***Abstract:***

# Introduction

This project will be a continuation of earlier rateMyLab work by the School of Computing: it would allow for further development of an app for Android tablets that would be placed in student labs/lecture theatres to gather from students: (1) end-of-class ratings; (2) text comments; (3) (possibly) attendance information.

The project will also provide a system which will allow for analysis of the feedback data (e.g. for particular lecturers, modules, etc.), which will be stored in a server-side database. The system will be accessible to all lecturers for their relevant classes and modules.

Throughout this project, the end user is to be the focal point. Various user-centred design techniques and strategies will be used, including rapid prototyping and user profiling. User testing will be thorough, with the app being tested in the end user environment (i.e. in a lab/lecture) in order to ensure that feedback is as relevant as possible.

# Background

Student feedback is an important tool that benefits all contributors to a lecture or lab class: (i) giving students a mechanism to report their assessment of the class in question and thereby influence related classes and (ii) helping lecturers to identify and tackle issues experienced by their students and thereby subsequently develop the curriculum accordingly (Johnson, 2009).

There is an already existing app (rateMyLab), developed by the School of Computing, which aimed to provide a mechanism for obtaining such feedback, but it needed improvement – although it was simple and quick to use, the feedback it obtained was seriously lacking in detail (it allowed users to specify whether a lab was good, bad or average – without specifying *why*). This problem is what we aim to solve with this new phase of development. The aim is to provide a mechanism which allows for richer feedback, whilst maintaining the simplicity and accessibility of the original rateMyLab app.

### Previous Work

On a more global scale, there has been similar work done in the form of Classroom Communication Systems (CCS). These systems have evolved, from being based on multiple-choice remote controls and then PDAs, to the more modern medium of Tablet PCs with wireless connectivity, the idea being that every student in a lab/lecture has one of these devices, and they are all connected through the medium of the CCS. These systems are very useful for bringing classes together in a collaborative effort, giving the mechanism for feedback and allowing for live in-class polling, and demonstration of both good and bad examples and scenarios. Group work is also made much easier as student do not have to physically move around to work together. (Theys, Lawless, & George, 2005)

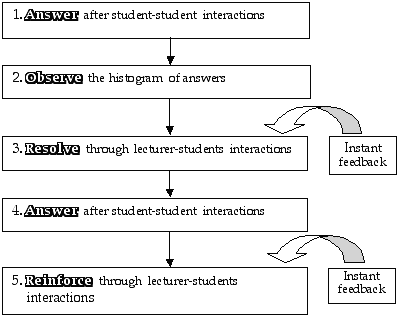


Figure 1: CCS learning cycle (Sharma, Khachan, Chan, & O'Byrne, 2005)

The main difference between rateMyClass and the traditional CCS is that, while CCS’s are more geared towards live feedback and dynamic lecture content (with increased student engagement the priority) (Dufresne, Gerace, Leonard, Mestre, & Wenk, 1996, p. 2), rateMyClass is a more static, narrowly-focused mechanism, with feedback being given only after the conclusion of classes; the main rationale being that lecturers can gain valuable insight which allows them to make improvements to these classes. The rateMyClass app is primarily for the benefit of lecturers, unlike the CCS.

Another similar piece of work involves the use of augmented reality (AR) systems (e.g. Google Glass) to provide lecturers with live feedback from students in a seamless fashion, through the use of various visualisations.

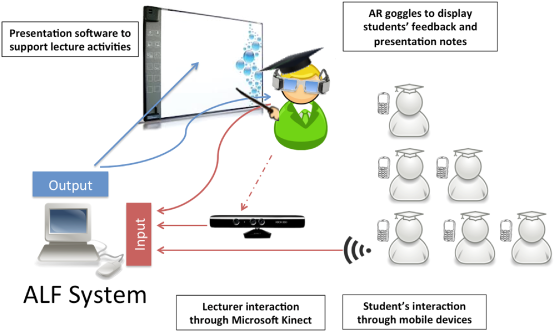


Figure 2: Architecture of the proposed ALF system (Zarraonandia, Aedo, Díaz, & Montero, 2013)

The proposed benefits of such a system would include the fact that such a system would avoid breaking the ‘flow’ of lectures, which often occurs when lecturers have to directly communicate with and ask questions of students, and that it encourages more students to give feedback by keeping their anonymity, removing the need for them to speak up in class (Zarraonandia, Aedo, Díaz, & Montero, 2013).

Additionally, there has been development of a smart phone app (see *Figure 3*) which allows student to check in to class and leave feedback (Foth, Fitz-Walter, Ti, Russell-Bennett, & Kuhn, 2012).

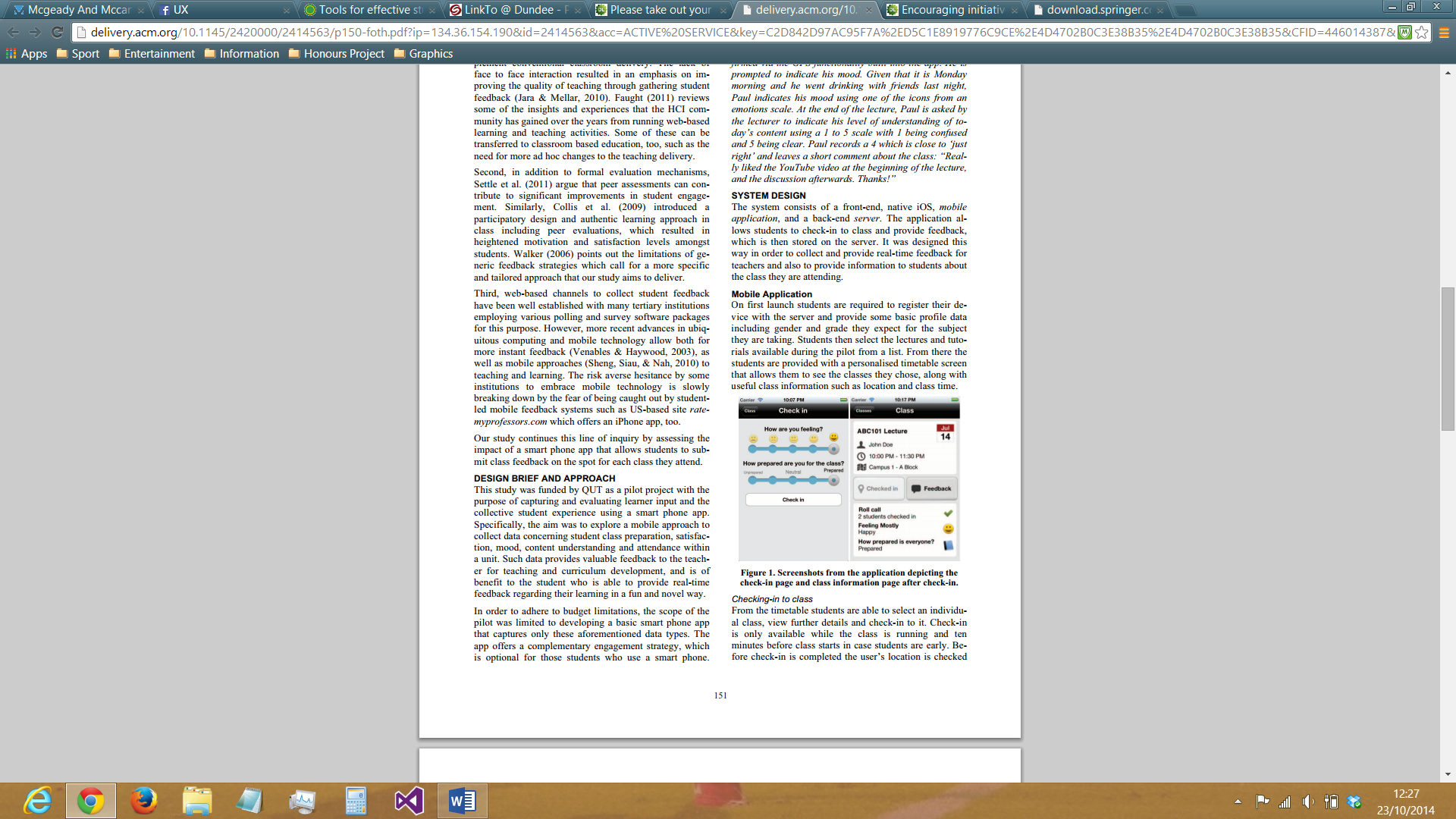


Figure 3: Screenshots from the app (Foth, Fitz-Walter, Ti, Russell-Bennett, & Kuhn, 2012)

This app provides the following functionality:

* Checking-in to class
  + Students are able to select a class from a list of classes and ‘check-in’ to it. This can only be done during the class (and 10 minutes before it starts). GPS is used to check that students are actually physically present at the class. Students can provide initial feedback as to ‘how they are feeling’ (ranging from bad to very unhappy (smiley) to very happy (smiley)) and ‘how prepared they are for class’ (ranging from ‘Unprepared’ to ‘Prepared’)
* Revealing feedback from other students
  + As soon as students ‘check-in’, they can view certain collective items of information about the other students who have also done so. For example, the total number of students who are checked in, and the average levels of happiness and preparedness of the class.
* Providing class feedback
  + Final class feedback can be given by student in the form of two 5-point Likert scales, one rating the class in general and the other rating how well they understood the information given in the class. Students are sent reminders 10 minutes before the class ends in order to make sure none of them forgets to give feedback

As regards tablet-based systems which more closely mirror the functionality, architecture and rationale of rateMyClass, there are no doubt many real-world examples, but, in terms of documented commercial solutions, it appears there are no systems which are really comparable.\*

\*Searches were made across Google Scholar, Dundee University Library and ACM Digital Library

# Specification

The following user stories were identified and developed after initial user interactions and discussions (*see Appendix A for full detail*):

As a student I want to…

* Be able to give feedback on how interesting the lab/lecture was (and why) so that lecturers can use this knowledge to make their classes more interesting
* Be able to give feedback on how innovative the lab/lecture was (and why) so that lecturers can use this knowledge to make their classes more innovative
* Be able to give feedback on how informative the lab/lecture was (and why) so that lecturers can use this knowledge to make their classes more informative
* Be able to give feedback on how interactive the lab/lecture was (and why) so that lecturers can use this knowledge to make their classes more interactive
* Be able to give feedback on how intelligible the lab/lecture was (and why) so that lecturers can use this knowledge to make their classes more intelligible

As a lecturer/member of staff I want to…

* Give students the option to select why a particular area/factor was bad or good (from a list of predefined statements) so that I can get more rich feedback, while students will not have to spend time writing out comments in their own words, and they cannot make irrelevant/inappropriate comments
* Be able to analyse the data after class so that I can make sense of the data and see the bigger picture, along with any trends, and adjust classes accordingly
* Be able to view the feedback(s) for individual classes so that I can determine their quality
* Receive a summary of feedback for each class that I conduct by email so that I can ascertain how well I have done
* Select which module and lecturer the class belongs to immediately upon starting the app so that The data can be referred to later in the correct context

