CM2020 Agile Software Projects

Final Report

Project

The design of an application which aids students with their studies as part of the BSc Computer Science degree at the University of London.

Team Member	Student ID
Romi Dhillon	190182674
Ali Gündüz	200358183
Rajin Lalgee	190132738
Chaohui (Sherry) Lu	190181633
Denis Sadovoy	160137123

University of London

Date: September 5th 2022

Table of Contents

1. Introduction

1.1 Aims

The goal of this study is to develop the Student Web Application, which aids students with their studies as part of University of London's Bachelor of Science degree in Computer Science. The aim of this report is to show the iterative steps, decisions, pivot points, and considerations that were taken when developing the student web application. This report builds upon the proposal completed for the midterm, which serves as a solid foundation to design and develop the application.

1.2. Objectives

The team would like to achieve a fully functioning, high quality application by using an iterative approach. To achieve the desired outcome, the following clear set of deliverables will need to be executed for a successful result:

- Assigning the team leader
- Creating the project plan and agreeing on general milestones
- Defining the team composition
- Defining the roles and responsibilities for each team member
- Conducting a literature review and research of methodologies, tools, or techniques to deliver a development project. This includes Project Management techniques, risk management techniques, development tools, iterative design processes and web development frameworks.
- Managing and capturing risks in a risk register (discussing risks as a team)
- Ensuring that all members are attuned to Agile Project Management techniques and sprints
- Agreeing on the tools to be used for development purposes and repositories.
- Conducting iterative design through prototyping (including UI/UX evaluation).
- Conducting iterative design through application development
- All team members attending regular sprint meetings every Sunday to mark the completion of a sprint (including risk reviews and project plan updates).
- Testing all parts of the system through functional testing, including test cases.
- Documenting all tests with effective error handling and justification for testing methods.

1.3. Scope

The scope of this study heavily depends on the activities that were conducted as part of the midterm. It is important to continue the study without duplicating effort or work that was previously completed. The scope of the midterm proposal is reiterated in 1.3.1 and the scope of the final project is detailed in section 1.3.2. It is important to compare the scope of both assignments, as a range of work was previously completed which is being built upon in this study.

1.3.1. Scope of midterm proposal

- Market research
- User requirements including data gathering, personas, user journeys, and user stories
- Functional design including use cases, use case models, and functional requirements
- Wireframes/prototypes including initial concepts, lo-fi, and hi-fi wireframes
- Technical design
- Ethical Audit
- Project Management

1.3.2. Scope of final project (this study)

- Planning and research activities including a literature review and market research of project management, risk management, development tools, and web development frameworks. Defining roles and responsibilities and creating a project plan and risk register.
- Prototyping and user-centered design, focusing on progressing the hi-fi wireframes from the midterm, whilst taking user experience and user interface design modifications into account.
- Updates to the midterm functional requirements and use cases.
- Conducting a change impact assessment and progressing the ethical audit from the midterm.
- Application design, architecture, and development. This includes process design, development operations (including agile methodologies), and version control.
- Application functional testing, including usability testing and user interface testing.

2. Planning and research

A range of planning activities were required before formally starting the development activities within the project. These planning activities are detailed in this section.

2.1. Literature review and research

There is a range of literature which evidences the benefits of using specific methods, tools, or techniques to support a development project. This section critically examines the existing literature on the topics stated below, which will aid in making decisions on the best supporting methods to develop the application and deliver the project.

- Project Management
- Risk Management
- Development Tools
- Web Development Frameworks

2.1.1. Project management review

Even within multiple sectors, there is a constant debate about the best project management methodology to deliver projects. According to the Association for Project Management (APM), the two most popular project management methodologies in multiple industries are 'Agile Project Management and Waterfall Project Management'[1]. For both methodologies, the available literature generally agrees on many points when assessing the strengths and weaknesses of each methodology. For example, in accordance with Forbes, the waterfall approach 'works best when there are defined requirements, defined milestones and dependencies, and when there is less risk of scope creep.' [2] Other well-established institutions such as the Project Management Institute (PMI) are quick to point out that the agile approach works best when the 'detailed requirements are unknown or subject to change, and when the team is co-located and requires collaborative teamwork.' [3] The International Project Management Association (IPMA) highlights some of the disadvantages of using the agile approach, stating that it 'demands management and prioritisation of the backlog, constant collaboration with stakeholders, and that sprints can be taxing on team members that are not accustomed to working in an iterative or adaptive approach.' [4]

The team wishes to work in a flexible manner, which focuses on iterative developments where the details can change based on pivotal decisions along the journey. There are six team members that are in different areas of the world, and each team member is heavily dependent on collaborative teamwork. Many of the team members are already accustomed to flexible and adaptive collaborative working within their professions that they work in. Even though the literature highlights that backlog management and sprints can be taxing on team members, there are six team members to split the work amongst. This allows for the multitude of activities to be shared and assigned without overloading team members. Based on the research and requirements of the team, the agile project management approach is best suited for developing the application and delivering the project. This will incorporate classical agile management techniques such as sprint planning, sprint execution, sprint meetings, backlog management, and Kanban boards. Even though the agile approach will be used for most of the project, a project plan with high level milestones will be

created to ensure that key milestones set out in the report objectives have been achieved. It is important to note that the team will not be bound to timelines stated in the project plan.

2.1.2. Risk management review

Risk management may help to minimise the impact of unforeseen events. Based on journals accredited to the Institute of Risk Management, managing risks adequately with a risk register (quantifying risk severity) can 'reduce costs, increase performance and reduce delays associated with product delivery.' [5] Even so, the Organisation of Certified Risk Managers (OCRM) state that 'theoretical risks are not always of a practical nature. It is difficult to eliminate ambiguity when scoring risk probability, impact, and severity; and risks are based on the perception of different team members.' [6] According to the OCRM, 'capturing risk is only as good as the minds that think of the risks. Major risks can be excluded if none of the team members think of higher priority risks.' [6]

Although it is true that all risks in a project cannot be accounted for, the team believes that capturing risks to the best of the team's ability is much better than not capturing risks at all. There is always going to be subjectivity of risk scores based on the opinions of team members, however, collective scoring in team meetings will help to reduce this affect. The team has decided to create a risk register to capture risks, score their severity, and develop mitigation strategies to minimise the impact or the probability of individual risks. The risk register will be reviewed and updated at each weekly sprint meeting.

