the students to spend a lot more time on the assignments that we had anticipated.

In the next offering of the course, the workload will be moderated, and the requirements for the class articulated to the students so that they know what to expect. Too much work is actually detrimental to the objectives of the class, which was to encourage students to think creatively. When the students are too busy implementing their apps, they end up not having the time and energy left to think harder about what they are doing. We note also that there was visible fatigue in the students towards the end of the semester and some were much less enthusiastic about the final project than we would have liked.

Moving forward, the motto for the class will be “more thinking, less doing” (though there will most definitely be some amount of “doing”).

4. **Self-Learning is Important.** From the feedback, it is clear that we have a number of students who really don't like the idea of being "forced" to learn on their own. Teaching is however not a popularity contest – and it is probably about time for students to learn that can learn without their teacher spoon-feeding them. We see no need to change this aspect of the course and in fact believe that perhaps more courses should adopt a similar model. The students might not like it much, but someday, they will (hopefully) understand that it is good for them.

5. **It's hard to make them smart; let's just try not to make them stupid.** We have found that some of our students are really quite capable on their own, and their technical competence world-class. Many of the apps developed were really quite impressive. We however make no claims over having "taught" them how to develop good apps. We merely provided these students with an environment conducive for them to learn what they needed to learn in order to do the right thing.

The teaching of content is passe. In our increasing complex and evolving world, even the teaching of skills is passe. What matters are perhaps the softer skills of creativity and attributes like passion.

As teachers, we also have to recognize the power of expectations. We are convinced that many of the students did well because they were expected to do well. In the context of CS3216, Much of teaching has to do more with convincing students to exercise their capacity to learn rather than teaching them content.

6. **Need to convince students that learning is worthwhile (and fun!).** As Randy Pausch says, teachers are really salesmen and our job is to peddle education to the students. If the students are already sold even before we start, then half the battle is already won. Facebook is cool, because the students already like it and think it's cool.

Furthermore, it is important for student to have an opportunity to do something they believe in and that they believe has value even after a course is over. In this regard, we believe that CS3216 has done the right thing. For the students who did well, they will also have developed a portfolio that is helpful for their resumes. One of the students for the course got an internship with the MIT-Gambit Game Lab partially because he did some really nice apps for his assignments. Another student was influenced by his fellow students in CS3216 to apply for NOC and he has successfully obtained a place and is now in Silicon Valley.

7. **Shared Suffering Promotes Bonding.** A very interesting phenomenon observed in CS3216 is that many of the students formed deep friendships. Even after the course is over, we find them posting photos of their outings and birthday celebrations on Facebook. There is probably a good reason for this: many groups spent nights together camping out at COM1 together to finish their assignments.

8. **Partnerships with Industry.** CS3216 has demonstrated that it is possible to form meaningful partnerships with the private sector and industry. One group of students developed an application for the social enterprise Gift and Take [2] as a assignment. It is unfortunate that the students seem to have given up on the assignment hence — but such partnerships are private arrangements between students and the external parties and it is up to them

to drive the projects after the class is over. Nevertheless, such partnerships are helpful because it makes the assignments and projects relevant. Since the conclusion of the class, we have been approached by several parties on the prospects of working with our students on their projects, including the National Library Board, Elsevier (Singapore) Pte Ltd. and a game company. We will continue to invite the industry to work with our students in future offerings.

## 4 Conclusion

Moving forward, CS3216 will have to keep evolving to remain relevant and true to its mission. While we were open to the idea of introducing Google OpenSocial, recent developments have been slow and so Facebook seems likely to remain the dominant social network application platform for the near future. We are currently evaluating the possibility of introducing Adobe Air, which is a web/standalone hybrid platform that will support disconnected operation for new mobile applications, in the next offering of the module.

We will also be leveraging on some of the lessons learnt this past semester. For example, the reports for two of the final project groups Fan Gang [11] and Get Help! [7] are currently being turned into case studies for the next semester. Fan Gang will be a case study on team dynamics and project management<sup>1</sup>, while Get Help! will be a case study on the design of good application interfaces.

