**CS3 Capstone Projects**

Stage 3 – Progress Report

Date of report: 11/09/2017  
Report period: 18/08/2017 – 11/09/2017  
Name of project: Student Marks System  
Name of client: Aslam Safla  
Names of team members: Claudia Grindlay, Andre Barreiros, Daniel Kruyt

* 1. **Task Activities during the report period**

Front-end:

* Selecting templates and basic design – Claudia Grindlay:

The initial design of the user interface was drawn up and Bootsnipp templates were found and edited according to design. The majority of the functions that need visual representation, such as the login page, search functions and table design, were created using HTML, CSS and JavaScript.

* Building the UI – Daniel Kruyt:

Using the templates and the initial design to compose various screens and views. This includes the homepages for both student and non-student users, and the data tables displaying the information stored in the database and presented using the system.

* Functionality of the UI – Andre Barreiros

Functions such as switching between pages, links that can be clicked and have some function and filtering data are complete. There are certain aspects of the functionality that are not fully implemented yet, for example, login validation. There is a form of authentication in place, but full authentication has not yet been implemented.

Back-end:

* Initial setup of the Java Spring Boot framework – Daniel Kruyt

Deciding on and then setting up the framework to be used in the system. This includes developing an understanding of how the framework works and how to connect it to the rest of the system.

* Analysis Class Models – Andre Barreiros

Using the analysis class models created during stage 2 and doing the initial implementation in terms of programs. This includes creating the semantic classes that will become the objects created using the data from the data base and sent through the Spring framework to the front end; including methods and hierarchy. These classes have been created, but not yet implemented.

* Model View Controller pattern – All

Working on the separate components of the MVC pattern and starting to connect them. The creation of many controllers to deal with objects. This is not yet functional and integrated into the entire system.

Project Manager Responsibilities:

* As well as the front and back ends of the system for the prototype and demo, a significant amount of time has been dedicated to the planning tasks in stages 1 and 2. These include setting out the scope, risks and roles associated with the project, as well as developing analysis and class diagrams to visualise the structure of the system. Also necessary were the planning aspects like Gantt charts, use case narratives and constructing test cases.

Testing:

* Testing preparation:

Each new element or functionality that is created is first tested by the team member who created it. Then once the member is happy with its functionality or look, they add it to the master branch of the shared git repository where it is tested with the rest of the system to see if it works. If it does, it is committed to the master branch. If not, this is reported to the member who created it and removed from the master branch.

* Testing outcomes:

Currently, the only tests that are required are checking that everything works within the system that is set up, that it compiles and looks as expected. Moving forward, the testing gets more rigorous and it is integral to set up specific tests for each element in the system

Minutes:

* First meeting:
* The requirements for the system are set out by the client, most importantly the types of users and the layers of UI
* Types of users: administrator, admin staff, lecturer, course convenor, TA and student, each with a specific level of security and rights
* Layers: it is necessary to split up the view of the tables of data into layers, so that the first layer has final marks and more detail is revealed as the layers increase.
* Roles are assigned: Andre as Team Leader, Daniel as Architect and Claudia as Communicator.
* Second meeting:
* Clarification on the structure of the system, including how the layers work, and the expectations of the system.
* More of the technical backend requirements were discussed with the client.
* Team meeting after the client meeting allowed for a touch base of where each member is in their development and to ensure that everyone is on schedule.
* Third meeting:
* Mostly the demo is discussed, including the necessary requirements and functionality for the prototype.
* Given the type of prototype, mostly frontend is expected with the existence of the backend, but not full implementation or functionality.
  1. **Planned activities for the next report period**

Frontend:

* Finalise design:

Finalising all frontend design aspects of the system; perfect the ‘look’ of the UI. Improve upon any design decisions made for the first stage of building the UI, based either on personal taste or change more to the approval of the client. An important design function that is necessary is the display of the marks in tables, and the decisions that need to be made based on the best format for this.

* Add functionality:

Complete implementing frontend functionality. This includes login authentication, which is important to ensure the security of the system, and that all of the necessary levels of access are adhered to. Another functionality that is necessary is allowing users with the appropriate rights to edit student marks.

* Connect with the backend

The most important activity moving forward is integrating the front and back end together and building the functionality between the two. This connecting the backend, where all of the data is stored and dealt with, to the frontend, where the data is simply displayed.

Backend:

* Set up database

The database needs to be set up and connected via the semantic classes and the controllers to the frontend. This includes setting up the passing of the SQL queries between the classes and the database.

* Set up Spring security

The main objective of this is to set up a proper system of authentication to introduce login validation and security. This is done using the Spring framework mentioned above. This will also set up what type of user the login is, therefore predicting which homepage screen needs to be shown by the UI.

* 1. **Problems**
* Setting up and understanding how the Spring framework works took more time and energy than expected
* More complexity than initially thought in aspects such as Spring security and database connections
* A necessary restructure of the system framework based on the discovery of the need for semantic Object Oriented classes in between the database and the Spring framework
* Wasted time on small errors in setting up the UI
  1. **Revised Schedule and Goals**

We are still on track with our original schedule and there is no need to create a new schedule or change any goals.