2.1.3. Development tools review

In order to develop the application, it is necessary to deploy tools which aid in completing the deliverables and development of the application in an efficient manner. The tools also need to complement the chosen agile project management approach. Based on studies conducted by a major digital consulting firm named Cognizant, 'development operations (DevOps) are an integral part of any project which combines coding with collaboration.' [7] The article goes on to say that 'cloud tools such as Microsoft Azure DevOps promote the application of agile methodologies whilst allowing integration of code from platforms such as GitHub.' [7] There are many tools and technologies that can be utilised for development operations, however, when conducting research, there are multiple organisation that highlight Microsoft Azure DevOps as efficient and contemporary tools to fulfil all needs of a development project. For example, Ernst and Young states that 'Microsoft Azure allows for the execution of Kanban boards, monitoring backlog activities, GitHub integration, Sprint boards, Sprint timelines, Pipelines, and Test Plans.' [8]

As part of the midterm, Slack was used for collaboration, and the justifications for usage were provided in the midterm report. Slack will continue to be used for collaboration and team meetings. Notion was previously used in the midterm for storing documents, however, a more sophisticated repository is required so that application code and documents can be developed in an incremental and agile manner. Based on contemporary literature, there are multiple cloud-based repositories that can be used to store, track, manage, and control changes to files. Stackify, a trustable online source states fifty different source code repository hosts, and mentions 'Bitbucket, GitLab, GitHub, Beanstalk, Codebase, Assembla, Gitkraken' [9] amongst many others. Market share

statistics for 2021 show that GitHub had a ‘market share of 81.77% in the global source-code-management market. Its closest competitors are GitKraken with 8.14%, SourceTree with 7.96% and Visual Studio Team Services with 1.07%.’ [10] Such market dominance may arise from the vast features stated on the GitHub site, which highlights many features for collaborative coding, automation, security, client apps, project management, and team administration.

Microsoft Azure DevOps is a platform which will allow the team to implement the agile techniques in an efficient manner and will allow integration with GitHub. It will be chosen as the primary platform to execute activities associated with development operations, providing the benefits highlighted in the researched literature. GitHub will also be used to store, track, manage and control the code and project artefacts/documents. This decision was made due to the sheer market dominance and advanced features available within the platform. A private repository will be created where all team members can work collaboratively to achieve project objectives.

2.1.4. Web Development Frameworks Review

It was decided in the midterm that Node.js, Express.js, and MySQL would be used to develop the web application. When starting work for the final assessment, it was identified that the features of the proposed system are very ambitious, and a suitable web development framework needed to be applied that prioritises speed and quick implementation. According to an article authored by Prashant S Lokhande in the International Journal of Advanced Computer Science Research, applications such as Flask that are written using Python can have advantages over other web development frameworks. The article states that ‘smaller projects are more suited to Flask’s lightweight extensible microframework, as it does not rely on other tools or programming libraries to function. This means that small applications can be created incredibly quickly and are highly scalable.’ [11]. A report written by Matthew S. Bonney at the University of Sheffield highlights that Flask is ‘highly flexible as it can be easily rearranged and moved around. This allows developers to change the structure of the application without causing major errors, leading to more efficient and iterative development which can respond to changes in features. By supporting modular programming, all parts of the structure are flexible, movable and testable.’ [12]

Express.js is mainly considered to be a backend microframework for Node.js, and Node.js is built on the JavaScript V8 engine. A front-end JavaScript library such as React can be used to develop front end features. A backend database such as MySQL, SQLite, or PostgreSQL is required to store data and integrate with the application. A report written by Nimesh Chhetri at the St.Cloud State University highlights some of the excellent features when using Node.js and Express.js, such as ‘superior security (authentication and login) and suitability for complex applications where multiple middleware modules are required to perform request and response executions.’ [13]

When researching, it was clear that there are multiple other options such as Django, Angular, Ruby on Rails, etc. However, a suitable application needs to be chosen based on requirements of development. The team is aware that the application does not contain real-time data and can be considered a small to medium sized application which requires multiple changes in a short period of time. Based on the requirements, it was decided that the Flask microframework would be used, as it provides the right level of flexibility and extensibility to develop both front-end and back-end solutions efficiently.

2.2.Roles and Responsibilities

The literature review and research resulted in justifications for choosing methods for project management, risk management, development tools, and web development frameworks. The team leader needed to be assigned, as it was time to start project activities. Roles and responsibilities were agreed in a slack meeting on 10th July and were assigned to team members based on their strengths, capabilities, and availability. Agreed roles and responsibilities for each team member are shown in Figure 2.1. Percentage contributions are discussed at the end of the report (section 8).

Name	Role(s)	Responsibilities
Romi Dhillon	- Team Leader/ Technical Lead - Application Developer - Market researcher - Application Tester	- Full application development using Flask - Usability Testing - User Interface Testing - Debugging and error handling - Literature Review and Research - Writing all sections of the report - Project Plan - Risk Register - UX and UI research - Functional requirements and change impact assessment - Leading sprint team meetings (every Sunday)
Rajin Lalgee	- Front-end application support - Research Engineer	- Development of front-end for landing page - Front-end modifications to the modules Page - Data Gathering
Ali Gündüz	- Research Engineer - Ethics/IP researcher	- Ethical Audit and documenting ethical/IP research - Assets research - Data gathering
Chaohui (Sherry) Lu	- UX Designer - Research Engineer	- Hi-Fi Wireframes and UX/UI Design - Asset collection and asset research - Package/library research
Denis Sadovoy	- Process Engineer	- Process Flow Diagram - Functional (usability) testing
Eesha Tirazia	- n/a (absent during assignment of roles)	- n/a

Figure 2.1 – Defined roles and responsibilities for each team member

A high-level project plan was created, highlighting key milestones that needed to be achieved as part of the project. After reviewing the literature, it was decided that the waterfall approach would not be followed, so the project plan only serves the purpose of highlighting key high-level activities, even though the timelines are flexible for each activity. The project plan is shown in Figure 2.2. The high-level activities will be achieved by executing many small tasks across multiple sprints. The project plan will only be reviewed to assess whether the high-level activities have been achieved.

Figure 2.2 – Project plan

2.3.Risk Management

The literature review highlighted the need to use a risk register to mitigate unforeseen events. Although a risk register is an evolving document through the life of a project, the very first version of the risk register has been shown in Figure 2.3. After conducting the first risk review meeting as a team, a total of eleven risks were noted within the risk register. Each risk was given a description, review date, category (based on the key at the top of the figure), impact score, likelihood score, and resulting severity score. A mitigation strategy was devised for each risk to reduce the impact or likelihood of the event occurring. Each risk was given a status and a target closure date. Each risk was regularly assessed at the end of each sprint meeting conducted on Sundays.

