title: Functional Spec

Software Requirements Specification for Assignment Management System (AMS)

Prepared by: Muhammad Umar & Jack Doherty

Date created: 05/12/2019

Table of Contents

1. Introduction

1.1 Overview

The product being which we will be developing is an Assignment Manageent System (AMS), which will manage the secure submission, management and distribution of assignment marks in a GDPR-compliant manner. Users in the system will have one of two roles, teacher or student. Ideally, user authentication will use the University's LDAP system.

Through the system, students will be able to:

- Securely upload assignments. Uploaded assignments will be timestamped.
- Depending on the setting of the assignment, a student may be allowed multiple uploads of an assignment.
- See a calendar of all upcoming assignments.
- Receive notifications of assignments, reminders of deadlines, and notification of results.
- Retrieve the results of all their assignments is a GDPR-compliant manner.

Through the system, teachers will be able to:

- Set the settings for their assignments. This will include, but not limited to:
  - due date
  - number of allowed uploads
  - notifications
  - marking scheme
- Upload assignments for students who signed up for their course.
- Mark assignments or upload marks from standard file types (.xls, .csv, etc.). This may include feedback to the student.
- Generate analytics of each module.

1.2 Scope

This system is being desigend for managing assignment. This system can also be used by any college/school as for managing their students assignments and teachers can gain analytics from the submission and the grade of the uploaded assignments. We are developing this system with the help all the free available tools relevent to the development of this system.

1.3 Glossary

FRONTEND

- **HTML:** Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser.
- **CSS:** Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML.
- **BOOTSTRAP:** Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- **JAVASCRIPT:** An object-oriented computer programming language commonly used to create interactive effects within web browsers.
- **JQUERY:** jQuery is a JavaScript library that allows web developers to add extra functionality to their websites especially with DOM manipulation.

BACKEND

- **.NET CORE:** A cross-platform, high-performance, open-source framework for building modern, cloud-based, Internet-connected applications.
- **MYSQL:** An Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL)
- **APACHE SERVER:** A robust, commercial-grade, feature-rich, free, and open-source HTTP web server.

2. General Description

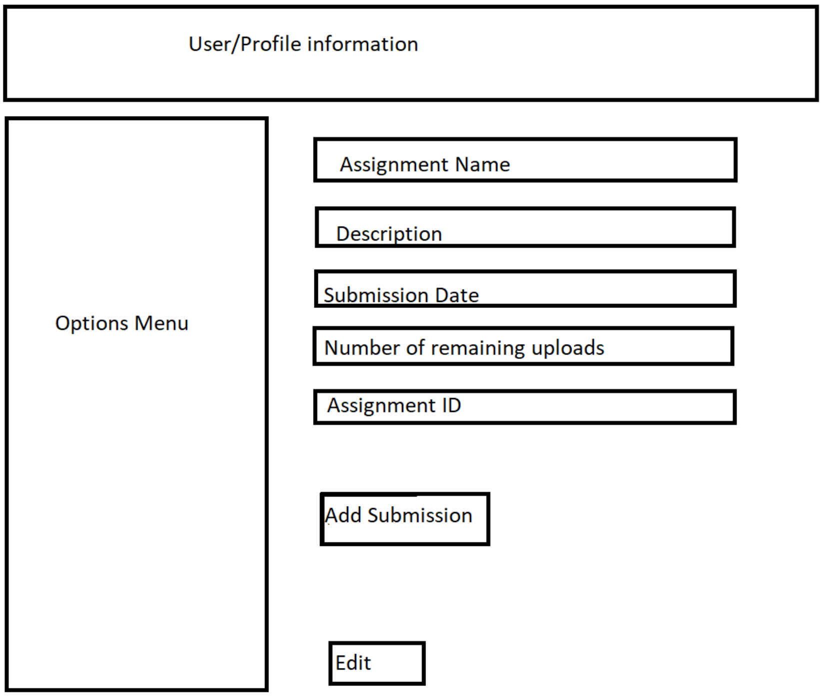
2.1 Product / System Functions

The following are the main functions of the Assignment managment system. These functions are prefatory and are liable to be edited or changed in the future.