As a dean I want to…

* To be able to view the feedback for all labs/lectures handled by a particular lecturer so that I can determine if individual lecturers are doing a good job or not
* Be able to view the feedback for all classes taken by a particular year group so that I can determine how well a particular year group is doing

# Design, Implementation, Testing & Evaluation

### Overall System Design

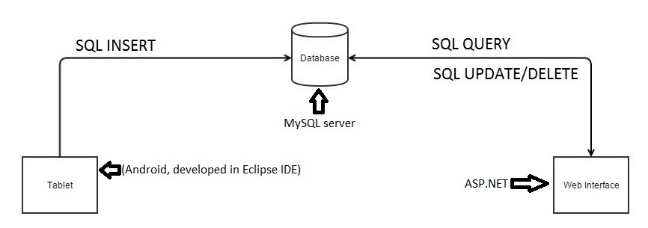


Figure 4: Architecture of the rateMyClass system

Before development of the web interface began, a decision had to made as regards which technology would be used to implement it. After much research and rumination, the dilemma became whether to use ASP.NET (a language which had been used previously) or a more challenging, unfamiliar language in Ruby on Rails. On the one hand, ASP.NET would be much easier to implement, with there being much experience of using this previously (unlike Ruby on Rails); but, once the steep learning curve had been overcome, by all accounts Ruby on Rails would actually allow for quicker and more automated generation of a web interface to the database. That being said, with the limited time within which the project had to be completed (also bearing in mind the steep learning curve required for programming in Ruby on Rails), it was decided that ASP.NET would be used in order to achieve the highest quality results possible.



Figure 5: Entity-Relationship Model of the rateMyClass database

# Sprint 1

### Design/Implementation

#### User Story 1.1: I want to be able to give feedback on how interesting the class was (and why)

Developing this user story involved the creation of an input method whereby the user could give a percentage rating for how ‘Interesting’ a class was. This was implemented through the use of a native Android widget, the ‘Seek Bar’. This feature allowed users to provide an integer value in the range of 1-100 (i.e. a percentage), through an input method which outwardly seems entirely analogue.

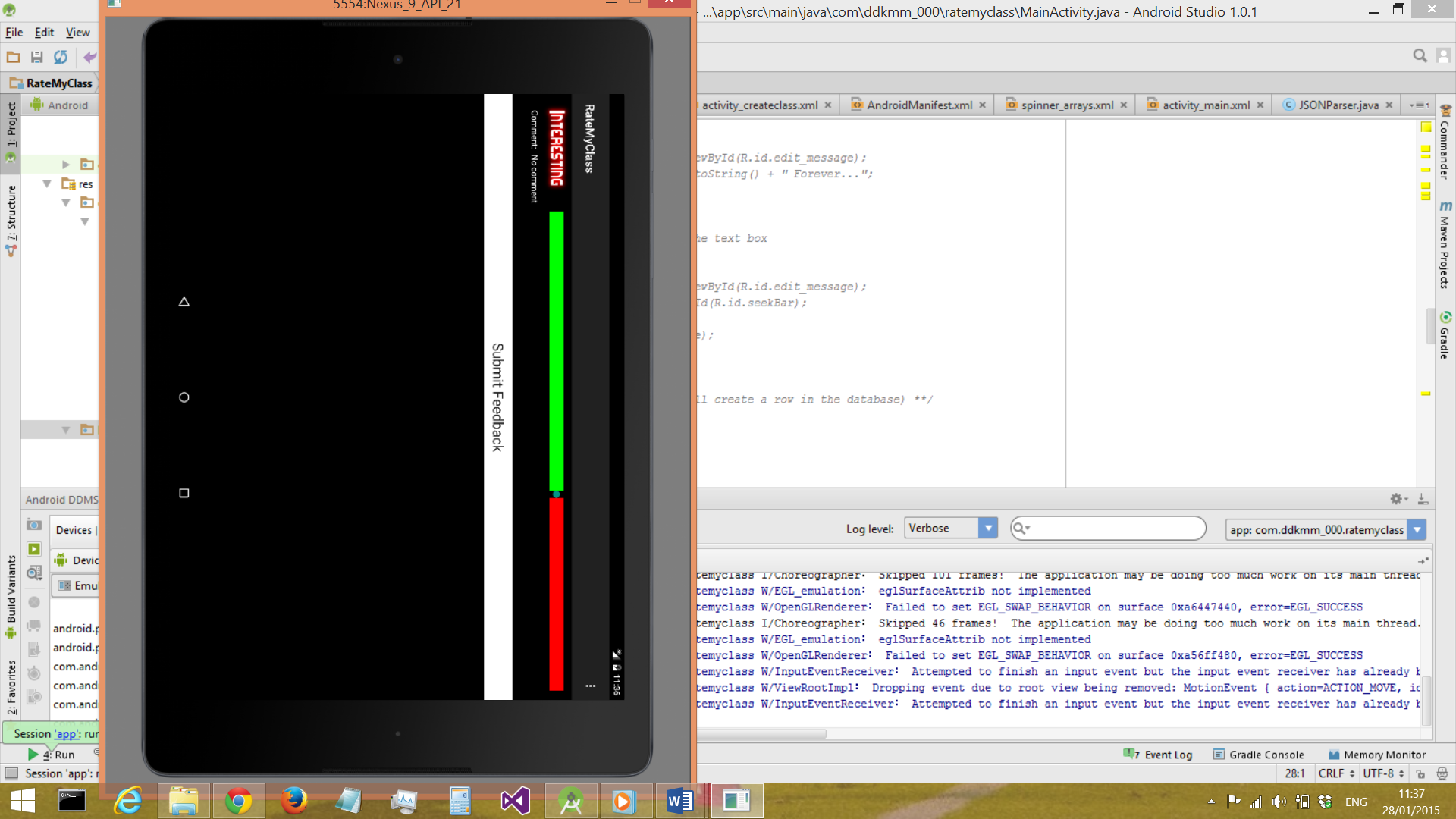


Figure 6: the Seek-Bar input method for giving a feedback 'score'

In the GUI, this value was not displayed to the user in order to preserve the seemingly analogue nature of the input, giving a more ‘natural’ feel to the interaction. The seek bar utilised a green-red split colour scheme (see *Figure 6*) which made the hidden value more clear to the user though, by making the green half of the bar increase in size as the selected value was increased.

#### User Story 2.1: I want to give students the option to select why a particular area/factor was bad or good (from a list of predefined statements)

Developing this user story involved the creation of an input method whereby the user could select one comment from a predefined list. This was implemented through the use of a native Android widget, the ‘Spinner’.

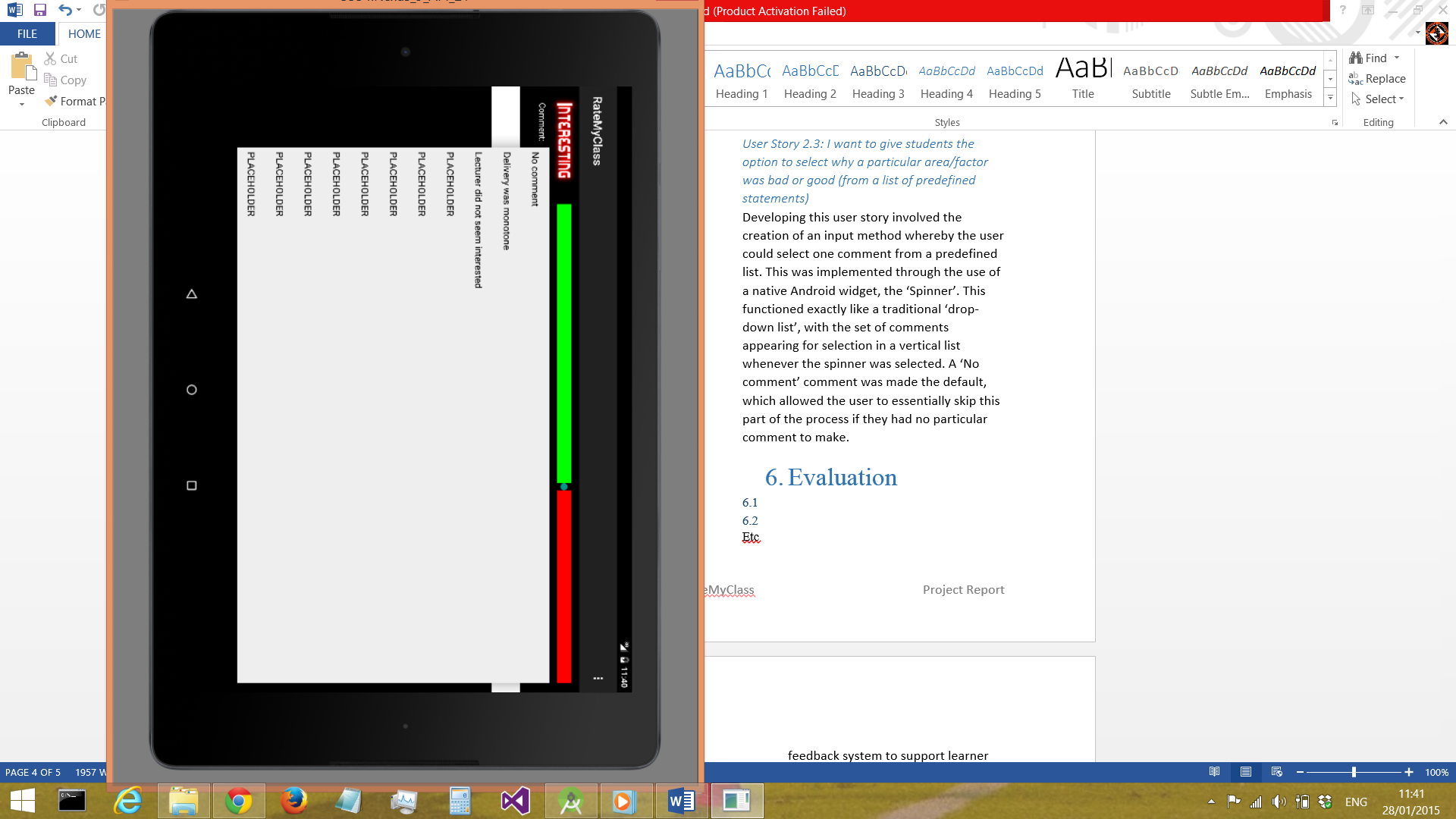


Figure 7: the 'Spinner' input method for comments

This functioned exactly like a traditional ‘drop-down list’ (see *Figure 7*), with the set of comments appearing for selection in a vertical list whenever the spinner was selected. A ‘No comment’ comment was made the default, which allowed the user to essentially skip this part of the process if they had no particular comment to make.

