

Project: Lecture traffic lights and feedback system  
Degree Scheme: G601 Software Engineering (With  
Integrated Year In Industry) MEng

Student: Morgan Jones (mwj7)

Supervisor: Chris Loftus (cwl)

Date: 06/01/2019

Version: 1.0

Status: Release

## Project Description

My Project is to build a lecture feedback system that will allow students to give feedback and ask questions anonymously throughout a lecture or workshop. The lecturer can then respond to this feedback during the session by re-visiting misunderstood material and/or answering questions posted by students. Any feedback given will be recorded in a database to be re-visited for review by a lecturer at some point in the future enabling the lecturer to easily see what was good/bad and adjust future content delivery accordingly.

The system will be a RESTful responsive web app accessible to both lecturers and students through mobile or desktop browsers. Students do not need to login to use the system. For lecturers to use the system to start a new session or review history they must be authenticated as staff via a login. Each member of staff can only view the history of the sessions they have started themselves. Lecturers will be able to start a session set to run until a certain time, for that duration a session code will be available for the students to submit feedback to that session. This code will be read out at the start of the lecture. Students can only provide feedback if they have a valid code for a currently ongoing lecture. Lecturers can end a session early if they want to.

After considering various technologies I have chosen to write the system using the Django web framework with a PostgreSQL database, this will require a web server and PostgreSQL database server to be deployed somewhere on the university network. The system must be capable of handling multiple lectures/workshops running simultaneously. I plan to use the CSS framework bootstrap 4 to more easily make the app responsive. I plan to use JavaScript and the HTML canvas to display some of the data in a visual way. Git and GitHub will be used for version control and back-up.

After considering multiple options I have decided to build my project using a personalised version of the Feature Driven Development agile methodology. The requirements and high-level system design will be produced at the start after understanding the domain and then a series of iterations will be undertaken to produce the chunks of the system's functionality. Lower-level design, implementation and testing will be done simultaneously through each of these iterations. Iteration work will each have its own feature branch and will be committed to the master upon being fully implemented and tested. Iteration reflection will be done informally through notes in my project diary and discussion with my supervisor.

## Proposed Tasks

- Learn the Django web framework [1]
- Learn about LDAP and use of an API that allows me to use it for staff authentication [2]
- Revise Bootstrap 4 [3]
- Revise PostgreSQL 11 [4]
- Design the user interface
- Produce a Requirements Specification
- Write higher level UML design for the overall system [5]
  - Use-Case Diagram
  - Component Diagram
  - ER Database Diagram
- Express web app as set of URLs with HTTP methods and response codes
- Build lower level UML design
  - Class Diagrams
  - Sequence Diagrams

- UI and Usability testing documentation will have to be produced [6]
  - Test Plan, Test Scenario, Test Case
  - Traceability matrix to map testing to requirements
- Build the login functionality
- Design and build the database
- Build the history view functionality
  - Build nice data graphs using JavaScript & HTML canvas [7]
- Build the session start functionality
- Build the student feedback functionality
- Preparation of the mid-project and final demonstration
- Work to write and collate for the final report
- Managing version control and backups [8]
- Maintain Record and Reflections in Project Diary

## Project Deliverables

- System Requirements Specification
  - Mirrors initial domain understanding at start of FDD [9]
- Higher-level component architecture, ER database diagram and use-case deliverables. Along with an expression of the system as URL routes with response codes?
  - This mirrors the initial modelling at the start of FDD. [9]
- List of main features to build (sections/chunks of work) and their order/dependency. This could be delivered/bundled with the previous deliverables.
  - 2<sup>nd</sup> and 3<sup>rd</sup> steps of FDD in one [9]
- Iterative Design and implementation of each feature/chunk of the system.
  - Lower level design will be done for each feature, this may cause the higher-level design to change if needed (low level class and sequence diagrams)
  - Code and unit testing will be done simultaneously
  - Possible Feature List (Currently unsure on order):
    - Login via LDAP functionality
    - Lecture History Retrieve, Update, Delete functionality
    - Lecture history nice JavaScript/canvas graphs addition
    - Lecture Create Session functionality
    - Student Provide feedback functionality
- Mid Project Demonstration (some way through building the features)
- UI Testing at key points – will require ethics form to be completed
  - List of tasks for tester to perform on both desktop and mobile
  - Half Dozen people, recommended to perform tasks
  - Each testers success or failure of the task is recorded
  - There will be a notes section for each task where the tester can make any comments or possible improvements about the UI components relating to the task.
- Usability Testing
  - Will require running multiple mock “lectures” each having students interacting through either a mobile or desktop device.
  - Need to book a room for this and have multiple volunteers at the same time.
- Final Report
- Final Demonstration