- Register
- Recieve Notifications
- Login
- Upload Assignment
- Upload Assignment submission

- Retrieve Result
- Edit Assignment
- Edit Assignment submission
- Retrieve Calendar
- Retrieve Analytics
- Grade Assignment

2.2 Interfaces



What a possible assignment submission page may look like

2.3 User Characteristics and Objectives

The product is a website and so can be accesed by anyone with a computer and internet access. However As it is a DCU assignment managment system, users will be verfied in some way shape or form so that only DCU Students and Teachers may login and use any of the features.

The Website will be highly accessible featuring an easy to use GUI to facilitate incoming first year computing or DCU students in general with little to no computing knowledge.

2.4 Operating Environment

- The development of this webapp will be done in Windows environment.

- The deployment of this webapp will be done on an Apache server in a Linux environment.
- The website will work on any opreating system that has an web browser which they can use to access our webapp.

2.5 Constraints

The contraints we face for this project are as detailed below:

- Database memory Due to the limit of our SQL account there is a limit to the number of users we can service as there would be only so many asignments and submission that we could store.

- Time This project is one in which we could continuously expand and add features and extra functionality to. We think the deadline will be just to demostrate the base required functionality.

3 Functional Requirements

3.1 Register

- Description:

This function is what registers users to the website. A Link will be provided on the home page tkaing a user to the DCU intranet where they will be able to enter their DCU credentials and register to use the website.

- Critically:

This function is of course critical for the website as the AMS system is a DCU facility and so can only be used by DCU students and faculty.

- Technical Issues:

Use of DCU LDAP system to authenticate users.

- Dependencies None.

3.2 Recieve Notifications

- Description:

During the registration process users will be able to select whether or not they would like to recive notifications of upcoming deadlines of assigments by their college email they provided during registration.

- Critically:

This function is is kinda critical beacuase students will be depend on these notification and these notifications can play an important role.

- Technical Issues:

User must recognise emails coming from our system as regualar emails and not spams.

- Dependencies:

This is dependant on register.

3.3 Login

- Description: This is a user login script that prompts user for their DCU credentials and uses the DCU intranet portal to sign them into the website.

- Critically: This is a critical function as this will decied either a user is an admin, student or teacher.

- Technical Issues: Users will be automatically send to their designated main page. As admin, student and teacher will have different interfaces.

- Dependencies: This is dependant on register.

3.4 Upload Assignment

- Description: This function is only available to the Teacher account type. From here the Teacher account type can upload "assignments" for students to upload submissions to.

- Critically: Critical function as this is the main purpose of this system is to let students upload assignments.

- Technical Issues: Teacher and student uploading assigenmnts will have differnet behaviours, when teacher uploads an assignemnts it will create a submission page and he/she will set the deadline, upload limit etc of that assignmemnt submission page but when student uploads an assignment they will only be able to upload or reupload their assignment.

- Dependencies: This is dependant on register, login, uploading the submission page and being a teacher.

3.5 Upload Assignment submission

- Description: This function is only available to the Student account type. From here the Student account type can upload "assignments" for students to upload submissions to.

- Critically: Like the assigenmnets submission for teachers this will be assignment submission where students will upload to the

- Technical Issues:

- Dependencies: This is dependant on register, login, upload asignment and being a student.

3.6 Retrieve Result

- Description: This function enables the student account type to view their grade for a graded assignent submission
- Critically: Critical to the application as it allows students to view their grade for a submission.
- Technical Issues: None that are apparent at this time.
- Dependencies: This is dependant on register, login, upload assignment and upload asignment submission.