	Category
Risk Categories:	
Strategic	S
Legal and Regulatory	L
Political/Economic	P
Environmental	E
Technical/Operational/Infrastructure	T

No	Date last reviewed	Risk description	Category	Risk Owner (Accountable)	Inherent Risk			Mitigation Actions	Status	Comments on Status	Target Closure
					Impact	Likelihood	Exposure				
1	22-Jul-22	There is a risk that all of the points within the marking rubric will not be fulfilled as part of the report	S	Romi Dhillon	High	Low	Med	Ensure that each activity within the marking rubric is focused on when planning activities on a weekly basis as part of sprints.	New	The requirements document uploaded to github expands on the points within the marking rubric. Tasks are being assigned to specifically cover the tasks in the rubruc.	20-Aug-22
2	22-Jul-22	There is a risk that the report will not have an adequate flow from start to finish, especially when adding multiple different artefacts.	T	Romi Dhillon	High	Low	Med	Get started with the report early and ensure that it takes the user through a journey	New	The report was completed by prioritising the journey and flow	20-Aug-22
3	22-Jul-22	There is a risk that different team members have different expectations regarding some of the pages within the application	S	Romi Dhillon	Low	Med	Low	Conduct additional team meetings so that all team members have aligned expectations associated with the content of each pages.	New	A meeting will be conducted during Sprint 3 or Sprint 4.	05-Aug-22
4	22-Jul-22	There is a risk that there will not be enough time to conduct all of the different types of testing required within the project.	S	Romi Dhillon	Med	Low	Low	Complete the first iteration of application development by 1st August. This will leave ample time for testing.	New	First iteration on target for completion by the end of sprint 3 (31st July)	01-Aug-22
5	10-Jul-22	There is a risk that different team members will not be able to attend meetings due to time differences (due to location)	S	Romi Dhillon	Med	Med	Med	Meetings are to be conducted at a time/day once a week which suits everyone. This decision will be made on slack on a weekly basis.	Closed	Team meetings are being held every Sunday and nearly all members have been turning up.	22-Jul-22
6	01-Jul-22	There is a risk that the same collaborative tools used in the midterm are not suited for software development	T	Romi Dhillon	Med	Med	Med	Ensure that all team members collaborate using Git and Azure DevOps.	Closed	Collaboration is taking place on GitHub and Azure DevOps regularly.	10-Jul-22
7	10-Jul-22	There is a risk that some aspects of functionality within the application are too ambitious. There are time constraints within the project, and there may not be enough time to implement honourous functionality.	T	Romi Dhillon	Med	Med	Med	Develop the application In an intensive manner, ensuring there is enough time to add complex functionality.	New	First iteration on target for completion by the end of sprint 3 (31st July)	10-Aug-22
8	10-Jul-22	There is a risk that team members have other modules and other commitments outside of work, leaving less time to dedicate to the project.	S	Romi Dhillon	Low	High	Med	Sprint task tracking will be conducted on DevOps and regular updates will be achieved in weekly team meetings.	New	Sprints and weekly team meetings are conducted to get regular updates.	01-Sep-22
9	10-Jul-22	There is a risk of COVID on a global scale causing health and economic impacts.	P & E	Romi Dhillon	Med	Med	Med	If team members are affected, then other team members will support the team member and pick up the additional workload.	New	Some team members have been affected by COVID, and other team members have pitched in to suport.	01-Sep-22
10	10-Jul-22	There is a risk of intellectual property and ethical breaches taking place when created the application.	L	Romi Dhillon	High	Low	Med	Ensure that ethical aspects are properly researched and do not use assets which may breach intellectual property rights.	New	To be conducted.	10-Aug-22
11	10-Jul-22	There is a risk that the application will be enhanced without updating the prototypes first.	T	Romi Dhillon	Low	Med	Low	Ensure that the UX Designer engages with the application developer.	New	UX Designer to engage with application designer in Sprint 3 and 4.	15-Aug-22

Figure 2.3 – Risk register

3. Prototyping and user-centered design changes

The task of prototyping and software design is not straight forward. In order to start developing software, it is necessary to start with low-fidelity prototypes and then progress to high-fidelity prototypes. In theory, this should be the end of the process for prototyping, allowing developers to develop the application. In reality, the Application Developer and the UX Designer had to ensure that user-centred design was taken into consideration and that the features could be implemented from a practical standpoint. User-centred design includes both user experience (UX) and user interface (UI) design. When developing the application, new design ideas were generated based on user-centred design approaches, and better design methods were identified as part of the process of design. This led to deeper discussions in meetings during the week, which were conducted separately to the Sunday sprint meetings. The modifications made to each page of the high-fidelity prototype along with the reasons for change have been described in the subsequent sub-sections. Each modification was made in an iterative manner before and during application development. There was a continual flow of ideas and healthy disagreements, which produced favourable results. It was a balancing act between team desires and realistic goals within the given timeframe. Any changes were also communicated to the team in sprint meetings, as such changes had the ability to impact other documentation (such as the application process flow diagram). The ability to make new decisions and change the course of direction on multiple occasions was facilitated by the agile project management approach.

3.1.Modifications to the landing page

The high-fidelity wireframe for the landing page that was completed in the midterm can be seen in Figure 3.1. Users that search for the website online are directed to this page. The purpose of the landing page is to provide information about the features, our team, and the links to sign-up/sign-in. Through additional discussions, the team felt that aspects of user experience needed to be improved by making the user feel as though they can reach out to the team and know more about the team. ‘‘User experience (UX) focuses on having a deep understanding of users, what they need, what they value, their abilities, and their limitations. For high quality user experience, information must be useful, usable, desirable, findable, accessible, credible.’’ [14] In order to facilitate some of these aspects, an additional section was added named ‘about us’ which provided useful information about the team members (as shown in Figure 3.2). The footer was also modified to include links to social media pages and contact information, leading to greater accessibility and findability. Such links allow users to reach out to the team on social media platforms and to stay updated with the latest news. A copyright was also added at the bottom of the page to ensure that the material displayed in the application cannot be utilised without permission, which promotes credibility.

‘‘User Interface (UI) design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions’’ [15] We improved the user interface design by changing terminology such as ‘login’ to ‘sign-in’ as we believed it was easier to understand. We also removed the ‘add review button’ within the first section of the landing page as we believed that it would confuse users when visiting the page.