### Evaluation/Retrospective

Looking back at Sprint 1, it was generally a very successful period of development, albeit some areas could have been improved upon. For example, the testing of the software perhaps could have been more rigorous – a decision was therefore made to include unit testing for the android app in the next sprint. Also, it would have better if the information regarding feedback from and interaction with end users had been captured in more detail. Therefore, during the next sprint a logbook could be used to capture any and every piece of user feedback throughout the period of development.

# Sprint 2

### Design/Implementation

#### User Story 1.3: I want to be able to give feedback on how informative the class was (and why)

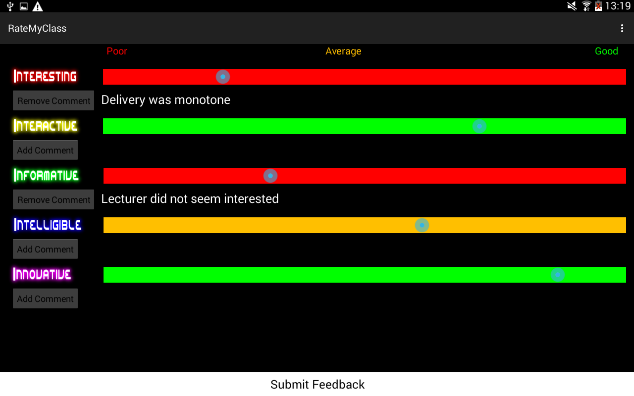
Developing this user story involved simply repeating the technique used to implement user story 1.1 (*see section ‘Sprint 1’*), replicating the input method. The only design aspect which needed to be considered was how the SeekBars would be laid out on-screen. 

Figure 8: Layout of multiple SeekBars

After user discussions, a decision was made to deploy these in a vertical layout (*see Figure 8, above*). These discussions generated two other improvements to the UI: The instead of being a red-green split, the entire SeekBar would change colour depending on the value selected (i.e. 0-33 is red, 34-66 is amber, 67-100 is green); and instead of having a basic Spinner function for comments, the Spinner would be hidden by default, with an ‘Add Comment’ button to activate it.

#### User Story 2.5: I want to be able to select which module and lecturer a class belongs to immediately upon starting the app

This user story was the first item which was to be on a separate screen from the main feedback form – initially, the app was to show a screen which allowed the user to select which module and lecturer the class belonged to, and whether it was a lab or lecture. Once these had been selected, then the main feedback functionality would be initiated.

In collaboration with end user (as always), it was decided that this screen would be very simple (in order not to hinder the feedback process), using Spinners once again to allow selection of the 3 values and prevent any invalid data being input (*see Figure 9, below*).

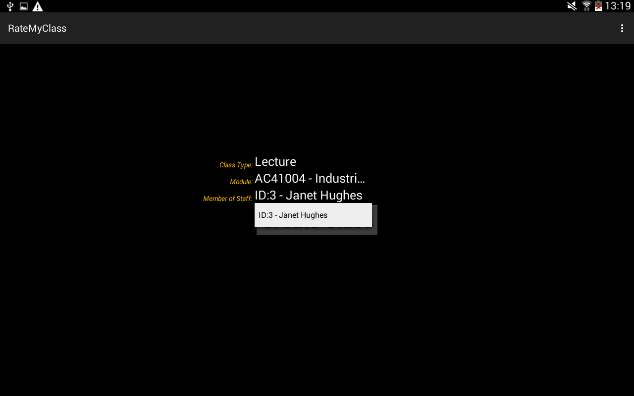
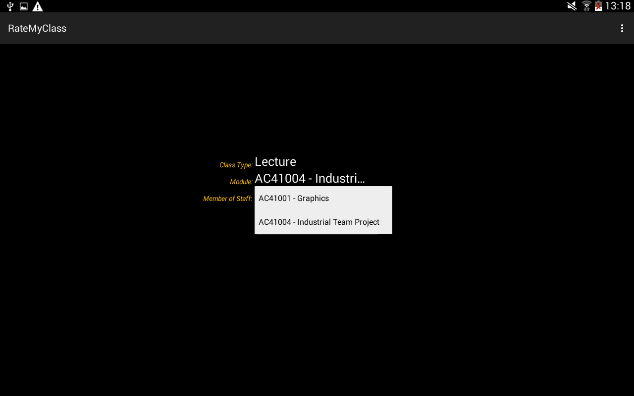
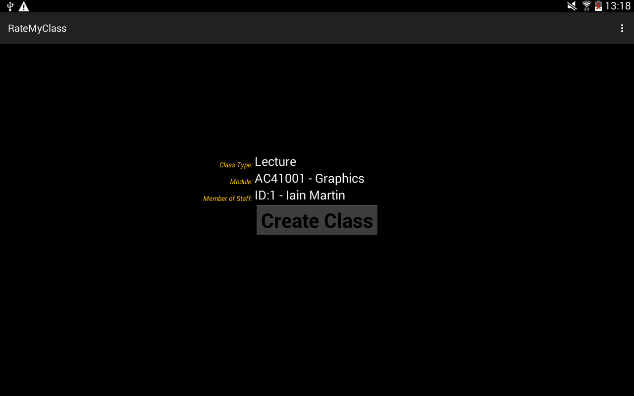


Figure 9: (Prototype) Class creation screen showing the 3 spinners

### Testing

As mentioned in the Evaluation of Sprint 1, this sprint was to include unit testing for the app. Accordingly, unit tests were written and run for both the main feedback activity and the class creation activity, testing the various UI features. As of the beginning of this sprint, development was to be test-driven, writing unit tests first and then writing code in order to pass these. The tests all passed successfully.

### Evaluation/Retrospective

Looking back on Sprint 2, there was definitely improvements made in a few areas, most notably unit testing was introduced and implemented in a satisfactory manner (although code coverage of the unit tests produced could have, and probably should have, been greater). In terms of recording user interactions, discussions and feedback, much improvement is still needed – there was no real increase in the amount of captured, recorded interactions. The next sprint will have to focus on improving this.

Some feedback was received from staff, and will be implemented in the next sprint:

* Module code, module name and lecturer name to be displayed on main feedback screen (in ActionBar)
* SeekBars in main feedback screen to be set at midpoint by default

## Sprint 3

### Design/Implementation

#### User Stories 1.2, 1.4, 1.5: I want to be able to give feedback on how interactive/intelligible/innovative the class was (and why)

Developing these user stories involved, as mentioned regarding User Story 1.3 (*see section ‘Sprint 2’*), simply repeating the technique used to implement User Story 1.1. As mentioned in the last Evaluation/Retrospective, a focus was made in this sprint on increasing and prioritising user interactions – the majority of the work done in this sprint involved enhancing the app’s interface in response to user feedback. Some items of feedback addressed in this sprint included:

* Class Creation Screen
  + Make font colour uniform across all items on screen
  + Revised ordering of selectable items (Module -> Type of Class -> Lecturer)
* Feedback Screen
  + Change seekbar thumb colour to black to make it more visible against coloured background
  + Change seekbar default thumb position to the middle
  + Make colouring of the titles of the 5 ‘I’s uniform in order to reduce cognitive load and avoid confusion
* Make font style uniform throughout the app
* Add gradient effect to all buttons/seekbars in the app in order to improve aesthetics
* Add menu option to exit (and also to create a new class if in the Feedback Screen)

After performing these user stories and refining the UI, development on the app was now at an end – the ‘final version’ can be seen in the screenshots below:

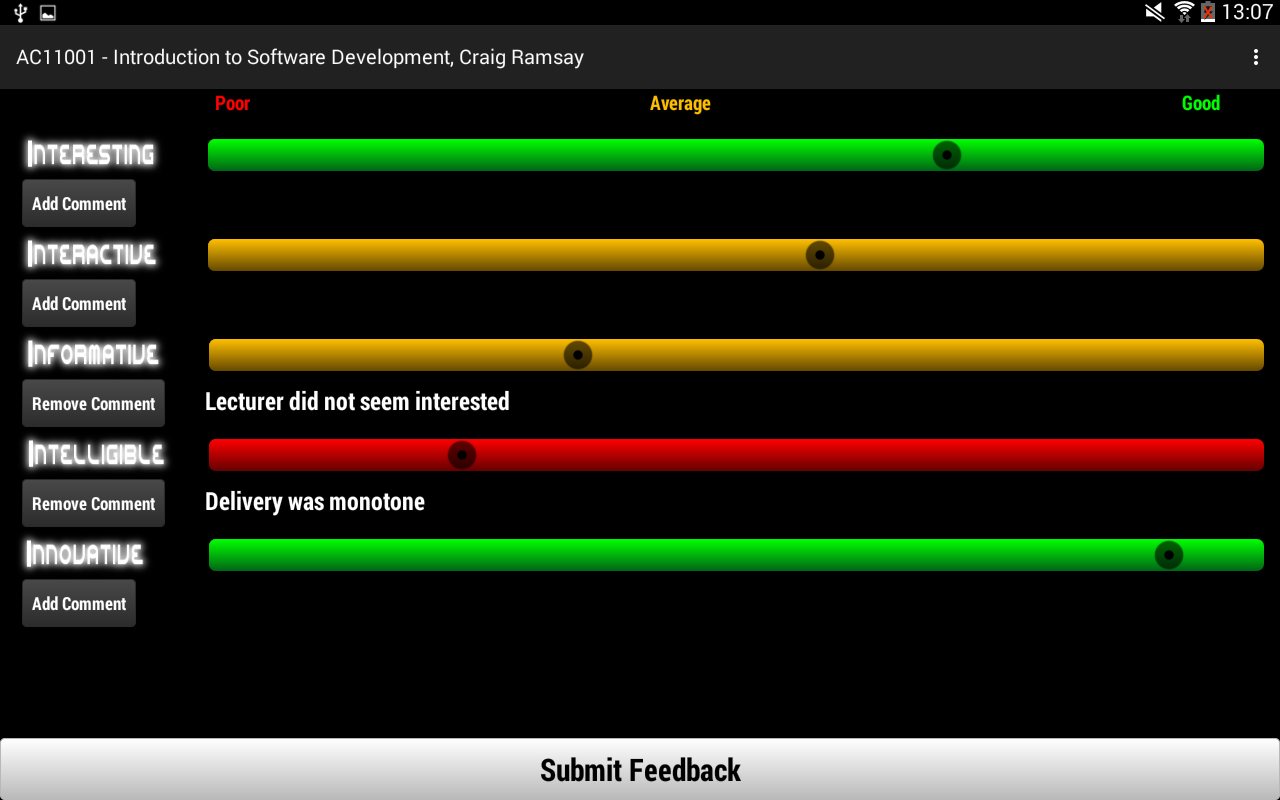
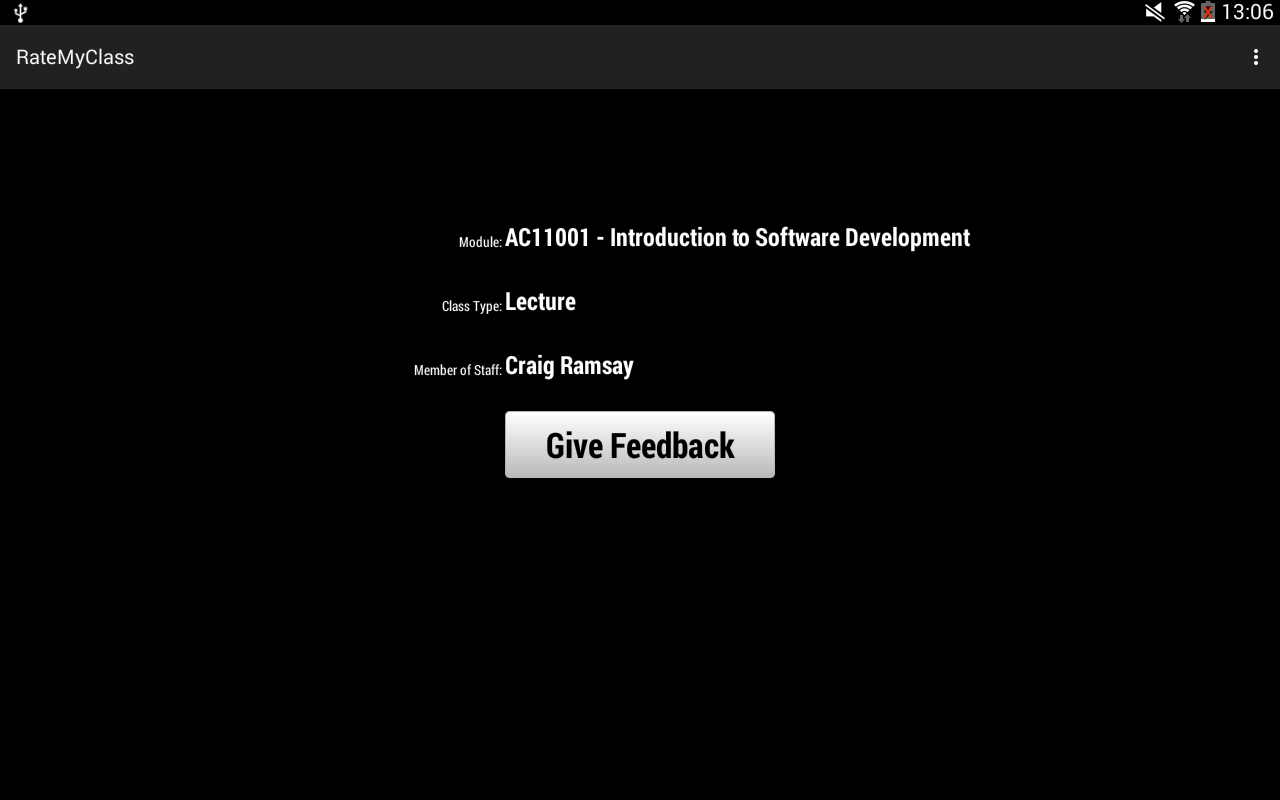


Figure 10: Screenshots of final app implementation

### Testing

Some more unit tests were introduced during this sprint, which tested the new UI items. These all passed successfully.

### Evaluation/Retrospective

In terms of the goals from the last retrospective, staff and student feedback was recorded to a more satisfactory level during Sprint 3. It was clear that user interactions as in general were more frequent and attempts were made to gather feedback from more sources (although not with a great deal of success – in the next sprint this could be rectified).

## Sprint 4

### Design/Implementation

#### User Story 2.3: I want to be able to view the feedback(s) for individual classes

#### User Story 3.1: I want to be able to view the feedback for all classes handled by a particular lecturer

#### User Story 3.2: I want to be able to view the feedback for all classes taken by a particular year group

In order to develop the above user stories which concern specific items of functionality for the RateMyClass website, the basic website structure and design had to be produced first.

After consultation with potential end users, a simple structure was decided upon for the website, with slight variation depending on the user’s level of access (see Figures 11, 12 and 13 below).