If we have to sum up CS3216 in one word, it would probably be “*passion*” (or “how to ignite that passion in our students”). There can be no greatness without passion. How do we help students find it? Did we succeed? Frankly, it’s too early to tell but the signs are encouraging — and we’d keep trying regardless.

## Acknowledgments

I am indebted to the many kind souls who sacrificed their time and energy to help make the course a success. My deep gratitude goes to Stan Jarzabek, Klarissa Chang and Lai Zit Seng for taking time off their busy schedules to give guest lectures. I would like to thank Roger Zimmerman, Kok Lim, Mohan, Hon Wai, Mihail, Raymond, and other SoC colleagues for helping with the grading of the final project. Many thanks also to the SoC Workshop for supporting the IT resource requirements for the course and handling the problems and complaints from the students. I would also like to thank the Department (especially Siau Cheng) for their support in the mounting of this new module. Last but not least, the contributions of the CS3216 teaching staff this past semester cannot be overstated — TA Kelly Choo and Tutors Chris Henry, Benny Ou and Gao Shan. Without them, there would be no CS3216.

---

<sup>1</sup>The Fan Gang project was quite a disaster in many ways and a live demonstration of Murphy’s Law. Nevertheless, the team had demonstrated that notwithstanding the problems, they learnt many lessons in the process. While the final deliverable is important, teaching is a process and not a destination. Because the team demonstrated that they actually learnt something, they were awarded a relatively good grade for their final project, notwithstanding some serious flaws in the execution/implementation. I also believe that there is a need to reward students for demonstrated learning even if there might be issues with the deliverable. We are in the business of the education and the students are here to learn and not develop a product.

## References

- [1] Cheah Huay Lian, Deo Amarawi, Edward Chua, and Caleb Chao. Farm Wars, March 2008. [http://apps.facebook.com/farm\\_wars](http://apps.facebook.com/farm_wars).
- [2] Chen Shaoqiang and Png Shao Wei. Gift and Take Facebook Application, March 2008. <http://apps.facebook.com/giftandtake>.
- [3] BJ Fogg and Dave McClure. CS377W: Creating Engaging Facebook Apps, September 2007. <http://credibilityserver.stanford.edu/captology/facebook/>.
- [4] Stephan Goh, Mei Gangwen, Vu Viet Quynh Huong, and Joshua Koo. My Music 2, May 2008. [http://apps.facebook.com/my\\_music](http://apps.facebook.com/my_music).
- [5] Google Inc. OpenSocial, November 2007. <http://code.google.com/apis/opensocial/>.
- [6] Munir Hussin and Nanthekumar Tamilselvan. Mtv cribs, May 2008. <http://apps.facebook.com/mtvmcribs>.
- [7] Kent Nguyen, Ngo Minh Duc, Chin Su Yuen, and Ahmed Wali Aqeel. Get help!, May 2008. <http://apps.facebook.com/gethelp>.
- [8] Kent Nguyen, Ngo Minh Duc, Chin Su Yuen, and Ahmed Wali Aqeel. Prosperity garden, March 2008. <http://apps.facebook.com/happycny>.
- [9] Randy Pausch. The Last Lecture, September 2007. <http://video.google.com/videoplay?docid=-5700431505846055184>.
- [10] Ali Razeen, James Yong, and Koh Zi Han. Facebook tankie, May 2008. <http://apps.facebook.com/fbtankie>.
- [11] Vincent Woon, Jean Lim, Maung Soe Thiha, and Nguyen Duc Phong. Fan gang, May 2008. <http://apps.facebook.com/fangang>.