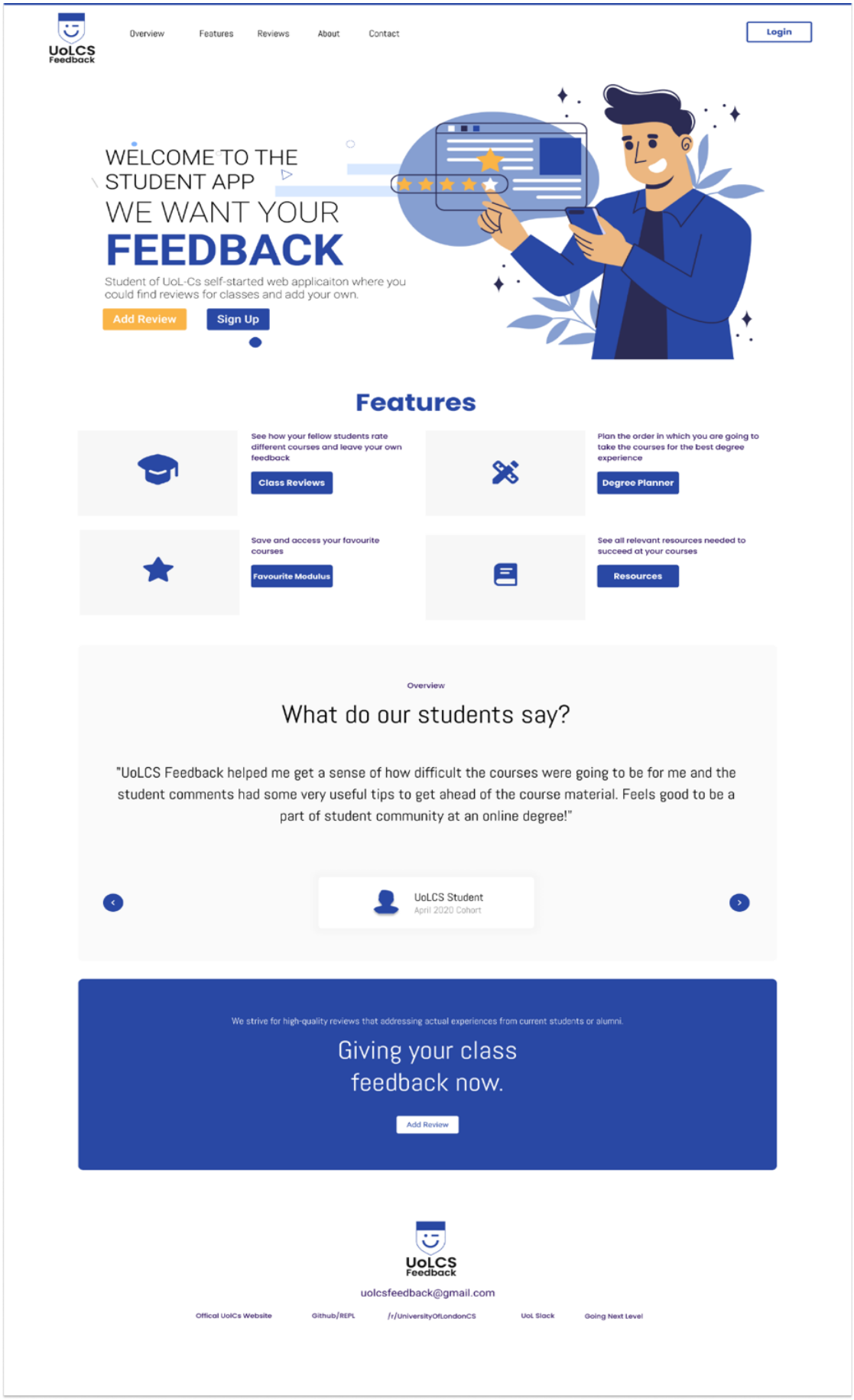


Figure 3.1 - Original HiFi wireframe for landing page from midterm

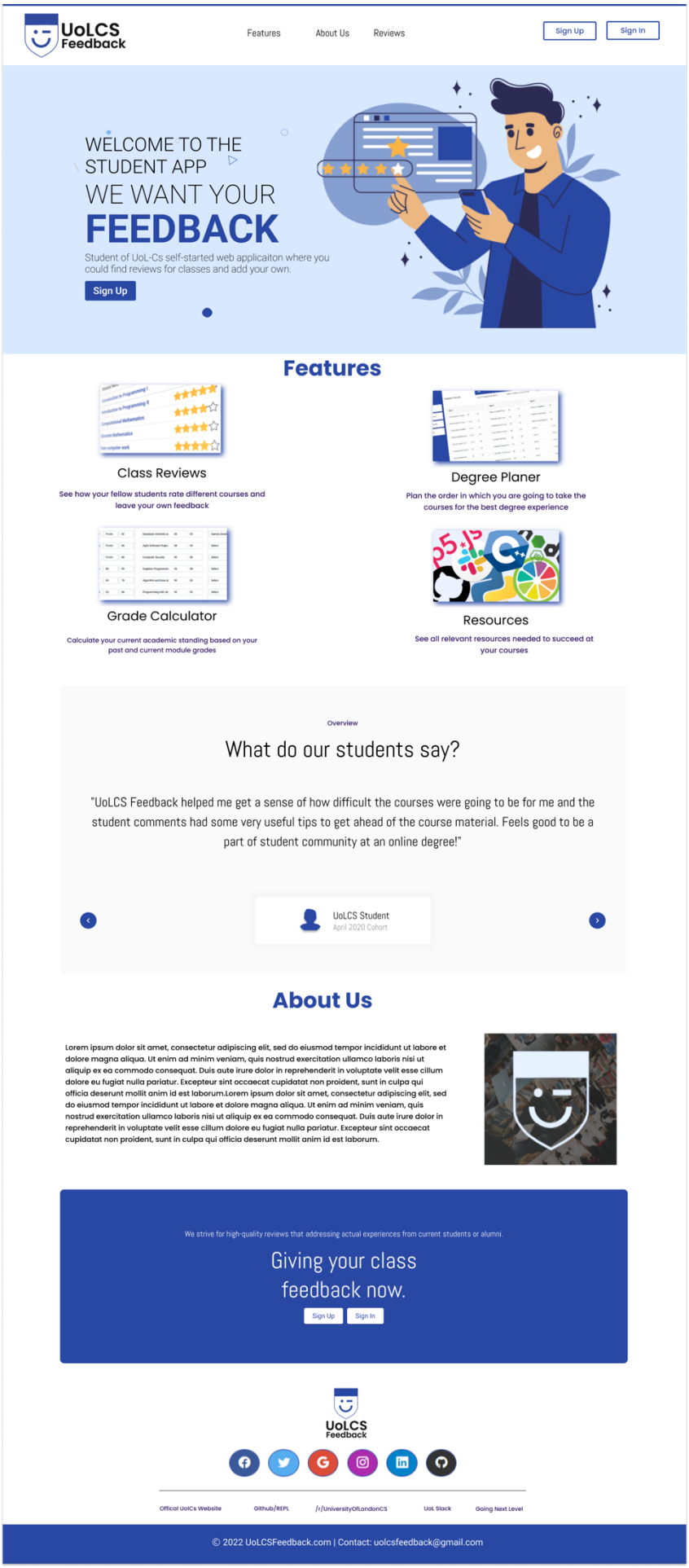


Figure 3.2 – Updated hi-fi wireframe for landing page

3.2.Modifications to the sign-up page

When visiting the landing page, the user can sign-in to the application after being redirected to the sign-in page. The original design for the sign in page was very simplistic as seen in Figure 3.3. From a user experience perspective, aspects of desirability, accessibility and credibility were improved by implementing greater security measures (Figure 3.4). The user needs to ensure that the confirmed password matches the original password, and that the password conforms to specific character rules as shown in Figure 3.4. This ensures greater security when signing up as a user of the application. The user interface was also improved by adding a home button directing the user to the landing page, and also by adding a footer. This