3.7 Edit Assignment

- Description: This function enables the Teacher account type to edit previously uploaded assignemnts. This enbales the user to change details of the assigment such as the due date, name etc.
- Critically: Not critical but important so that students do not have to delete a submission a resubmit in order to change the details of a submission.
- Technical Issues: None that are apparent at this time.
- Dependencies: This is dependant on register, login and upload assignment.

3.8 Edit Assignment submission

- Description: This function enables the Student account type to edit previously uploaded assignemnt submissions. This enables the user to change details of the submission such as the files, or reupload new files before the deadline.
- Critically: Another critical function as this will allow user to upload assigments many times, also keeping in mind the upload limit given by their teacher.
- Technical Issues: Database will be used heavily here as the user might make a lot of changes to their assignment.
- Dependencies: This is dependant on register, login and upload asignment submission.

3.9 Retrieve Calendar

- Description: This function enables the Student account type to view their calendar. This enables the user to view their upcoming deadlines for assignments
- Critically: This function not critical but is nice to have, so studnets can see their upcoming assignment which will be shown on a calander nicely.

- Technical Issues: Accesing info on all assignments from the database for certain user and diplay it on a calender under correct dates.
- Dependencies: This is dependant on register, login and upload asignment.

3.10 Retrieve Analytics

- Description: This function enables the Teacher account type to generate statistics regarding the previously uploaded assignment. This may include grades, submission times etc.
- Critically: Not at all critical but adds to the functionality of the website.
- Technical Issues: Teacher will either send the results by mail or on the webapp.
- Dependencies: This is dependant on register, login, upload assignment and have upload asignment submission.

3.11 Grade Assignment

- Description: This function enables the Teacher account type to upload a grade for a previously uploaded assignment submission. This may be done by marking the assignment via the website or by uplading marks from standard file types(.xls, .csv). This can also inlcude feedback/comments regarding the submission
- Critically: This function is vital to the application as Students need to be able to recieve grades for their submissions.
- Technical Issues: This will require Teacher user types to be able to upload marks via files.
- Dependencies: This is dependant on register, login, upload assignment and upload asignment submission.