## A Schedule for CS3216, AY2007/2008, Semester 2

Date	Topic(s)	Instructor(s)	Time
14 Jan 2008	Lecture: Introduction to CS3216 & Anatomy of a Facebook Application	Dr Ben Leong	2 hrs
21 Jan 2008	Lecture: Principles of Software Engineering	A/P Stan Jarzabek	2 hrs
28 Jan 2008	CS3216 Final Project Pitching Session	Dr Ben Leong	2 hrs
4 Feb 2008	Lecture: User-Centric Web Applications	Dr Klarissa Chang	2 hrs
11 Feb 2008	Guest Lecture: Facebook Development with Rails	Jeff Lim	2 hrs
16 Feb 2008	CS3216 Final Project Pitching Party	Dr Ben Leong	3 hrs
18 Feb 2008	No class	-	-
23 Feb 2008	<b>MIDTERM BREAK - No class</b>	-	-
3 Mar 2008	Code Review for Assignment 2	Dr Ben Leong	1 hr
	Guest Lecture: Facebook & Security	Er Chiang Kai	1 hr
10 Mar 2008	Facebook Application Seminar	Dr Ben Leong	2 hrs
17 Mar 2008	Beyond the hype of Facebook: Sustainable Applications and Approaches for Monetization	Dr Andreas Weigend	2 hrs
19 Mar 2008	Techniques and Pitfalls in Provisioning for a Million Eyeballs	Lai Zit Seng	1 hr
	Facebook Application Seminar (continued)	Dr Ben Leong	1 hr
24 Mar 2008	Understanding the *REAL* World: Plenary Session with Local Entrepreneurs	Marc Goh, Paul Yeo, Ash Singh, Dr Kuo-Yi Lim, Dr Sivam Krish, Chin Kwek Loong & Ng Chin Leng	2 hrs
31 Mar 2008	Guest Lecture: Exploiting Open-Source for Fun & Profit	Harish Pillay	2 hrs
7 Apr 2008	Guest Lecture: Google Web Toolkit	Ho Wee Chong	2 hrs
14 Apr 2008	ISM Presentation: Developing Applications for Multiple Social Networking Platforms	Zheng Junyi	1 hr
	The Last Lecture	Dr Ben Leong	1 hr
16 Apr 2008	Final Project poster presentation		-

## B Midterm Survey

In this section, we present the responses for a mid-term survey conducted after the midterm break. The sample set consists of 44 respondents out of 51 students.

### B.1 Reasons offered for taking CS3216

The following is a list of the reasons offered by students on why they decided to take CS3216:

1. It seemed interesting, challenging, unique and informative.
2. Sounds fun
3. To do something different. Its like either do it now, or forever miss it. With regards to qn 3, the assignments are definitely not hard. But it takes up 100% of your time to make a proper app.
4. I wanted to try something no one else does before. And I was interested to see how I can work with different kind of people from different background.
5. Thought it was interesting and different from other NUS modules.
6. It sounded interesting, and quite a detraction from the normal NUS type modules. Also, thought it would be a good chance to get my hands dirty with Facebook application development
7. I dont know anything about how to write a web app. Taking this module will help me to do this. Besides, I can meet new friends, quite good lah....
8. I thought that it will be a fun experience. Being a facebook developer sounds cool to me. And, the module will be something very different from the normal boring NUS module anyway, so why not right?
9. I had intended to pick up web programming in an accelerated manner through this course and at the same time learn about the basics of software development.
10. find it interesting and new. was addicted to facebook.
11. Interest in web programming and find this new module very interesting. Wish to know more ppl who share the same common interest as me.
12. To try out something new and learn facebook development and to build on to my portfolio. To meet like-minded people. And I prefer practice-oriented learning to theory-oriented learning, since I like puddings.
13. I wanted to meet more ppl.
14. Its new and interesting, so I decided to give it a shot