## Annotated Bibliography

- 
- [1] "Django documentation," Django Software Foundation, [Online]. Available: <https://docs.djangoproject.com/en/2.1/>. [Accessed 26 January 2019].

*The documentation for the Django web framework which I will use to write my web app*

- 
- [2] "LDAP library interface module," python-ldap project team, [Online]. Available: <https://www.python-ldap.org/en/latest/reference/ldap.html>. [Accessed 28 January 2019].

*Reading about a python module that provides access to LDAP C API, this could be used for staff authentication.*

- 
- [3] "Bootstrap Documentation Introduction," Bootstrap, [Online]. Available: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>. [Accessed 31 January 2019].

*The Documentation for the Bootstrap 4 CSS framework. Bootstrap will be used to make my app responsive.*

- 
- [4] "PostgreSQL 11.1 Documentation," The PostgreSQL Global Development Group, [Online]. Available: <https://www.postgresql.org/docs/11/index.html>. [Accessed 26 January 2019].

*Documentation for the PostgreSQL RDBMS. Postgres will be used as my web apps database.*

- 
- [5] "Visual Paradigm," Visual Paradigm, [Online]. Available: <https://www.visual-paradigm.com/>. [Accessed 30 January 2019].

*Used to download free modelling software and learn how to use it to produce my system designs.*

- 
- [6] "Software Testing - Documentation," Tutorials Point, [Online]. Available: [https://www.tutorialspoint.com/software\\_testing/software\\_testing\\_documentation.htm](https://www.tutorialspoint.com/software_testing/software_testing_documentation.htm). [Accessed 2 February 2019].

*How to write good testing documentation for my UI & Usability testing.*

- 
- [7] "HTML Canvas Drawing," Refsnes Data, [Online]. Available: [https://www.w3schools.com/graphics/canvas\\_drawing.asp](https://www.w3schools.com/graphics/canvas_drawing.asp). [Accessed 30 January 2019].

*Reference for HTML/CSS/JavaScript., used for detailed revision of canvas manipulation using JavaScript.*

- 
- [8] "Become a git guru," Atlassian, [Online]. Available: <https://www.atlassian.com/git/tutorials/>. [Accessed 31 January 2019].

*Tutorials used as a reference to revise git usage. Git is used for my version control & backup.*

- 
- [9] S. Goyal, "Major Seminar On Feature Driven Development," [Online]. Available: <http://csis.pace.edu/~marchese/CS616/Agile/FDD/fdd.pdf>. [Accessed 28 January 2019].

*Details of Feature Driven Development agile methodology. Used as reference to map my project plan to my process.*

- 
- [10] "Pipenv & Virtual Environments," [Online]. Available: <https://docs.python-guide.org/dev/virtualenvs/>. [Accessed 30 January 2019].

*Learning how to set up python virtual environment to only install packages needed for my app locally to the project.*

- 
- [11] "pip freeze," PyPA, The Python Software Foundation, 23 January 2019. [Online]. Available: [https://pip.pypa.io/en/stable/reference/pip\\_freeze/](https://pip.pypa.io/en/stable/reference/pip_freeze/). [Accessed 31 January 2019].

*Revision of pip for installing python packages & pip freeze for outputting a text file containing dependencies. This can be used to easily set up python environment when used on another machine.*

- 
- [12] "CSS - Make it Pretty!," Django Girls, [Online]. Available: <https://tutorial.djangogirls.org/en/css/>. [Accessed 30 January 2019].

*Tutorial on using bootstrap with Django.*

- 
- [13] "Stack Overflow," Stack Exchange, Inc., [Online]. Available: <https://stackoverflow.com/>. [Accessed 26 January 2019].

*Stack overflow used for many queries in multiple areas.*

---