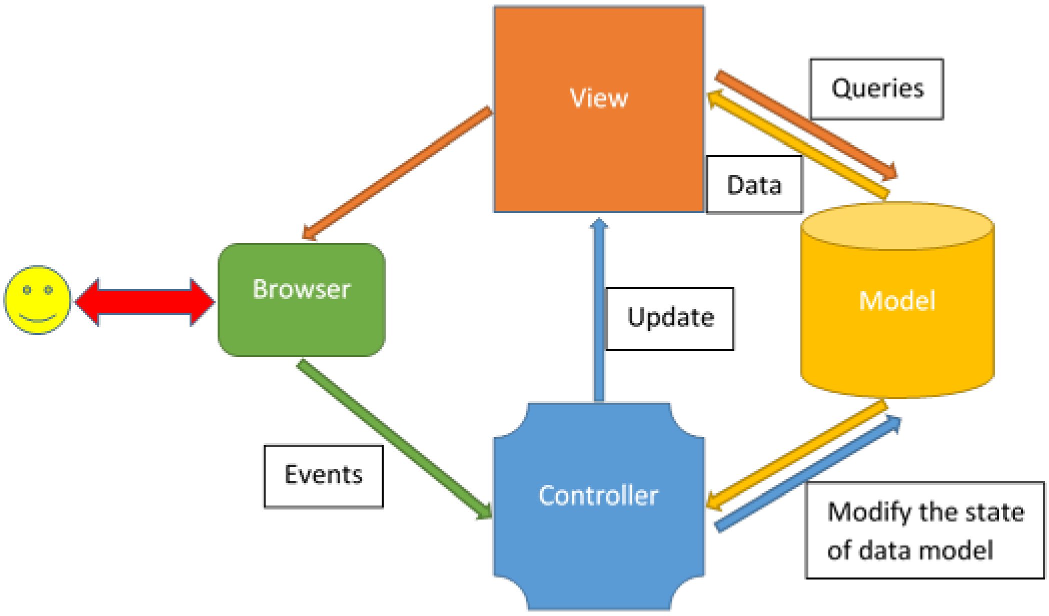
4. System Architecture

4.1 Backend

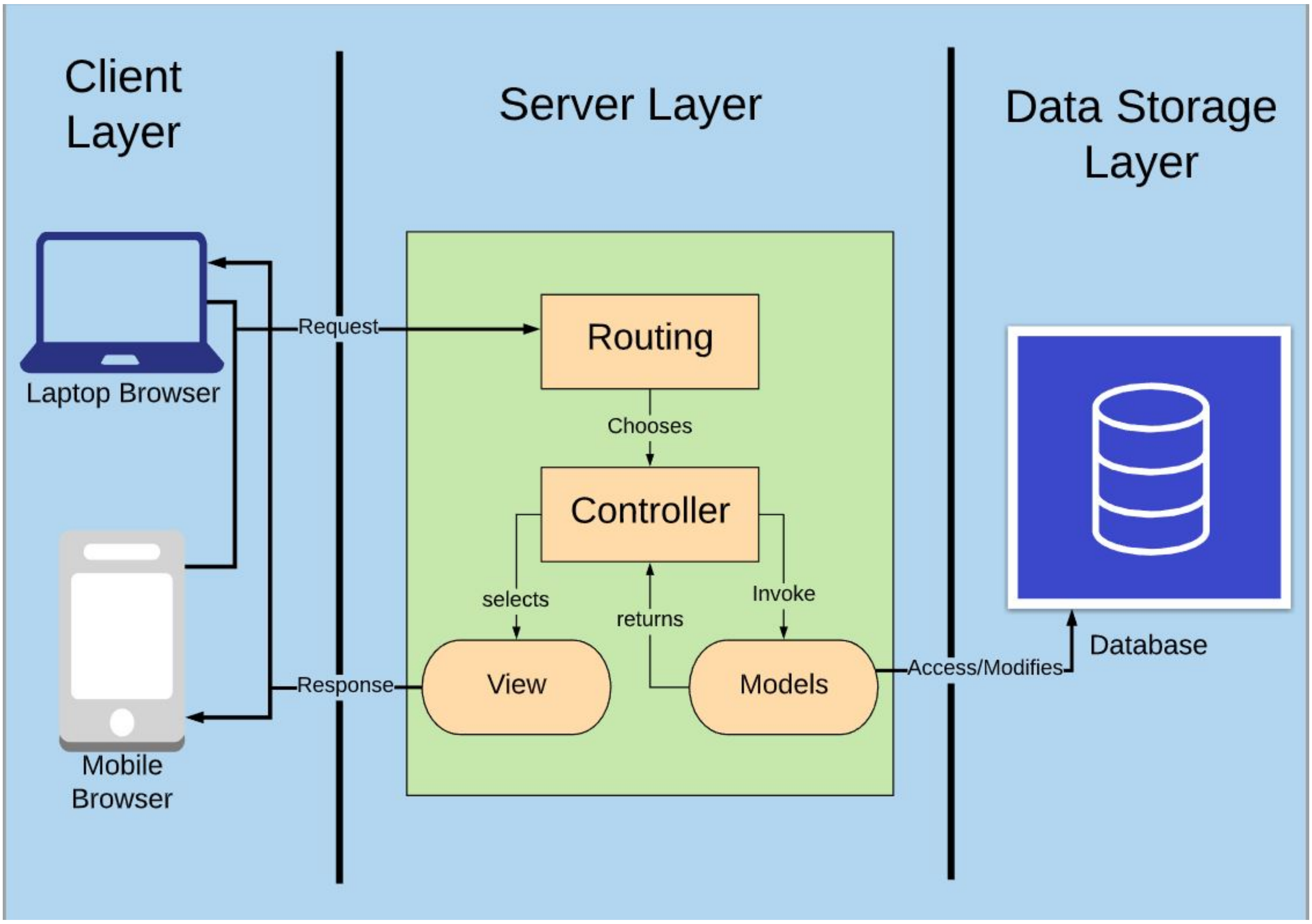
- We will be using .NET CORE Model-View-Controller (MVC) design pattern to implement our web app. This architectural pattern separates an application into three main components: the model, the view, and the controller.
  - **Models:** Model objects are the parts of the application that implement the logic for the application s data domain. Often, model objects retrieve and store model state in a database. For example, a Product object might retrieve information from a database, operate on it, and then write updated information back to a Products table in SQL Server.
  - **Views:** Views are the components that display the application s user interface (UI). Typically, this UI is created from the model data. An example would be an edit view of a Products