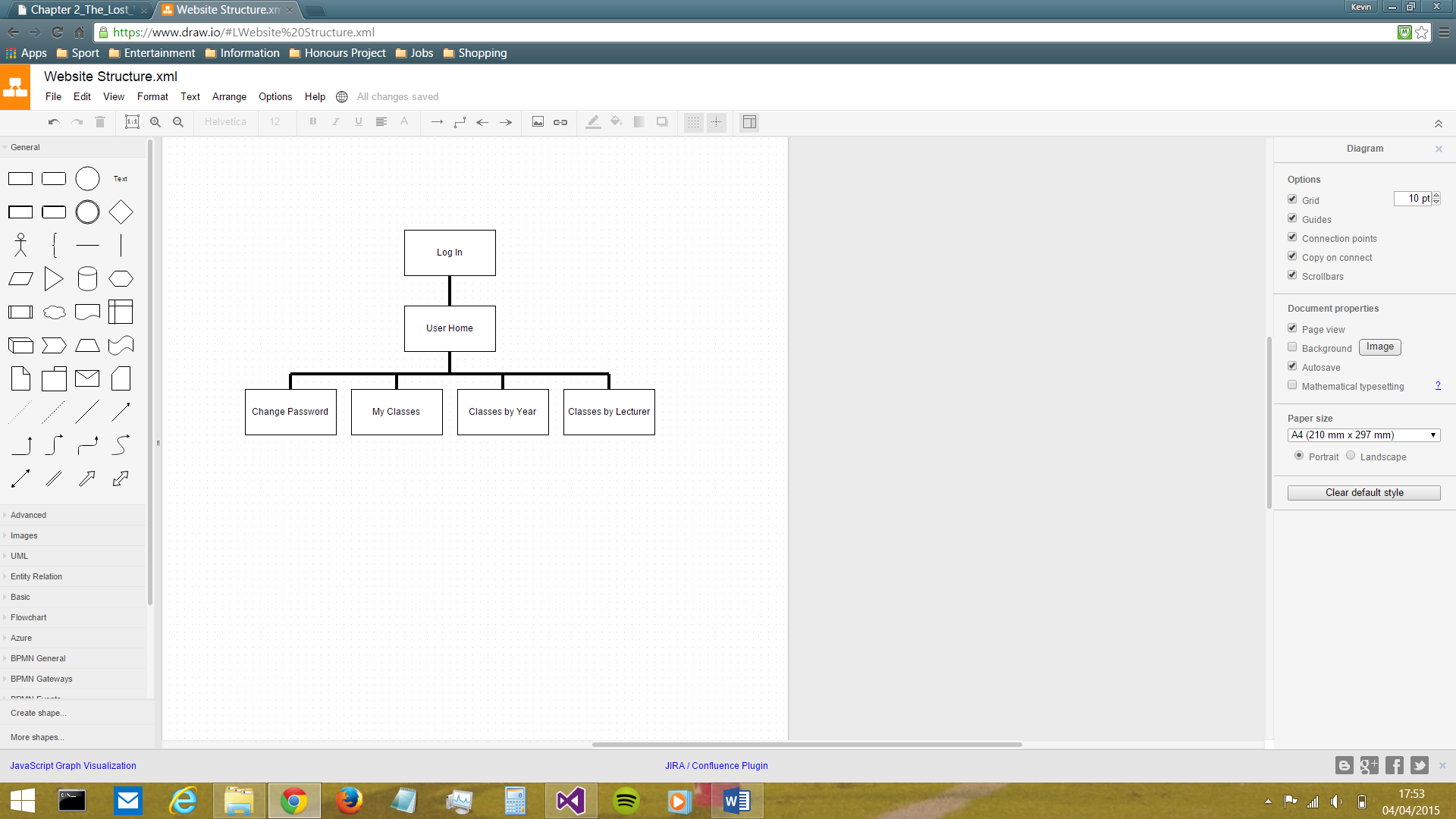


Figure 11: Website structure when logged in as a Dean

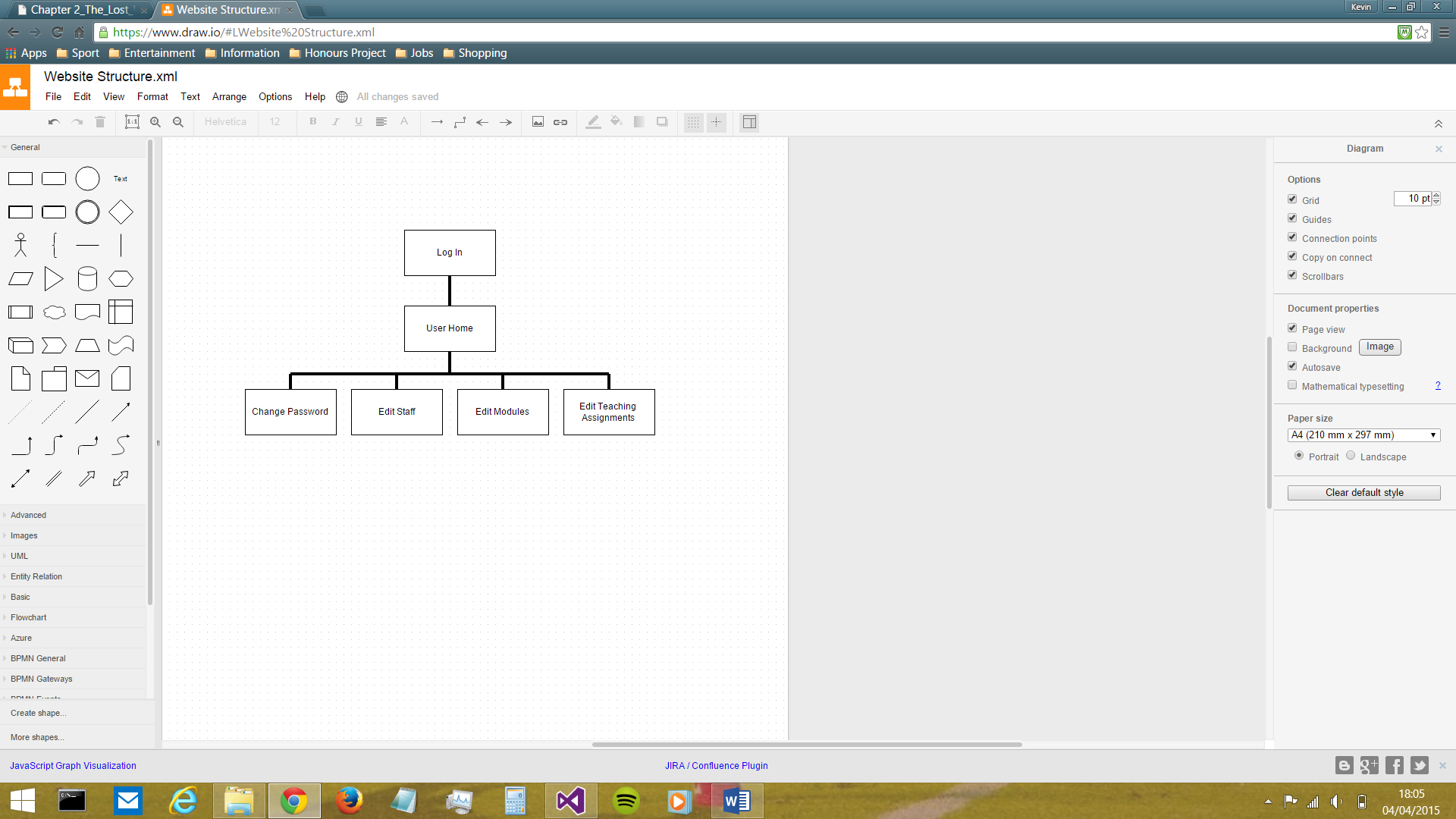


Figure 12: Website structure when logged in as an Admin

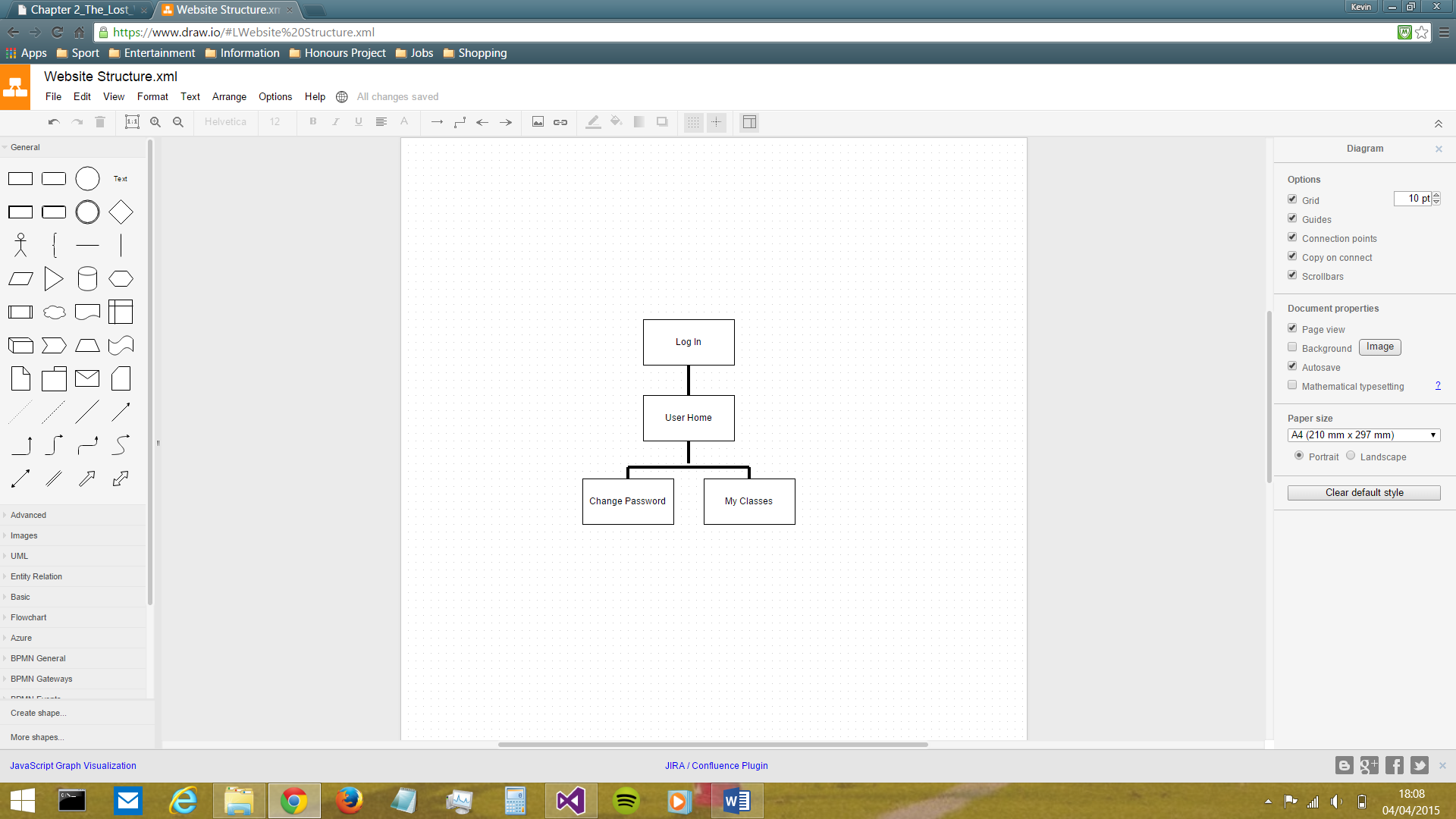


Figure 13: Website structure when logged in as a Lecturer

The login system was designed such that staff members would log in using their email and password which are stored in the RateMyClass database. These can be changed at any time by ‘admin’ members of staff, while each user also has the option to change their own password at any time as they so desire.

The differing levels of access were based upon the 3 staff types set out in the RateMyClass database: namely Lecturers, Deans and Admins. Lecturers have can only view their own feedback for classes they have conducted, whereas the Dean can view feedback for every class delivered in the School of Computing, reflecting the reality that the Dean alone is responsible for reviewing other staff members’ performance. The dean can also view the feedback for each year, in order to ascertain which, if any, programmes of study need looked at. Admin staff have an entirely different role from the other 2 – instead of using the website as a means of viewing feedback, they are responsible for the maintenance of the database through the medium of the website. They can perform CRUD operations on the 3 database tables which are out of the app user’s sphere of control, i.e. the staff, the modules and the teaching assignments.

In terms of the UI design, bearing in mind that the app and the website were intended to be viewed as a single system (i.e. ‘RateMyClass’) there was an almost unanimous agreement amongst potential end users that there should be strong coherence and consistency between the interfaces of both, in order to make the usage of both seamless and easy and facilitate transfer of knowledge (Satzinger & Olfman, 1998). Therefore, the minimalist greyscale theme from the app was brought forward to the website also (see Figures 14-23). Keeping the colour scheme greyscale has the benefit of ensuring users with Colour Vision Deficiency will have no problems reading the text (National Disability Autority (of Ireland), n.d.).

