CS-M14 Industrial Project
Interim Report

Matthew James Lewis (523015)

March 21, 2012

Contents

1 Introduction

The aim of this document is to report on the progress that has been made on my industrial project since the submission of the *Methodology & Requirements Document* and the *Specification Document*on the 9thth of January 2012.

To analyse the progress made so far, I have decided to describe the architecture I chose for the system together with discussing the development of each prototype I produced according the plan I originally set out in the *Methodology & Requirements Document*. These details can be found in Sections 2 and 3 respectively.

Since the start of the project, I have become more aware of the risks involved with developing this system. As a result, I feel that it is necessary to update the risk analysis which was undertaken in the *Methodology & Requirements Document*. The updated risk analysis can be found in Section 5.

With significant progress already made on the project and with it currently on schedule for completion I have produced an updated time plan to illustrate this. This can be found, together with an updated Gantt chart, in Section 6.

2 Project Overview

2.1 Architecture

The architecture I chose to use while developing the system is loosely based on the Model-View-Controller (MVC) architecture. The MVC in architecture, in its simplest form, is where the user interface or *view* of the application is developed separately from the actual application logic.
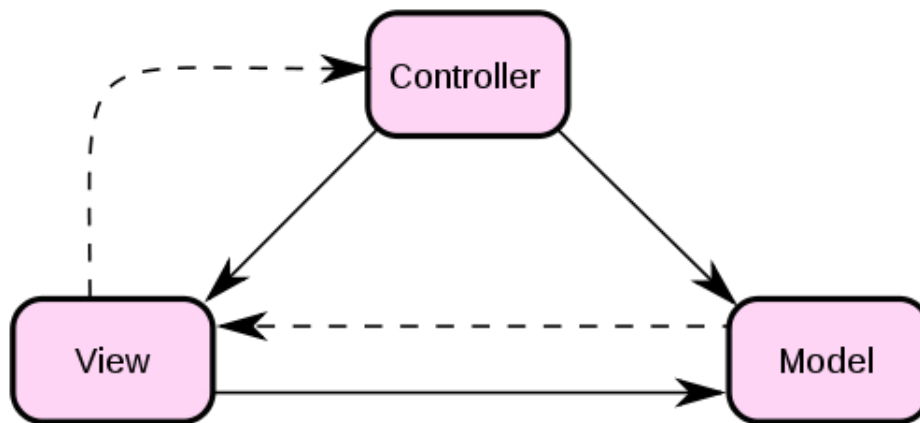
Figure 1: Model-View-Controller Software Architecture [**?**]

Model The part of the application is where the main functions of the application are contained. For example, if interaction with a database is involved, it is where the various transaction functions are defined.

Controller The Controller is involved with processing the interaction with the users of the system. After receiving an input from a user it uses the relevant model to perform certain actions.

View The part of the architecture is the user interface of the application. The data displayed on the view is 'pushed' from the model to the view.

However, for this project I have decided to modify the MVC architecture slightly by only using controllers and views. Each web page of the system that is accessible on the web can be seen as a *controller* that performs the program logic and the retrieval/updating of data. The controller then displays the appropriate view to the user.

2.1.1 Data Access Objects

Due to the centre's existing database structure and systems it is a requirement that the new system is compatible with the current database structure. Therefore, as a result of this, I have decided to produce numerous Data Access Objects (DAO) to provide an abstract interface for interacting with the database.