15. It was something very different from the usual modules found in NUS, and it's a first time for this module, so I thought it'd be fun to be one of the pioneers. Plus it was a Facebook module which allowed me to develop my ideas and whatever I want, and to create applications I always wanted to.
16. To learn more about facebook and social networking sites.
17. I perceived this module as a vibrant module where I could meet with interesting people in addition to a professor who has broken off from the norms of NUS.
18. Partly because I was in a research group that needed to deploy a game to collect network-usage data. Another reason was that I felt that it would be challenging to take a module without knowing how exactly it will end.
19. I wanted to take a module which actually allowed me to do what I want, make what I want.
20. Hope I can do something cool; make new friends, learn new things...
21. I am interested in Open Source and want to get the hang of FB programming.
22. Mainly curiosity...
23. Great opportunity to learn anything and everything. Chance to do something out of the world.
24. Out of interest
25. I thought that a module of this nature would be interesting, and that there was something i could take away from it to apply in other modules.
26. I want to try something new
27. by then the application of this module started, i was just starting to using facebook. think it would be great if some time could use the app i made on my own.
28. Hoping for something cool, which I guess, I got more than just cool. Haha
29. I wanted to meet and work with interesting and smart people. A course like this would be very likely to attract talented students to join it.
30. 1. Ben is cool! 2. Facebook and Web 2.0 is the future of new media. As a CM student, I think it is necessary to know the theories and technology in it. a new module it is challenge know more people who have strong programming skill or enterprise spirit Dr.Ben :)
31. The FB platform is currently the hot new thing and it was a great chance to get involved and see first hand what the fuss is all about.
32. I thought it's fun.
33. To learn new skills and meet other people

34. I wanted to give myself an excuse to learn web programming languages (PHP).
35. Coz its fun and SOMETHING USEFUL and practical! I can actually create something and show it to the world (unlike the other not so fun modules)

## B.2 Summary of Results

1. Did you think the assignments were designed well to meet your learning needs?

Response	Frequency
Complete waste of time, I learnt nothing.	0 (0%)
There was little to be learnt	3 (6.8%)
I learnt some stuff, but not a lot	15 (34.1%)
I learnt quite a lot	16 (36.4%)
I learnt more in CS3216 than all my other classes put together	10 (22.7%)

2. How would you assess the difficulty of the assignments?

Response	Frequency
They were too easy	0 (0%)
They were not much of a challenge	5 (11.4%)
Average, just like my other classes	10 (22.7%)
They were challenging	26 (59.1%)
I almost wanted to commit suicide	3 (6.8%)

3. How do you find the module workload?

Response	Frequency
Too light, I want more work!	0 (0%)
On the light side.	0 (0%)
It's like any other normal module.	1 (2.3%)
It tends to be on the heavy side.	23 (52.3%)
Way too much workload, I can hardly breathe!	20 (45.5%)

4. How would you rate the lectures?

Response	Frequency
They are completely useless. I learnt nothing.	1 (2.3%)
I have no idea what's happening half the time	4 (9.1%)
Lectures are no different from the other classes on campus	11 (25%)
Lectures are clear and I am able to follow the material quite well	13 (29.5%)
Lectures are way cool. Easily the best class that I've taken at NUS	15 (34.1%)

5. What is your overall impression of CS3216 thus far?

Response	Frequency
This is a horrible class. Truly regret choosing it.	0 (0%)
It's alright, but mostly painful.	13 (29.5%)
Just like any other module	0 (0%)
It's a good module and I definitely enjoy it.	14 (31.8%)
CS3216 rocks! Coolest class I have taken in my life.	16 (36.4%)
Skipped	1 (2.3%)



### B.3 What students say they learnt

The following is a list of the responses from students taken from the midterm survey when they were asked what they think they learnt from CS3216:

1. I've learnt more about application design and logic. Forced myself to discuss database structures with the programmers. :) Got to know really smart friends!!
2. Do not restrict yourself! And facebook applications are not that difficult to make!
3. New programming techniques
4. I did not learn much from the lectures and seminar but I learnt a lot from working with people, knowing the right people for the job and knowing my own strengths and weaknesses.
5. How to design a popular application
6. Different faculties work differently.
7. Learnt about good programming practices, which is useful. Learnt about teamwork as well. Learnt that everything can be commercialized to make money, sigh.
8. Sleep less, study less, how to keep up with other modules and survive.
9. Some programming knowledge, some Photoshop/Illustrator/Flash skills, networking skills, marketing skills
10. Well, most of it is to know people. And getting to really-know people.
11. A lot of things, too many things to list down.
12. How to create a facebook application as well as greater understanding of social networks. Learnt how to derive a business model for my final app as well.
13. I've learnt how to write a web app, how to work as a team.
14. How to think.
15. Most importantly, how the FB platform works and what it can (and can't do). The broader implications over the whole internet as well indirectly.
16. I have learnt a lot so far. I have learnt how to create applications that suck and serve no real purpose to anyone and I have learnt how to make really fun applications that have a lot of potential. The coolest thing is that the assignments I make in this class don't just go to waste, they are actually use by real people elsewhere. I don't think there are many modules that can do that.
17. a. Designing a FB app is easy. Designing a GOOD FB app is not. b. Groupmates matter. c. Sometimes you have to be the bad guy and slavedrive. d. Advertise yourself and your skills. it helps. e. Good programmers != good application f. Don't feature creep. g. KISS (Keep it simple, Stupid) h. Make something that works as opposed