enhances accessibility to key pages within the application. The University of London student application logo was also placed at the top right of the page to ensure that there is a level of consistency and standardisation in the design, enhancing findability.

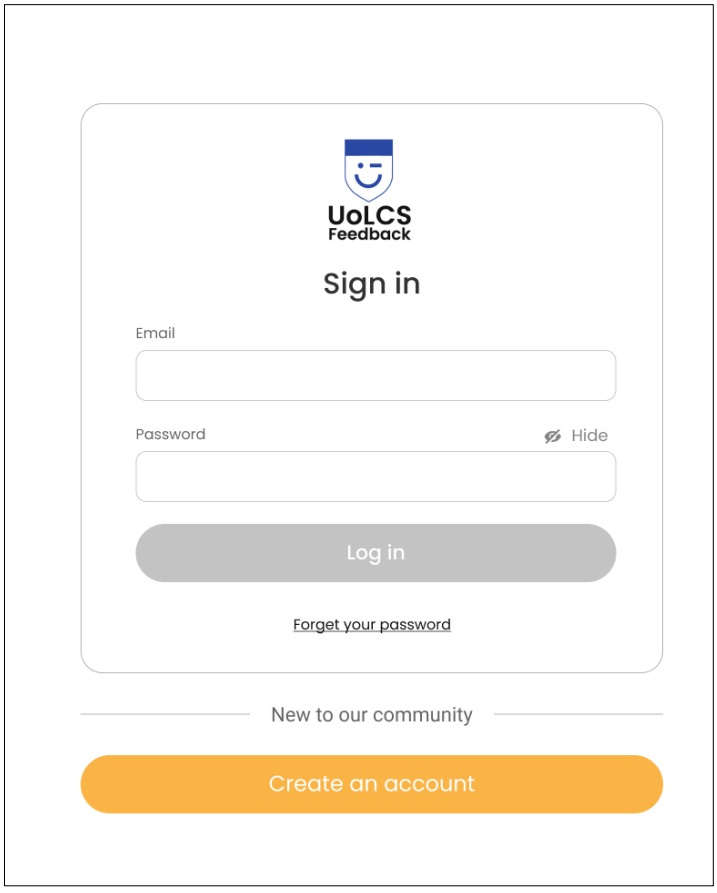


Figure 3.3 – Original hi-fi wireframe for the sign-up page

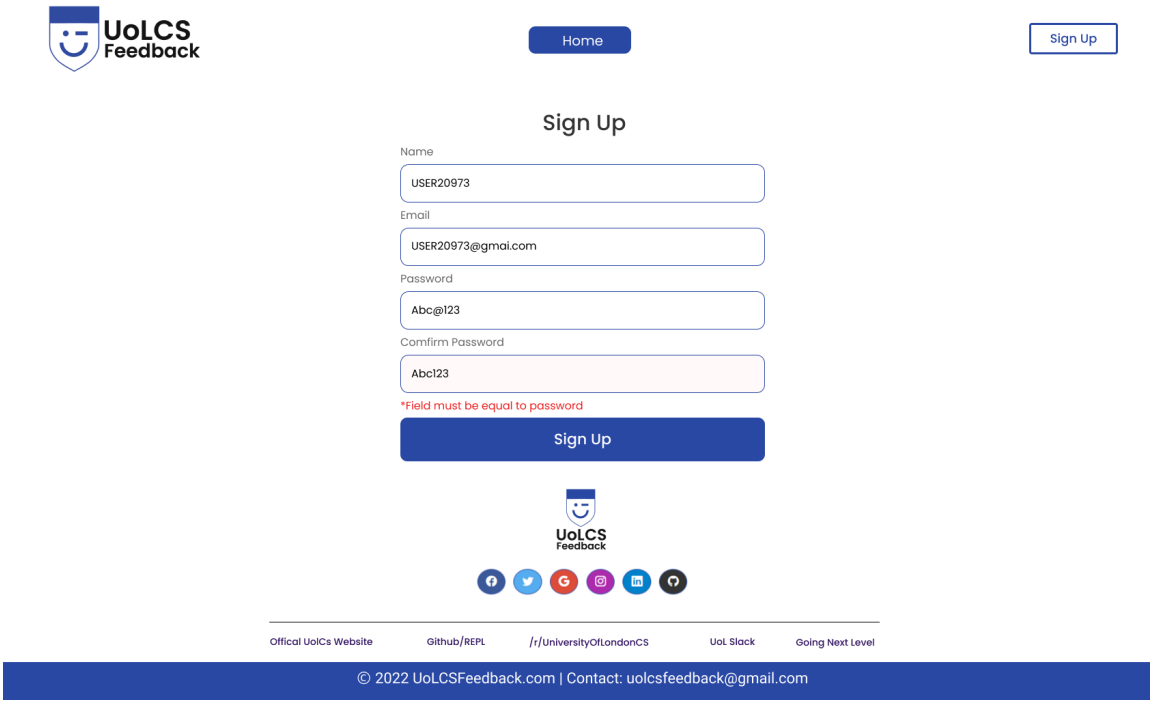


Figure 3.4 – Updated hi-fi wireframe for sign-up page

3.3.Modifications to the sign-in page

Similar aesthetic updates were made to the sign-in page as those made for the sign-up page. When discussing the process flow of activities with the team, it was decided that the user needs the ability to reset their password, as it is only natural for a certain percentage of users to forget their password. Figure 3.6 shows the addition of the forgot password link to the design. By adding forgot password functionality, the level of accessibility and usability is increased.