table that displays text boxes, drop-down lists, and check boxes based on the current state of a Products object. ate of a Products object.

- **Controllers:** Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn queries the database by using the values.



- Image of our system architecture.

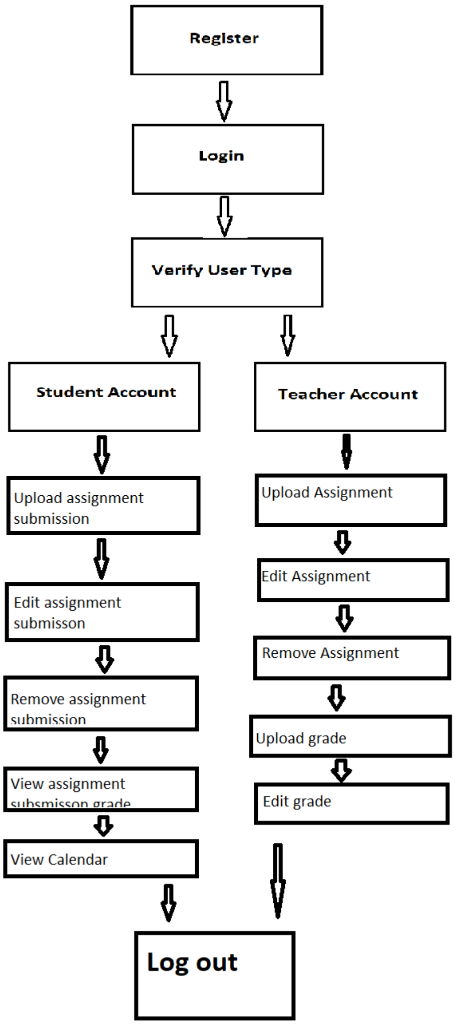


4.2 Frontend

- With the help of HTML, CSS, Javascript, Bootstrap and jQuery. We will be building the UI of our system.
- UI will be user friendly and very minimalistic.