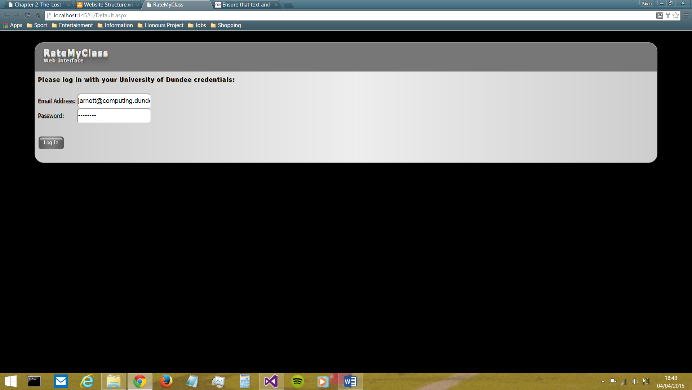


Figure 14: the 'Log In' screen

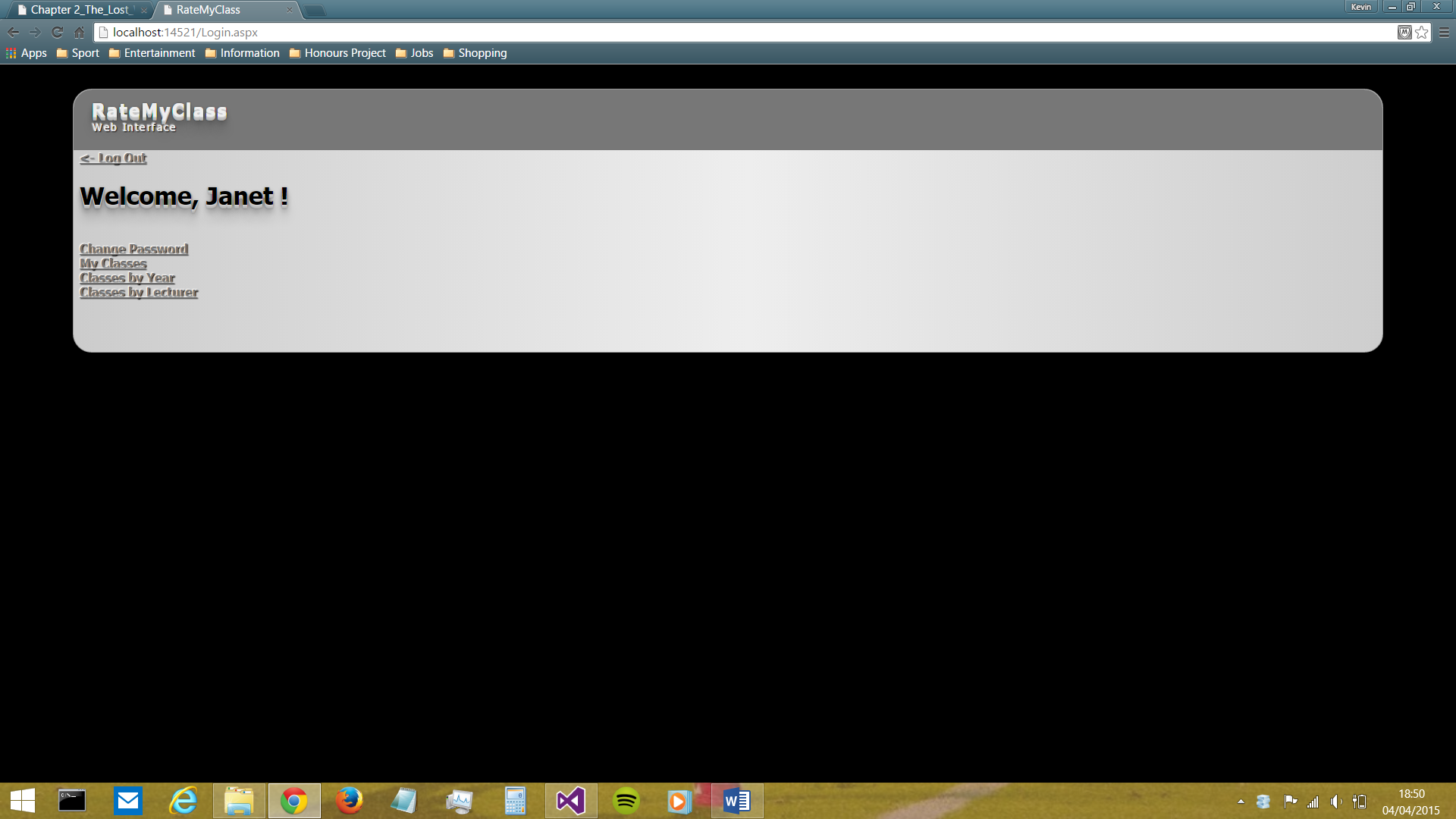


Figure 15: the 'User Home' screen

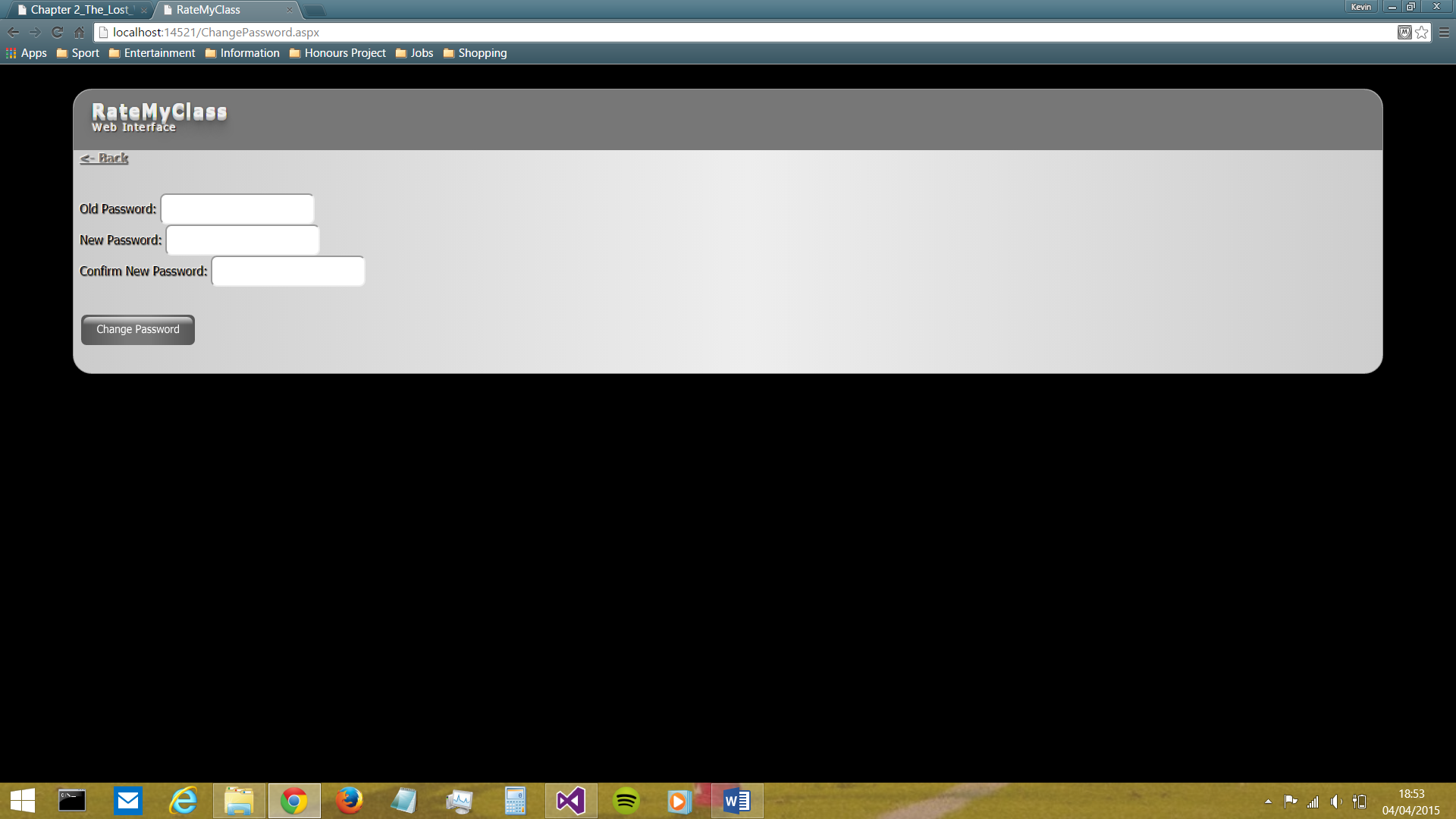


Figure 16: the 'Change Password' screen

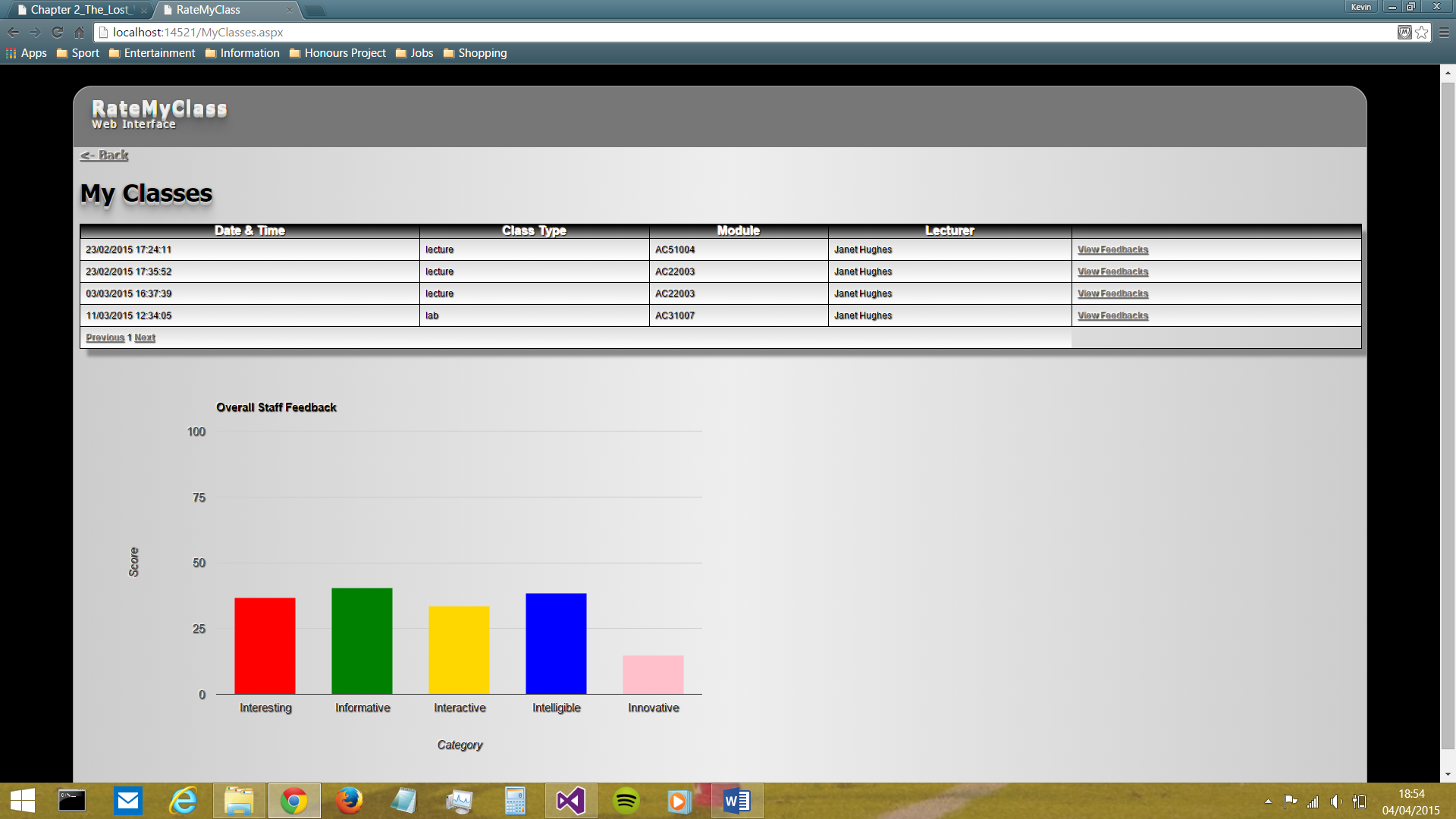


Figure 17: the 'My Classes' screen

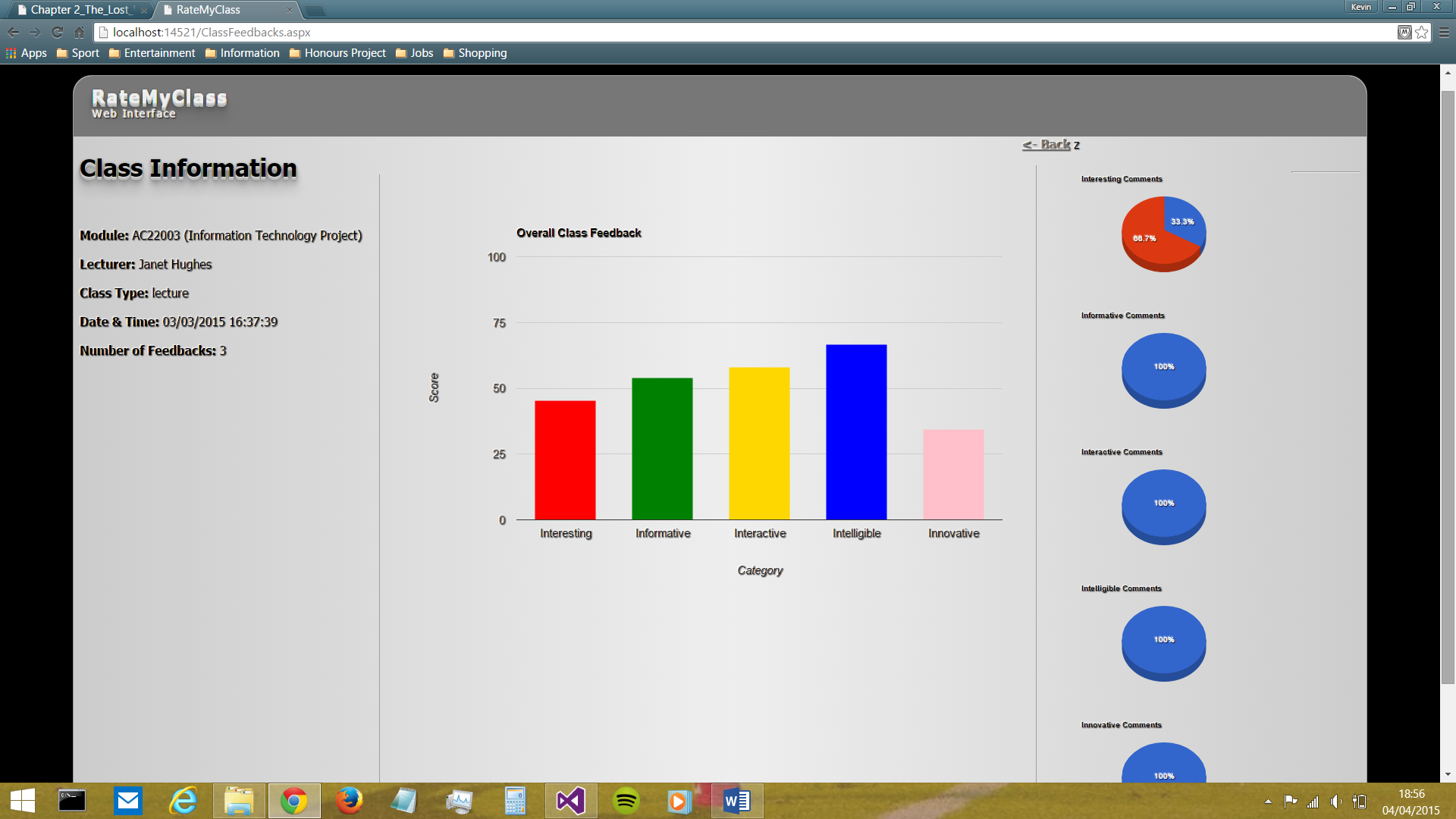


Figure 18: the 'Individual Class Feedback' screen

Figure 18 (above) shows the screen which displays the feedback information for a single class. During the design of this screen, staff indicated that, rather than showing the details of every individual item of feedback, it would be beneficial to portray the average feedback ‘score’ for the class. This was then taken forward, and a decision was made to display the average rating, along with the distribution of comments, for each of the 5 ‘I’s. In terms of the portrayal of these values, there was a choice to be made:

* Display ratings and comments in a linear fashion (e.g. in paragraphs or table(s))
  + Rejected as it was found that this method did not allow for an instant initial impression of the general nature of the data
* Use a form of visualisation to represent to the data
  + This was deemed the best option as these could utilise ‘pre-attentive attributes’ like difference in shape (e.g. varying heights of columns in column chart) to represent the differences in data visually, reducing the time and effort the human mind has to take in order to interpret the data (Few, 2004)

Taking this forward into the implementation, the Google Charts API was utilised in order to generate both a column chart to display the average ratings and 5 pie charts to display the distribution of comments across the 5 ‘I’s. This API was chosen as it was simple and easy to use, with a very good level of support available due to its popularity.