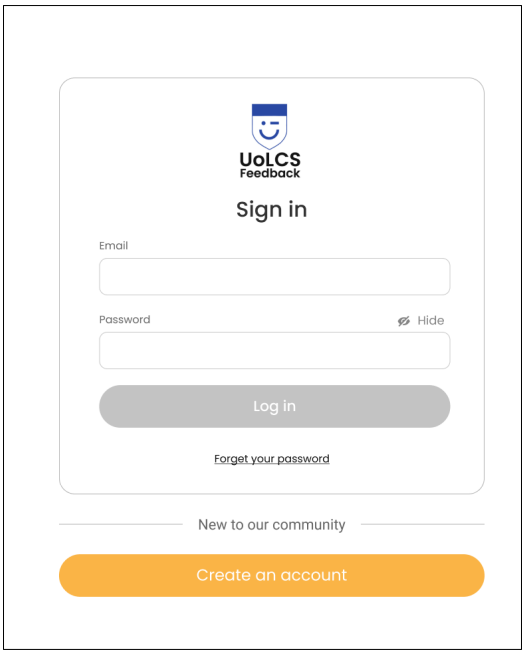


Figure 3.5 – Original hi-fi wireframe for the sign-in page

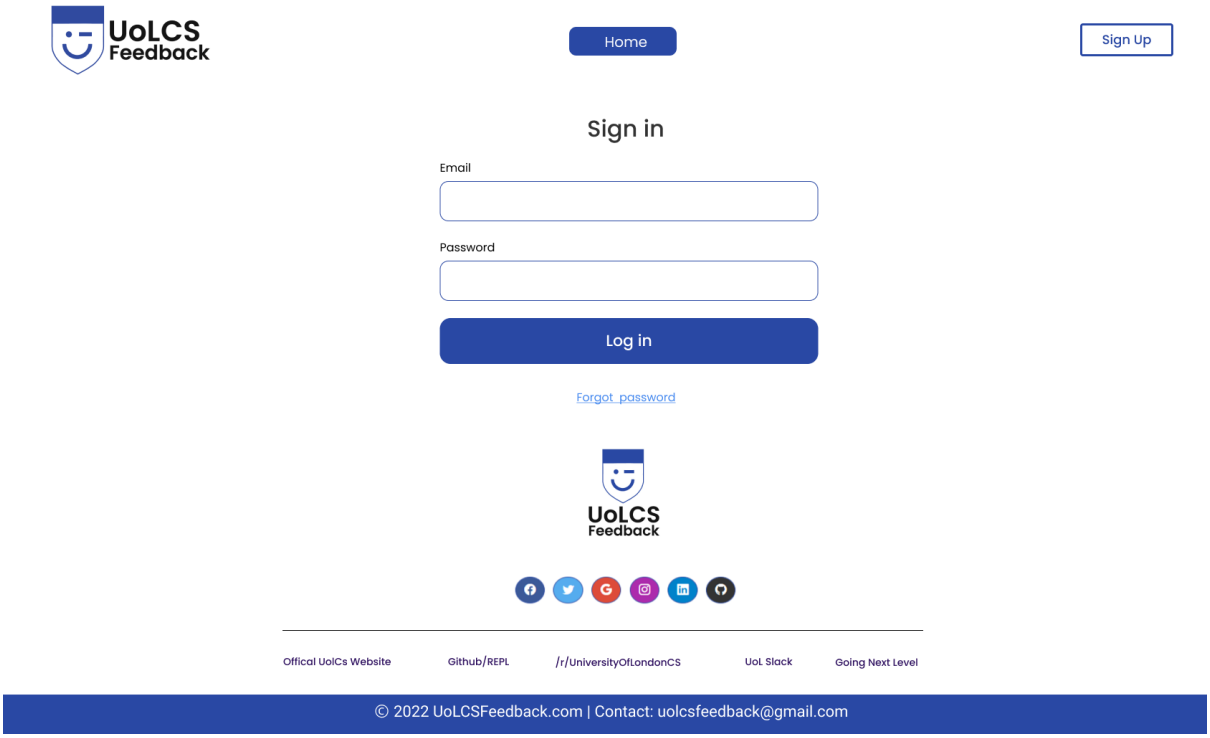


Figure 3.6 – Updated hi-fi wireframe for the sign-in page

3.4.Modifications to the dashboard

The dashboard is encountered once the user signs into the application.