5. High-Level Design

5.1 High Level Design Diagram



5.2 High Level Description

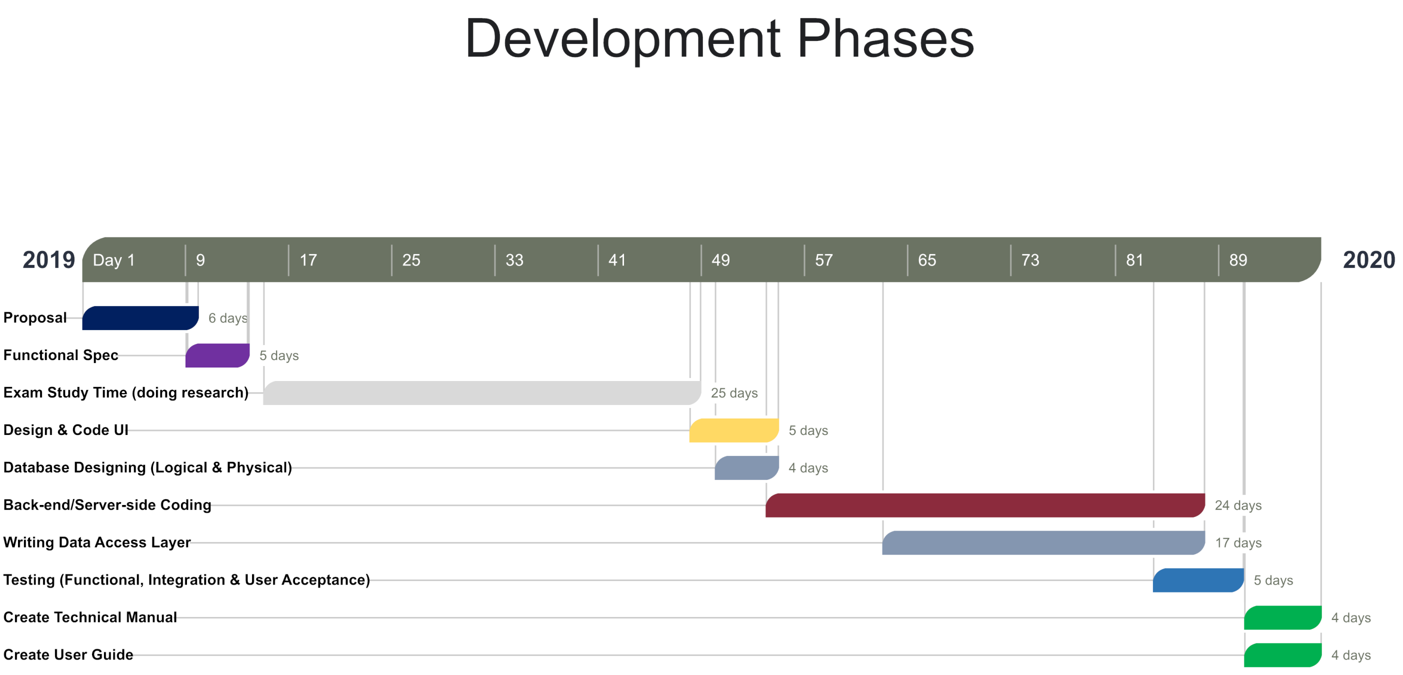
Fig 5.1 is explained below.

- Step 1: Register Register DCU credentials to give you the ability to sign in to the site.
- Step 2: Login Login to the Website using the previously registered DCU credentials.
- Step 3: Verify User type The account type of the user is verified and determined to be one of two types: Student or Teacher. Depending on the type the user is given acces to different portions of the site and different permissions.
- Step 4A: Student Account login Student account type login to the Student account type area.
- Step 4B: Teacher Account login Teacher account type login to the Teacher account type area.
- Step 5A: Upload Assignment submission Student uploads a submission for an assignment.
- Step 5B: Upload Assignment Teacher uploads an assignment to which students can upload submissions.

- Step 6A: Edit Assignment submission Student edits there previous assignment submission.
- Step 6B: Edit Assignment Teacher edits their previously uploaded assignment.
- Step 7A: Remove Assignment submission Student removes their previous submission for an assignment.
- Step 7B: Remove Assignment Teacher removes a previously uploaded assignment.
- Step 8A: View Assignment grade Student views their grade/mark for a graded assignment submisison.
- Step 8B: Upload grade Teacher uploads a grade for a previously submitted assignment submission.
- Step 9A: View Calendar Student views their Calendar featuring upcoming deadlines for their assignments.
- Step 9B: Edit Grade Teacher edits a previously uploaded grade for an assignment submission.
- Step 10: Logout Both account types can log off.