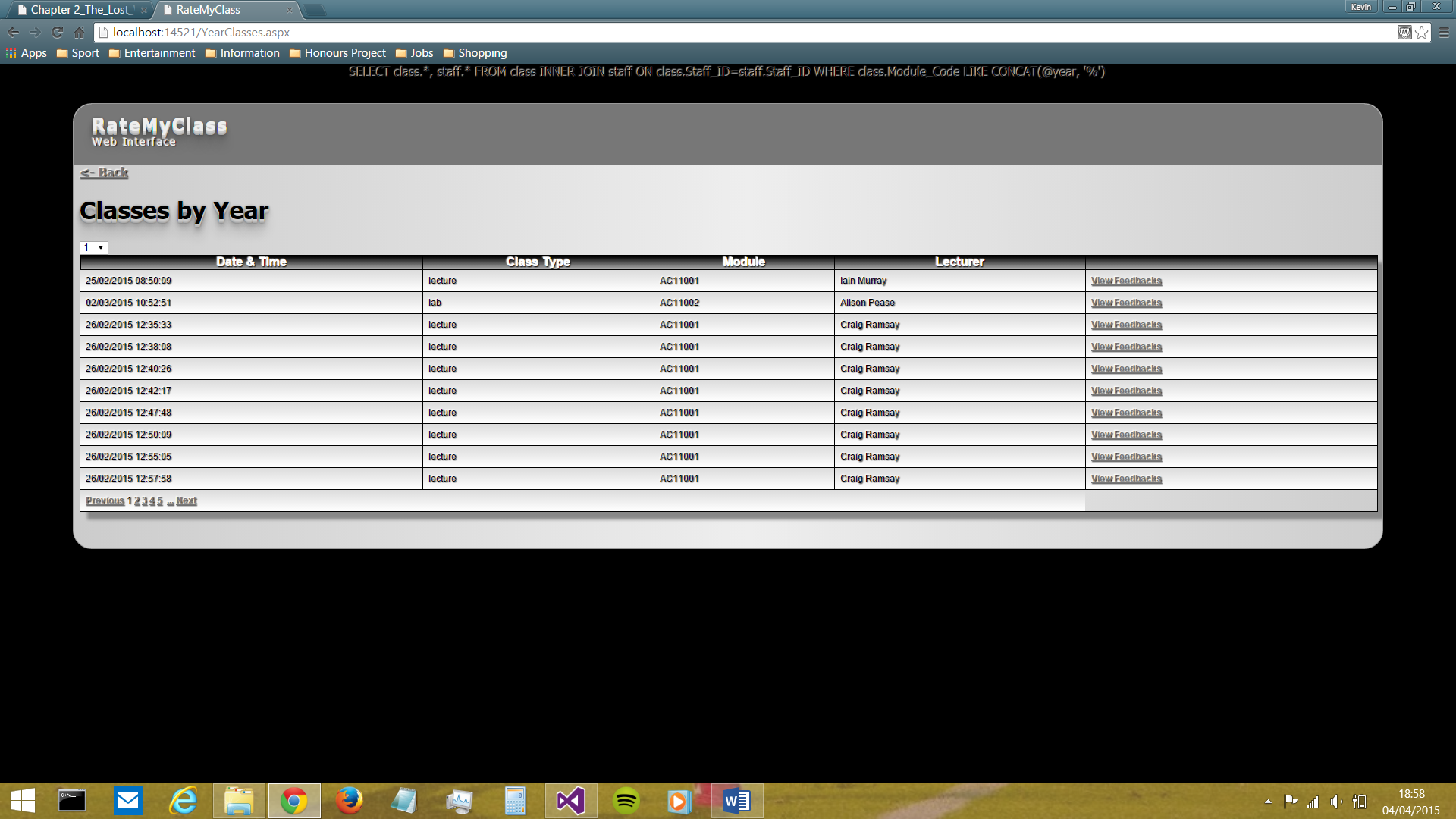


Figure 19: the 'Classes by Year' screen

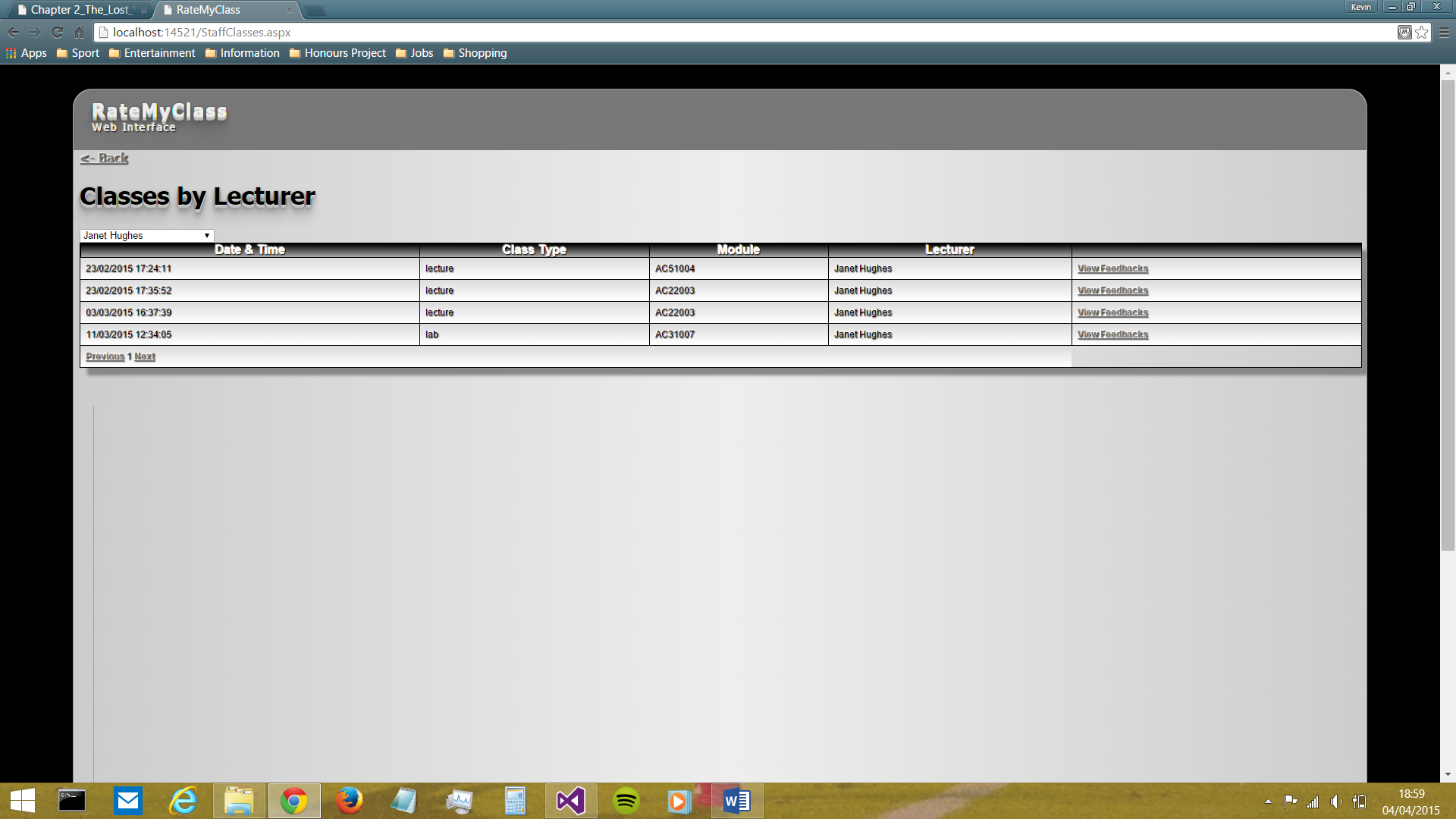


Figure 20: the 'Classes by Lecturer' screen

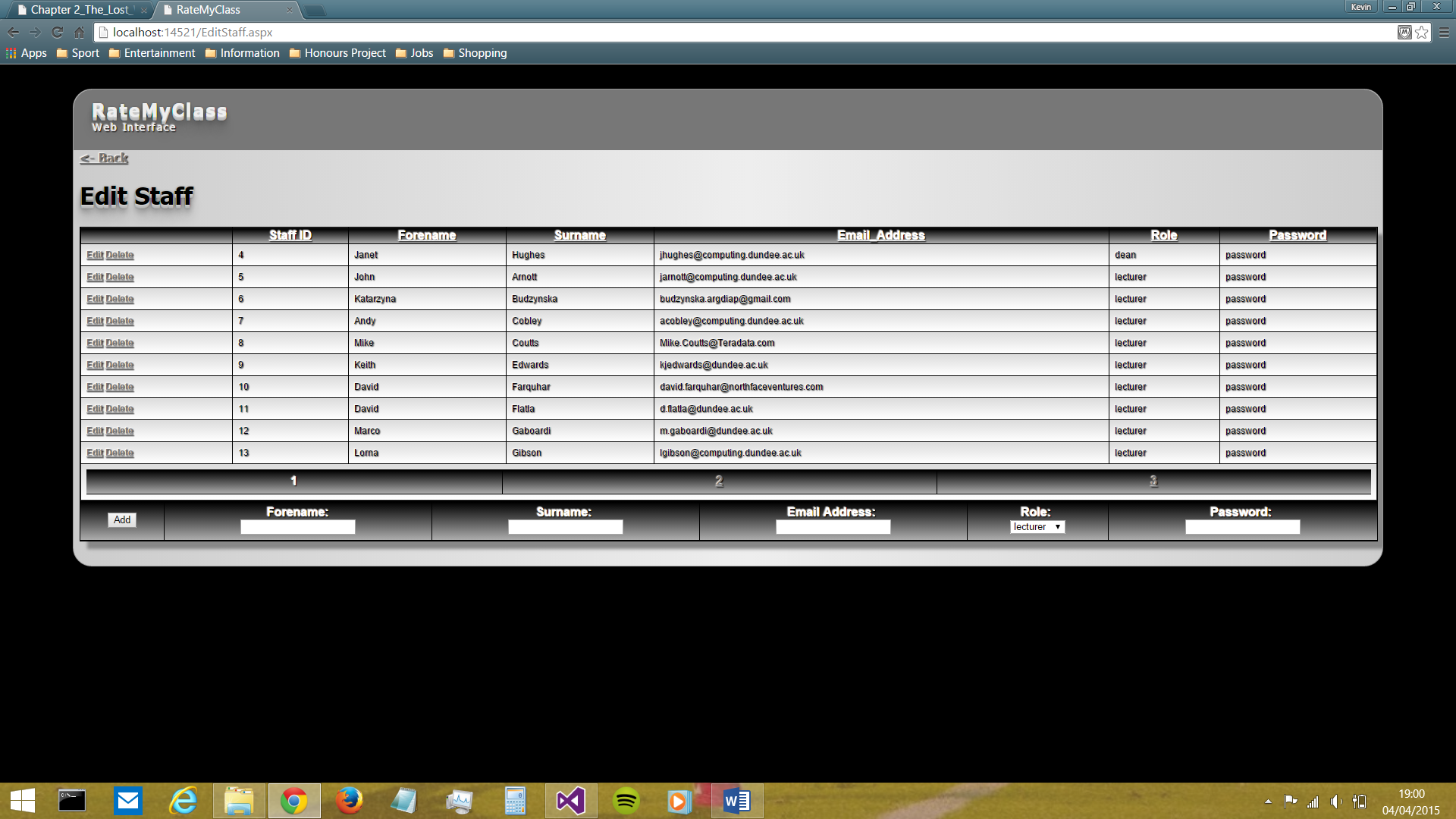


Figure 21: the 'Edit Staff' screen

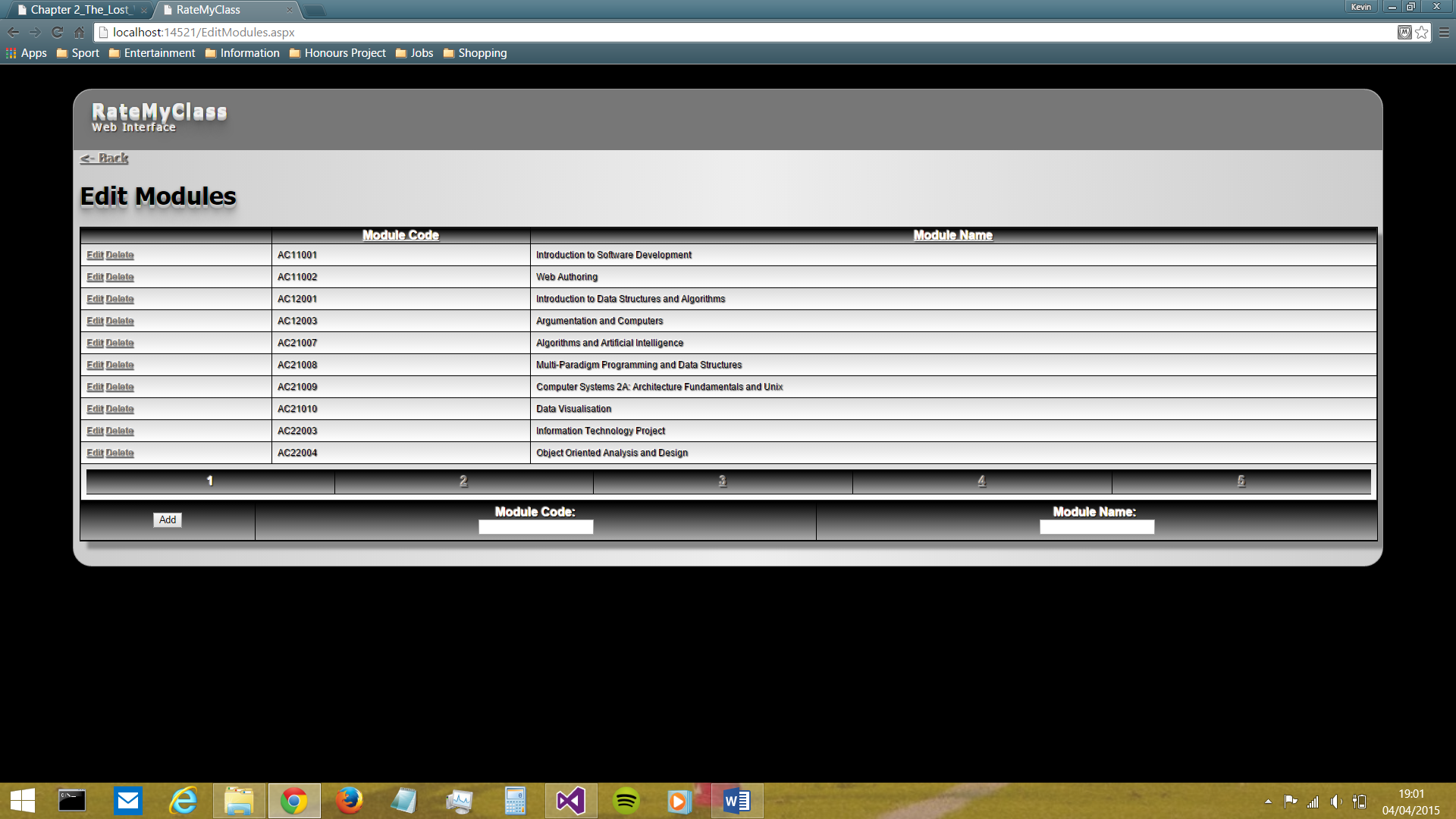


Figure 22: the 'Edit Modules' screen



Figure 23: the 'Edit Teaching Assignments' screen

The three screens shown above (Figures 21, 21 and 23) highlights the ‘edit’ facilities available to admin staff. The main problem involved in the design of these screens was how to include all of the CRUD functionality in one place, in way that did not require a large cognitive load to be place on the user. After much research, it was decided that the GridView C# class would be used, as it provides the means to display data from a data source (in this case, the RateMyClass database) in a simple, easy-to-read table whilst also allowing for automatic generation of edit and delete buttons adjacent to each record. This was a very useful way to bring ‘read’, ‘update’ and ‘delete’ functionality together into one place. The only thing left to provide was therefore the ‘create’ functionality, and this was included below each GridView in a simple input form. Feedback from staff regarding these screens was very positive, particularly regarding the simplicity of the design.

### Testing

### Evaluation/Retrospective

# Summary and Conclusions

# Acknowledgements

# References

Dufresne, R. J., Gerace, W. J., Leonard, W. J., Mestre, J. P., & Wenk, L. (1996). Classtalk: A Classroom Communication System for Active Learning. *Journal of Computing in Higher Education*, 3-47.

Few, S. (2004, September 4). *Tapping the Power of Visual Perception.* Retrieved from Perceptual Edge: http://www.perceptualedge.com/articles/ie/visual\_perception.pdf