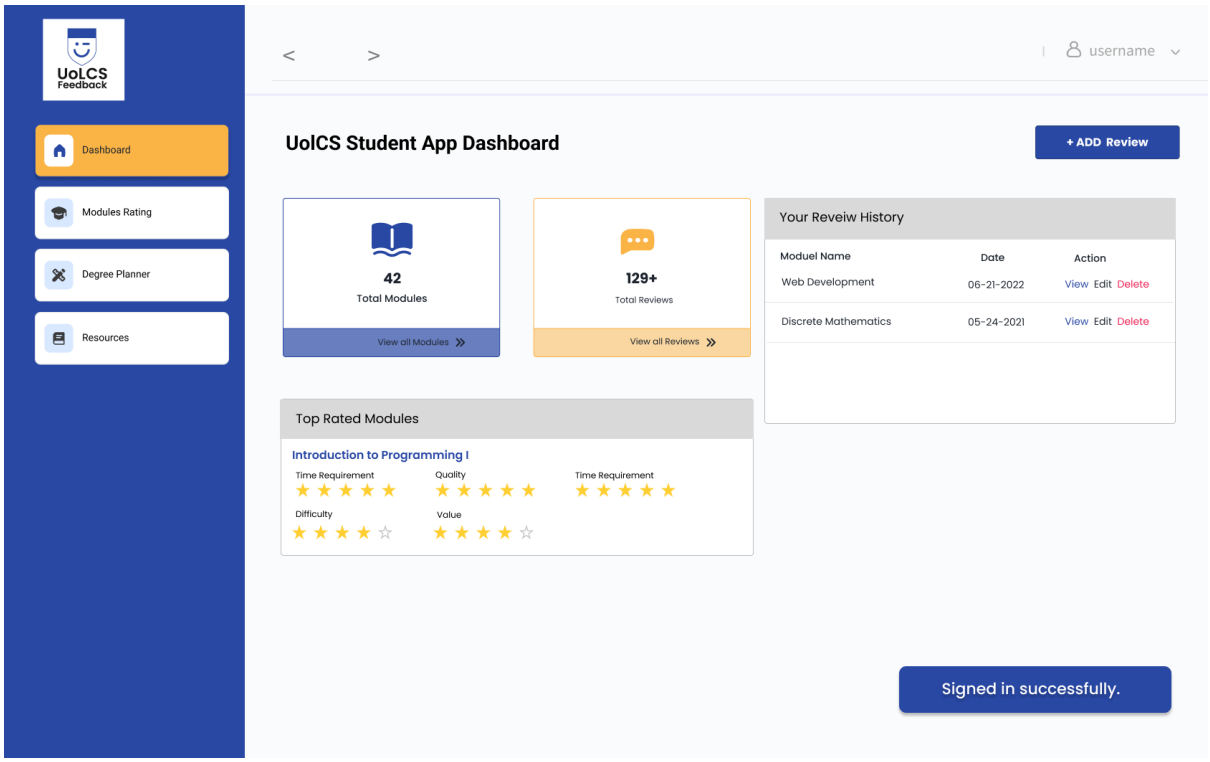


Figure 3.7– Updated hi-fi wireframe for the dashboard page

The page was mostly kept the same, except that a notification was added to the bottom right of the screen which states that the user has signed in successfully (as shown in Figure 3.7). A similar notification stating 'successfully signed out' has been added when the user signs out of the application. Such an addition adds clarity for the user and enhances credibility within the application. Certain users like to be cautious for security reasons and want to know the status of whether they are signed in or signed out.

3.5.Modifications to the module ratings page

The original design for the module ratings page (Figure 3.8) allows users to leave module ratings by adding a review, assigning a rating and leaving comments. The application had many features to develop within a short space of time. Due to time constrains, it was decided that the filter functionality and the ability for other users to upvote specific reviews would be removed from the application. Although there are only 24 modules within the degree, the filter functionality would have been a good addition, as it allows users to filter modules by year, topic area, and ratings. By allowing users to upvote reviews, it would allow the community to prioritise the highest rated reviews. The application will still function to a high standard without the filter and upvote functionality, but these features will be implemented in a future release. By removing the filter functionality and upvote ability (as shown in Figure 3.10), we are removing aspects of positive user experience. Even so, there is a balancing act between implementing functional requirements that are considered a must, and those which are considered desirable.