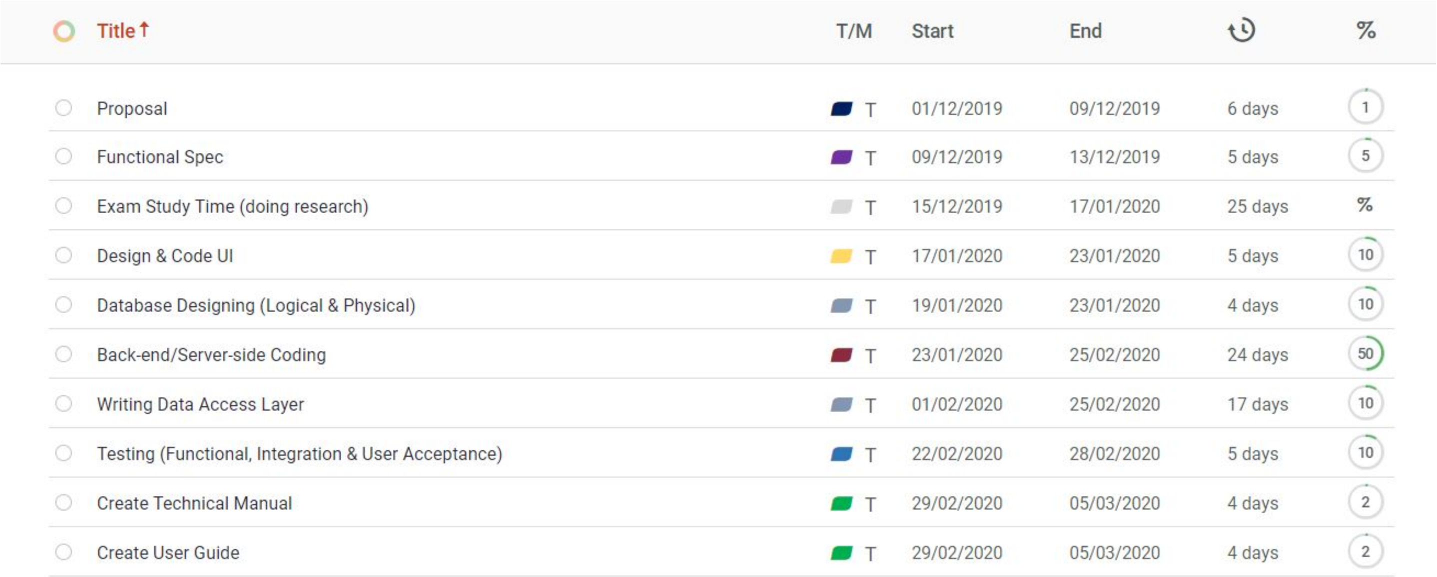
6. Preliminary Schedule

6.1 Gantt Chart



2020-ca326-jdoherty-AMS

6.2 Development TimeLine



○	Title ↑	T/M	Start	End	↺	%
○	Proposal	🟦 T	01/12/2019	09/12/2019	6 days	1
○	Functional Spec	🟪 T	09/12/2019	13/12/2019	5 days	5
○	Exam Study Time (doing research)	⬜ T	15/12/2019	17/01/2020	25 days	%
○	Design & Code UI	🟨 T	17/01/2020	23/01/2020	5 days	10
○	Database Designing (Logical & Physical)	◼ T	19/01/2020	23/01/2020	4 days	10
○	Back-end/Server-side Coding	🟥 T	23/01/2020	25/02/2020	24 days	50
○	Writing Data Access Layer	◼ T	01/02/2020	25/02/2020	17 days	10
○	Testing (Functional, Integration & User Acceptance)	🟦 T	22/02/2020	28/02/2020	5 days	10
○	Create Technical Manual	🟩 T	29/02/2020	05/03/2020	4 days	2
○	Create User Guide	🟩 T	29/02/2020	05/03/2020	4 days	2

7. Appendices

- https://www.w3schools.com/ (to learn HTMl, CSS, Javascript, Bootstrap and jQuery)
- https://www.mysql.com/
- https://dotnet.microsoft.com/download
- gitlab.computing.dcu.ie (version control)