to something that tries to wow the pants off everyone but doesn't work. i. SOC zones are unstable. j. Web application users are a fickle lot. k. Its not always about making something revolutionary. you can always take something good and make it better. l. NICHE MARKETS m. You know you're good when you have a fanbase.

18. I learnt a lot from this module, not only programming knowledge, but friends, and inspired a lot by them!
19. Everything this course was designed for + more, such as teamwork. The best thing was that I got to know a lot of talented people!
20. Facebook application development, and a bunch of stuff from the different lectures. I also learnt that the proof of the pudding is in the eating, and that I like to eat puddings.
21. I've learnt from the other students in this module, and the potential and capabilities to do great things.
22. Project management, developing software in a new platform.
23. Software Engineering Principles, Security, RoR Many of the lectures which were intended to get certain principles across were successful. Along with the academic material I also learnt more about working in teams and networking.
24. Trivial things like the facebook API aside, mostly non-technical skills : interacting and dealing with people, reading lengthy reflections and musing on different perspectives etc. Oh, learning to use Ajax was pretty cool too :P
25. I think it's more the experience for me. I like having various guest lecturers come in to speak, which gives the class a wider view upon the world, rather than have one single lecturer rattle on for the entire semester. I also like the activity within the class forum. This is truly the most active IVLE module I have seen in my few years in NUS. I really like the life and vibrancy in the class.
26. Cool things really depend on idea, luck and lots of effort.
27. Many things about web development that I haven't had time to learn before: PHP, CSS, HTML, Database, JScript... and the facebook platform
28. A rather big picture view of current web technology, doing apps on facebook, some theories on security, project management and propagation via the web
29. How hard could it be to make cool stuff.
30. A lot! Project management, source control, the confidence to learn a new language in one day (but i shall hope never to do that again, ever.), teamwork and management, how social platforms work and its direction in future, most importantly making friends and contacts who CAN do work, so that in future we know who to look for to get work done.
31. I get a better idea about the facebook platform. I have hope to actually learn to program and develop application. However at the pace and the focus, it is hard for non programmer to actually learn much.

32. I have learnt what I had wanted to learn, but painfully. I had hope I would be taught, but instead I was forced to self learn. This is VERY BAD.
33. Loads of web programming, software development, how to create cool stuff in Facebook.
34. Mostly on the personnel management side.
35. I learnt more lifeskills than anything else, which is great in a way. I think Ben said he wanted us to learn something like this in the process
36. Working with people... planning ahead and strategizing...

## C End-of-Semester Module Feedback

The following is the end-of-semester feedback by the students on the module (“Overall Opinion of Module”). The sample set consists of 33 respondents out of 51 students:

ITEM SCORE	Excellent	Good	Satisfactory	Unsatisfactory	Poor
Module	16 (48.48%)	11 (33.33%)	5 (15.15%)	1 (3.03%)	0 (.00%)
Module at Same Level (Dept)	88 (18.45%)	193 (40.46%)	153 (32.08%)	33 (6.92%)	10 (2.10%)
Module at Same Level (Fac)	129 (18.17%)	290 (40.85%)	227 (31.97%)	45 (6.34%)	19 (2.68%)

The following are the answers from the respondents to the question “Please comment on the strengths and weaknesses of the module, and suggest possible improvements”:

1. I learnt a lot from this module, not just in terms of software engineering. Workload-wise a few adjustments might be in order.
2. Too many separate, independent assignments. As a result, people get caught up with maintaining their previous applications (if they are successful). Should implement a large project to be divided into different milestones and graded accordingly.
3. Strengths: 1. Cross-Faculty module allows us to leverage on strengths from different skill-sets. 2. Engages us with external experts to enhance learning. 3. Selection criteria ensures that only well qualified students are taking the course, as opposed to students who are just “passing time” Weakness: 1. Workload is too heavy
4. This is the best module i had ever taken in NUS, i learnt so much knowledge from this module, not only technical skills but also communication skills and how to work with different people. The module is well prepared and the lecturer put in so much efforts to make this module successful. The module can be further improved by organizing the schedule more effectively.
5. Strengths - Hands on, opportunity to meet many talented students/entrepreneurs, challenging, diverse lectures and perspectives, space to express creativity (Relative) Weaknesses - Insufficient time to really enter into depth, many deadlines, shifting of grading requirements
6. I believe that the description of the module is misleading.
7. Too much workload, course objectives not clearly defined. Of course i understand this is the first time this module is held, so things are experimental in nature, but subsequent runs should hopefully be improved on
8. READ ALL, IMPORTANT!!! This is one of the best courses i have taken in the past three years at NUS. It has certainly redefined my NUS experience. There weren't any better months i spent at NUS, and surely no other module in which i met so many great people, and learned so much, I am truly indebted to all the people who made this course possible.

I had never seen so much talent and energy in a room before. By all means all FOE teachers should consult Prof. Ben on how to make their courses more interactive, and make learning fun by engaging students in projects designed for real learning. Otherwise, I am certain that FOE may soon be known as a place where a majority of students lose their interest in their major due to severe lack of interaction between the students and teachers. More courses like this shall be introduced for 1st year students, as they are the people who have the passion and energy to do something big and discover more. During the rest of the years, I have only seen them become more mechanical and disinterested in what they are taught. So, in order to compete at the global level, it is necessary that freshmen are introduced to courses designed like this one in their first semester, so that they can discover more, and sustain their interest in their major. If you need any advice on how to do it, feel free to contact Prof. Ben, he is among the few people who recognized the problems that NUS students are facing, and came up with solutions to enable students to learn. The way he has innovated in other courses of his is an example in itself. I cannot emphasize it anymore!

9. More than expected workload, but had lots of fun, learnt quite a lot of stuff, and know great people.
10. can learn a lot, but workload's heavy
11. A little too much work for 4 MCs worth, but definitely a cool not-to-be-seen-in-NUS-anymore module.
12. Very cool and vastly different from the average module. Very interesting. Open ended nature of assignments and degree of competence/talent of the class sets the bar sky high, and increases the workload many times.
13. too heavy workload
14. Workload is towards the very heavy side. But nevertheless I find this module not like the normal NUS modules I have taken
15. Strengths - teaches us a lot of different things, from software engineering to selling. Weaknesses - certain things are a little disorganized, but this is the first batch, so it's okay I guess.
16. Workload is insane. Really appreciate the professor's effort in organising the guest lecturers and plenary sessions, etc. Lectures are generally fun and something to look forward to.
17. This is a very fun and diverse module that encourages students to think and explore various possibilities. However, it is too time consuming as it has a lot of different assignments and projects, and sometimes several at the same time which makes things a bit confusing. There should be a review of the number and type of assignments throughout the course of the module.
18. Extremely time consuming and suitable only for students with immense dedication and self discipline. Requires great amounts of self learning and exploration.

19. very challenging, but then I guess everyone in the class loves challenges ;)
20. Too difficult in the face of other modules. This module could potentially take up 100% of the students' time. Might be good to span 2 semesters instead, because good applications, whether on evolving platforms or stable ones, take time to develop. If not, following the tried and tested solutions out there don't really allow us to exercise our creativity and definitely doesn't encourage us to wonder into uncharted waters.
21. Too much workload and too ambitious.
22. Strengths: A brand new module in which students can learn very practical things. There are a lot of guest lectures who came to share their valuable experiences. Students benefit a lot from doing their own Facebook applications. Weakness: The workload is a bit too much.
23. An experimental class with a different teaching style. Had a very good experience taking this module and would recommend this to fellow students who are not afraid of a challenge.