Each DAO class contains all the numerous database operations required for a specific table such as querying, insertion or deletion of records. For example, the DAO used to interact with the storage samples in the database is shown in **Listing ??**.

```
class SampleDAO {
  public function getUnsubmittedSamples($swanCode) {...}

  public function getSubmittedSamples($swanCode) {...}

  public function getCompletedSamples($swanCode) {...}

  public function deleteSample($sampleID) {...}
}
```

Listing 1: Sample Data Acces Object

As shown, the *SampleDAO* class contains a number of functions which are all based on a single table or a related tables. As another example, the DAO for *Students* contains the functions in **Listing 2**.

```
class SubmitterDAO {

    public function getSubmitterByCode($submitterCode) {...}

    public function getSubmitters($swanCode) {...}

    public function removeSubmitter($submitterCode) {...}
}
```

Listing 2: Submitters Data AccessObject

Again each of the functions defined in the Data Access Objects are not concerned about the application logic. They contain only the functionality for performing a specific query on the database and returning the result as shown in the example function in Listing 3 .

```
public function getSubmitterByCode($submitterCode) {

   $sql = sprintf("SELECT * FROM 'submitters' WHERE
        'Submitter_code'='%s'", $submitterCode);
```

```
        $result = Database :: performQuery ($sql);

        if (Database :: countRows ($result) == 1) {
            return Database :: fetchAssoc ($result);
        } else {
            return false;
        }

}
```

Listing 3: Function for retrieving submitter data.

By using this architecture, it has separated the program logic from the database structure, allowing changes to be made to the database structure without changing the logic of the application. For example, using the previous function, if the method for storing the submitters changed, only the query would need to be changed.

3   Project Progress

As outlined in the initial planning of the project I decided to divide the project into 6 prototypes, each one adding additional functionality until the project is completed and all requirements were met. The details of each prototype are outlined below.

Prototype 1 The first prototype will concentrate on designing and producing a new modern layout that the application will use.

Prototype 2 This prototype will be used to introduce the authentication and session functionality. It will also be used to add the 'submitters' functionality.

Prototype 3 The next prototype contains two major sections of the application. First it will add the ability for customers to submit new samples and then it will introduce the sample lists allowing the customers to view all of their samples.

Prototype 4 The fourth prototype will add all the management functionality required by the staff at the centre to move the samples through each stage of processing.

Prototype 5 The fifth prototype will be used to introduce the ability to attach result files to a sample allowing the customer to download or view them.

Prototype 6 The final prototype will add the drawing tool to the sample submission page.

As initially planned I am currently completing on the 5th prototype before moving onto the final prototype. In this section I will discuss the development stage of each prototype detailing any difficulties that were experienced. I have also included a list of the requirements that each prototype has met.

3.1 Prototype 1

The first prototype involved producing a new modern layout to replace the out-dated existing layout currently used in the existing system. This involved designing a new universal HTML & CSS layout to be used on each of the pages on both the customer web application and the administration web application. I first decided upon a light grey and blue colour scheme for the site before moving onto designing the HTML structure of the pages.
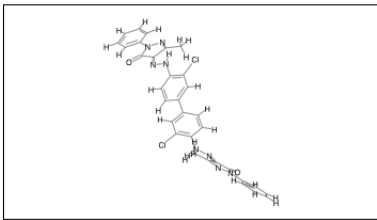

Figure 2: New HTML/CSS Layout

A major concern when developing this first prototype was to ensure that the new layout was compatible with all major browsers so that the system displayed identically in each one.

Overall, I feel that the new design for the system is in-line with todays standards but still keeps a simple and clean look and feel.

Requirements Met

ID	Requirement Description	Status
NFREQ1	The application must be accessible on the world wide web.	Complete
NFREQ2	The application must be compatible with all major web browsers.	Complete
NFREQ4	The application must provide a simple user interface with a modern look and feel.	Complete
NFREQ6	The application must use modern web technologies	Complete

3.2 Prototype 2

The next prototype was used to introduce the authentication of the centre's customers onto the new system. This involved producing a new login screen for the system together with the logic for authenticating user's credentials against those stored in the database.

This prototype was also used to implement the 'submitters' functionality allowing users to add, edit and delete submitter accounts. The submitter accounts are used to identify individuals within an organisation so that they can identify who submitted each sample to the centre. During prototype 3 the ability to choose what submitter to submit a sample under was added.

While adding the submitters functionality I found the existing database structure for storing submitters to be problematic and, with permission from the centre, I modified the structure.