Foth, M., Fitz-Walter, Z., Ti, J., Russell-Bennett, R., & Kuhn, K.-A. (2012). Please Take Out Your Phones: On the Spot Solicitation of Student Feedback in Class . *Proceedings of the 24th Australian Computer-Human Interaction Conference* (pp. 150-153). Melbourne: ACM.

Johnson, B. (2009, February 3). *Student Feedback Helps Teachers Grow*. Retrieved from eduTopia: http://www.edutopia.org/student-feedback-accountability-teachers

National Disability Autority (of Ireland). (n.d.). *Ensure that text and graphics are readable by users with limited vision or reading disorders*. Retrieved from The Centre for Excellence in Universal Design: http://universaldesign.ie/Technology-ICT/Digital-TV-equipment-and-services/guidelines-for-digital-tv-equipment-and-services/On-screen-interfaces/Ensure-that-text-and-graphics-are-readable-by-users-with-limited-vision-or-reading-disorders/

Satzinger, J. W., & Olfman, L. (1998). User interface consistency across end-user applications: the effects on mental models. *Journal of Management Information Systems - Special section: Managing virtual workplaces and teleworking with information technology*, 167-193.

Sharma, M. D., Khachan, J., Chan, B., & O'Byrne, J. (2005). An investigation of the effectiveness of electronic classroom communication systems in large lecture classes. *Australasian Journal of Educational Technology*, 137-154.

Theys, M. D., Lawless, K., & George, S. (2005). Tablet computers and the traditional lecture. *Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference* (pp. T2G-7). Indianapolis, IN: IEEE.

Zarraonandia, T., Aedo, I., Díaz, P., & Montero, A. (2013). An augmented lecture feedback system to support learner and teacher communication. *British Journal of Educational Technology*, 616-628.

# Appendices