Figure 3: Customer Login Screen

Requirements Met

ID	Requirement Description	Status
REQ1	The application must allow customers to log in to the system using their account details.	Complete
REQ2	The application must provide the ability to add or remove submitters from the customers account.	Complete
REQ4	The application must allow customers to submit samples under different submitter identities.	Partially

3.3 Prototype 3

This prototype has required the most amount of time during the development of the project as a whole. It contains two major parts of the system; the ability to submit new samples and the lists of all samples already submitted.

I began with creating a simple and functional form for the customers to complete when submitting a new sample. This form requires that the user enters all required data about their sample while also allowing them to decide which services they require from the centre. This form can be seen in Figure 4.

Along with entering basic data about their sample the customers can also provide a sample structure file (.mol). Once uploaded, the sample structure is displayed graphically in the form. An example of the system displaying a chemical structure file can be seen in Figure 2.

Figure 4: New Sample Form

The next major part of the system I introduced in this prototype was the ability to view the lists of samples and to be able to filter them by certain criteria. Each sample can be in one on five stages; unsubmitted, submitted, received, in lab and complete. For each one of these categories there will be a list of samples that are currently at this stage. In order to organise the 5 sample lists I introduced a tab bar with 5 tabs, one for each stage.

Figure 5: Tab bar displaying five sample categories

Clicking on each of the tabs displays the list of samples that are currently at that stage in the lab. However, as it is possible for there to be hundreds of samples in each list I added the ability to filter the samples to within a specified date range.

Figure 6: Filter bar allowing for filtering samples

Along with the ability to filter samples I also added basic pagination to the sample lists. This allows the user to choose how many samples they wish to view per page with the rest being accessible by selecting a page number. If desired, it is also possible to view all the samples in one list.

Requirements Met

ID	Requirement Description	Status
REQ3	The application must allow customers to submit new samples for analysis.	Complete
REQ4	The application must allow customers to submit samples under different submitter identities.	Complete
REQ5	When submitting a new sample, the application must ensure customers enter various required information about the sample.	Complete
REQ6	The application must allow customers to track the progress of any samples they have submitted.	Complete
REQ7	The application must allow samples to be created and saved so that they can be edited and submitted at a later date.	Complete
REQ8	The application must provide functionality for drawing or uploading a chemical structure file for a submitted sample.	Partially
REQ9	The application must provide a list of all the customers samples in an organised manner.	Complete
REQ10	The application must allow the sample list to be sorted or filtered by various parameters.	Complete

3.4 Prototype 4

The fourth prototype was designed to add all the management functionality required by the staff at the centre in order for them to move the samples through each stage of processing. For example, moving a sample from the 'In Progress' stage to the Completed Stage.

When moving a sample from 'Submitted' to 'Received' the user assigns the sample a sample code that is used internally by the centre to uniquely identify samples. This code is also attached to the physical sample received via post and essentially ties the online data to the sample vial so that it can be tracked while at the lab.

To move a sample from 'Received' to 'In Progress' involves finding the record on the web application and setting it as 'In Progress' by clicking a button. This then moves the sample into the 'In Progress' list and will allow the staff at the centre to attach analysis results to the sample record. However this functionality is currently being completed in prototype 5. While 'In Progress' the staff can also record data about what analysis techniques have been performed on the sample. This can be seen in Figure 7.

Figure 7: Recording analysis techniques used

At each stage I have also allowed for a sample to be moved back a stage in case any errors were made. It is also possible to unsubmit a sample if the submitter has missed any information that may be needed during the analysis.

Requirements Met

ID	Requirement Description	Status
REQ13	The application must allow the user to view all the samples submitted by a customer.	Complete
REQ14	The application must allow the sample list to be filtered by a time period.	Complete
REQ15	The application must allow the user to unsubmit samples.	Complete
REQ16	The application must allow the user to view the details of submitted samples together with the chemical structure that the customer provided when submitted.	Complete
REQ17	The application must provide functionality that allows the user to attach result files to a sample after it has been analysed by the lab	In Progress
REQ18	The application must allow the user to move samples through the various stages of processing.	Complete
REQ21	The application must allow the user to delete samples.	Complete
REQ22	The application must follow the labs current sample identification system when booking in and processing samples.	In Progress

3.5 Overall Progress

Although the project is required to be compatible with the current systems used by the centre, it has been straight-forward to develop. With extensive experience with PHP and MySQL, there have not been any major problems to overcome.

4 Requirements Progress

4.1 Funtional Requirements

Customer Web Application

ID	Requirement Description	Status
REQ1	The application must allow customers to log in to the system using their account details.	Complete
REQ2	The application must provide the ability to add or remove submitters from the customers account.	Complete
REQ3	The application must allow customers to submit new samples for analysis.	Complete
REQ4	The application must allow customers to submit samples under different submitter identities.	Complete
REQ5	When submitting a new sample, the application must ensure customers enter various required information about the sample.	Complete
REQ6	The application must allow customers to track the progress of any samples they have submitted.	Complete
REQ7	The application must allow samples to be created and saved so that they can be edited and submitted at a later date.	Complete
REQ8	The application must provide functionality for drawing or uploading a chemical structure file for a submitted sample.	Partially Complete
REQ9	The application must provide a list of all the customers samples in an organised manner.	Complete
REQ10	The application must allow the sample list to be sorted or filtered by various parameters.	Complete
REQ11	The application must allow for the customer to retrieve the results of any completed samples they have submitted either by downloading them or displaying them in a on-line viewer.	In Progress

Sample Administration Web Application

ID	Requirement Description	Status
REQ12	The application must allow for the lab staff to login to the system.	Incomplete
REQ13	The application must allow the user to view all the samples submitted by a customer.	Complete
REQ14	The application must allow the sample list to be filtered by a time period.	Complete
REQ15	The application must allow the user to unsubmit samples.	Complete
REQ16	The application must allow the user to view the details of submitted samples together with the chemical structure that the customer provided when submitted.	Complete
REQ17	The application must provide functionality that allows the user to attach result files to a sample after it has been analysed by the lab	In Progress
REQ18	The application must allow the user to move samples through the various stages of processing.	Complete
REQ19	The application must provide the ability for a new year codes to be started.	In Progress
REQ20	The application must allow the user to view samples for previous year codes.	In Progress
REQ21	The application must allow the user to delete samples.	Complete
REQ22	The application must follow the labs current sample identification system when booking in and processing samples.	In Progress

4.2 Non-Functional Requirements

ID	Requirement Description	Status
NFREQ1	The application must be accessible on the world wide web.	Complete
NFREQ2	The application must be compatible with all major web browsers.	Complete
NFREQ3	All code should be fully commented and documented.	Ongoing
NFREQ4	The application must provide a simple user interface with a modern look and feel.	Complete
NFREQ5	The application must be easily maintainable.	Complete
NFREQ6	The application must use modern web technologies	Complete

5 Updated Risk Analysis

As the project has progressed the risks associated with the project have changed. For this reason, I have included an updated risk analysis. For each risk identified I have also assigned a likelihood of occurrence (LO) from 1(low) - 5(high) and a potential impact (PI) from 1(low) - 5(high).

5.1 Non-technical Risks

The non-technical risks I have considered along with the steps taken to prevent them or action to be taken should they occur are outlined as follows:

ID	Risk	LO	PI
1	**Failure to adequately judge the time and resources required.** As the project is approaching completion and with it being on schedule, this risk is now significantly reduced since the beginning of the project.	1	4
2	**Underestimation of non-project-related workload** As lectures are almost complete and the amount of non-project-related workload is reduced, it is now possible to focus completely on finishing the development of this project. Therefore this risk is again significantly reduced.	1	4
3	**Illness** With lectures being complete allowing for more time to work at the project it is now less likely that illness will effect the project. However it still cannot be ruled out that I may suffer from illness during the remaining time left for the project which may cause the work rate to decrease. In my original plan, I allowed for enough time at the end of the project in case this problem arrises.	3	2
4	**Failure to meet requirements** With 5 of the 6 prototypes already complete and meeting requirements, it is highly unlikely that I fail to meet the remaining requirements in the final two prototypes. However, as this project is based on set requirements, any failure to meet them would have a serious potential impact.	1	5
5	**Failure to match customers vision of the project** As this project is being developed for a company not in the software business it is unlikely that they know exactly what they would like the application to look like and how it should behave. This could potentially result in the customer not being happy with the finished product. To combat this issue I have produced a number of partial prototypes of the application which have been evaluated after each has been completed. This has provided me with feedback on what has been produced so far and any modifications that were required have be made during development and not after the project has been fully coded.	2	3

5.2   Technical Risks

I have also identified a number of technical risks for the project and these are also outlined as follows:

ID	Risk	LO	PI

1	**Feature creap.** To prevent this issue I produced an accurate and complete requirements which I have followed closely while developing the project. It has ensured that all required features are implemented first before adding any additional functionality if time permits.	1	2
2	**Compatibility with all major web browsers** A major issue when developing web applications is the wide range of web browsers used by people to access the world wide web. Each browser's engine works differently and can mean that one layout that works in Firefox, for example, may not work in Microsoft's Internet Explorer. While developing the layout for this project I thoroughly tested all page layouts on a multitude of browsers to ensure that the page is displayed the same in each one and all functionality works the same. I also used an online service that produces screenshots of a specified URL in a greater range of web-browsers. This allowed the layout to be checked even more thoroughly.	5	3
3	**Failure to adequately learn a required new technology.** I avoided this risk by choosing two technologies with which I have extensive experience with both academic and personal projects. I have also chosen technology that have are the most popular and have an extensive support base if any problems arise that I may not have experience with.	0	3

6 Updated Time Plan

6.1 Prototype Time Plan

Prototype	Task	Date
Prototype 1	New Layout	Complete
Prototype 2	Authentication & Sessions Submitters functionality	Complete
Prototype 3	Submit Samples Sample Lists	Complete
Prototype 4	Management Functionality	Complete
Prototype 5	Result files (download & viewer)	3rd April 2012
Prototype 6	Drawing tool	20th April 2012

6.2 Remaining Milestones

Milestone	Deliverable	Date
Milestone 2	Interim Report	30th March 2012
Milestone 3	Poster Presentation User Manual Design Document Testing Document Narrative and Reflective Account	4th May 2012

6.3 Gantt Chart

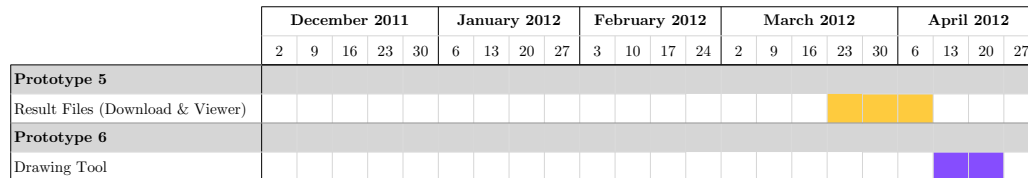|	December 2011					January 2012				February 2012				March 2012					April 2012			
	2	9	16	23	30	6	13	20	27	3	10	17	24	2	9	16	23	30	6	13	20	27
Prototype 5																						
Result Files (Download & Viewer)																	�	▮				
Prototype 6																						
Drawing Tool																				▮	▮	

Figure 8: Project Gantt Chart

7 Conclusion

Having currently completed four of the six prototypes which I had planned I feel that I have made excellent progress towards completing all the requirements that I set out in my *Requirements Document.*

Overall, with the fifth prototype almost complete, I feel that the project is still on track to be successfully completed on schedule with all features and functionality which were identified before beginning the project.