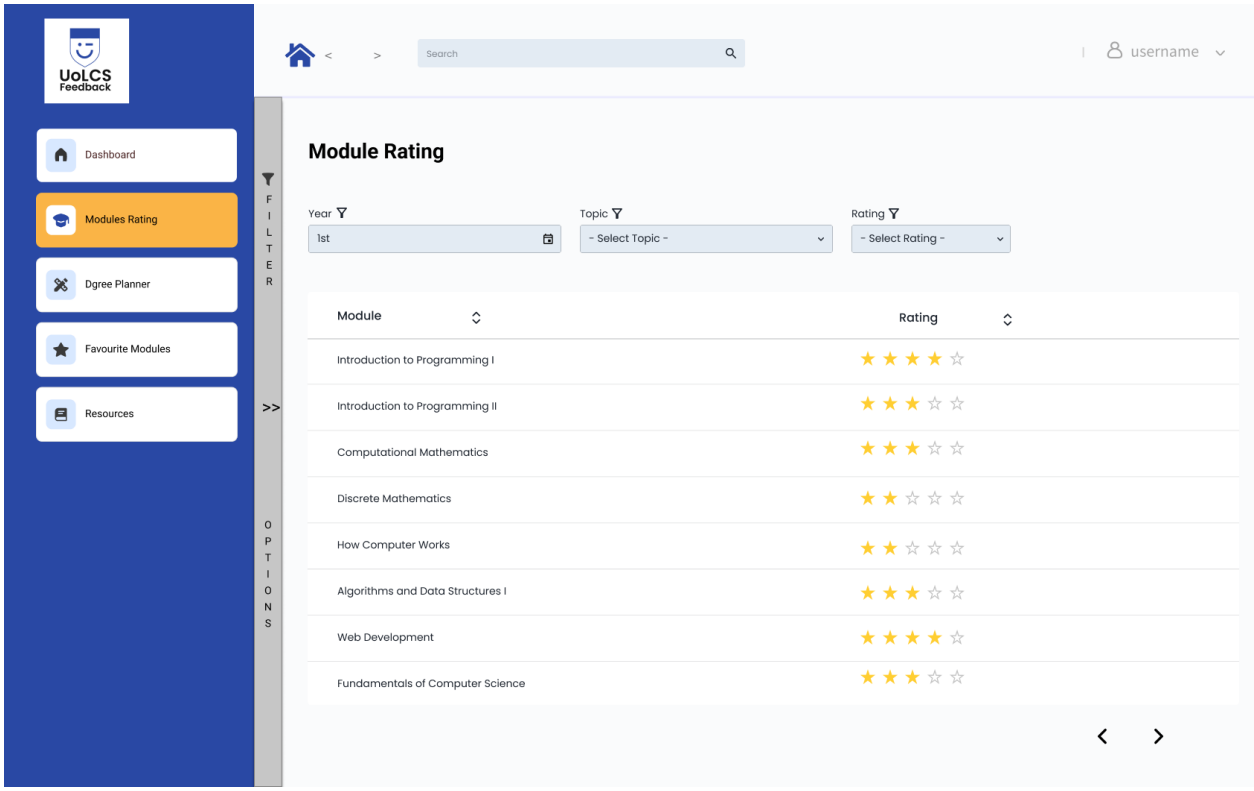


Figure 3.8 –Original hi-fi module rating page 1

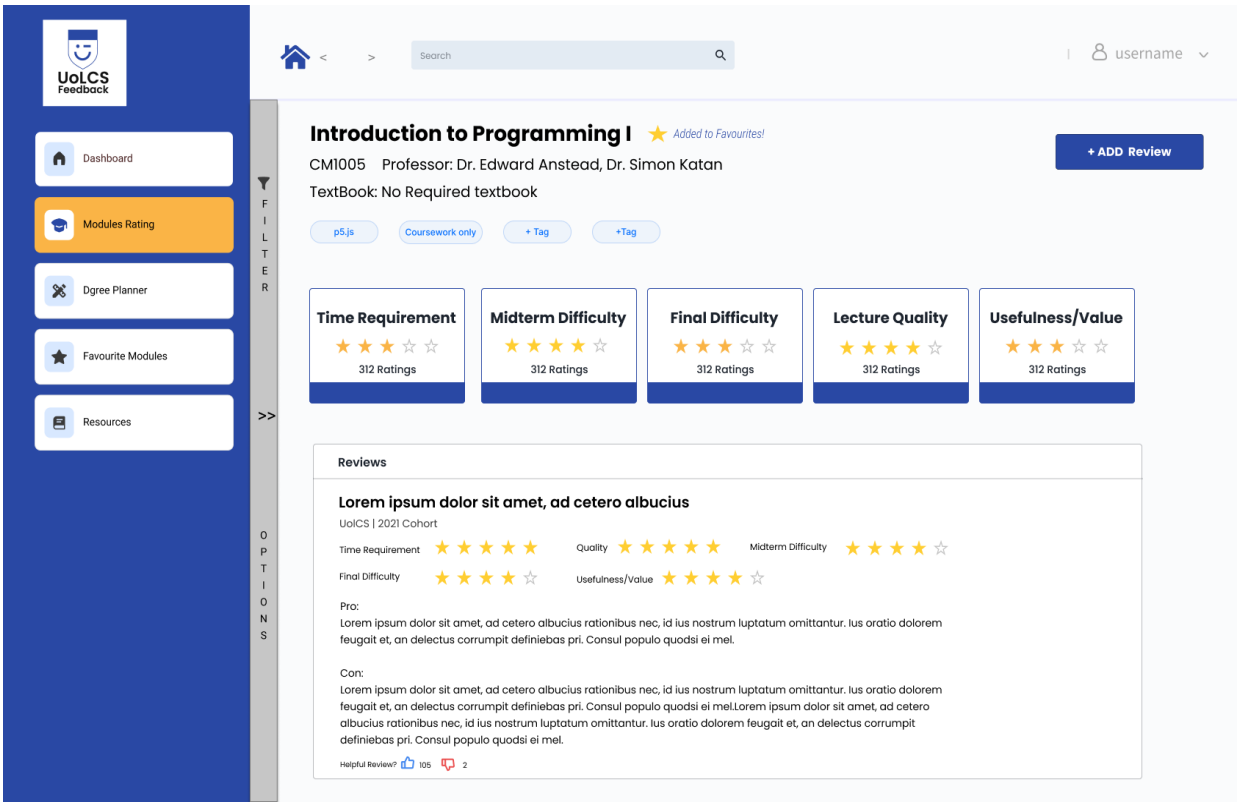


Figure 3.9 – Original Hi-Fi Module Rating page 2

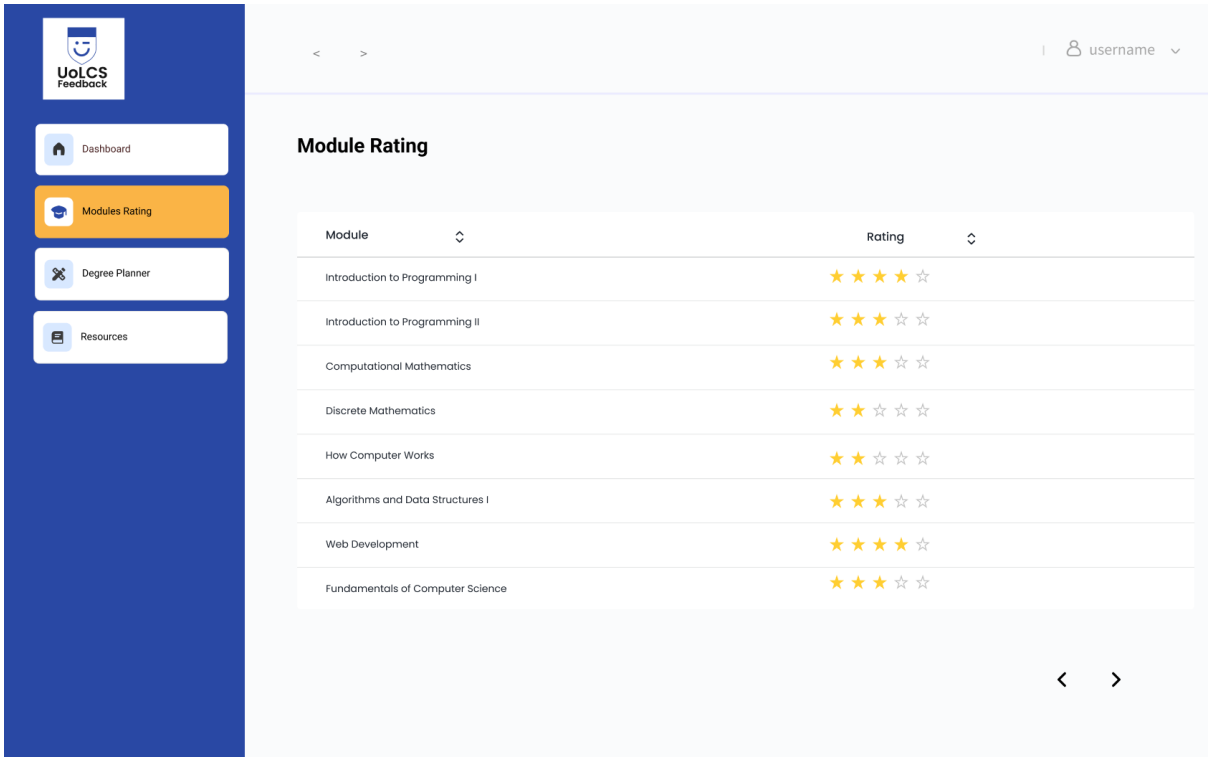


Figure 3.10 – Updated hi-fi module ratings page 1

3.6.Deletion of the favourites page

After logging in to the application, it was previously decided that a ‘favourite modules’ tab would exist within the navigation bar which would transport the user to a favourite modules page within the application (wireframe shown in Figure 3.11). The original intention was to allow students to store their favourite modules in the ‘favourite modules’ page and to keep notes and documents associated with their favourite modules. After much deliberation, it was decided that this functionality did not add great value to the user’s experience. The ability to add notes is already available within the resources page, and the user already rates their favourite modules within the modules page. The functionality of the favourite modules page is already inherent in other areas of the application, which means that the page is not useful or desirable. Due to this revelation, it was decided to omit this page from the application. You may have noticed that the updated wireframe in the module ratings page (Figure 3.10) does not have a favourite modules tab in the navigation bar as it was removed from the updated design. This is consistent across all the updated wireframes.

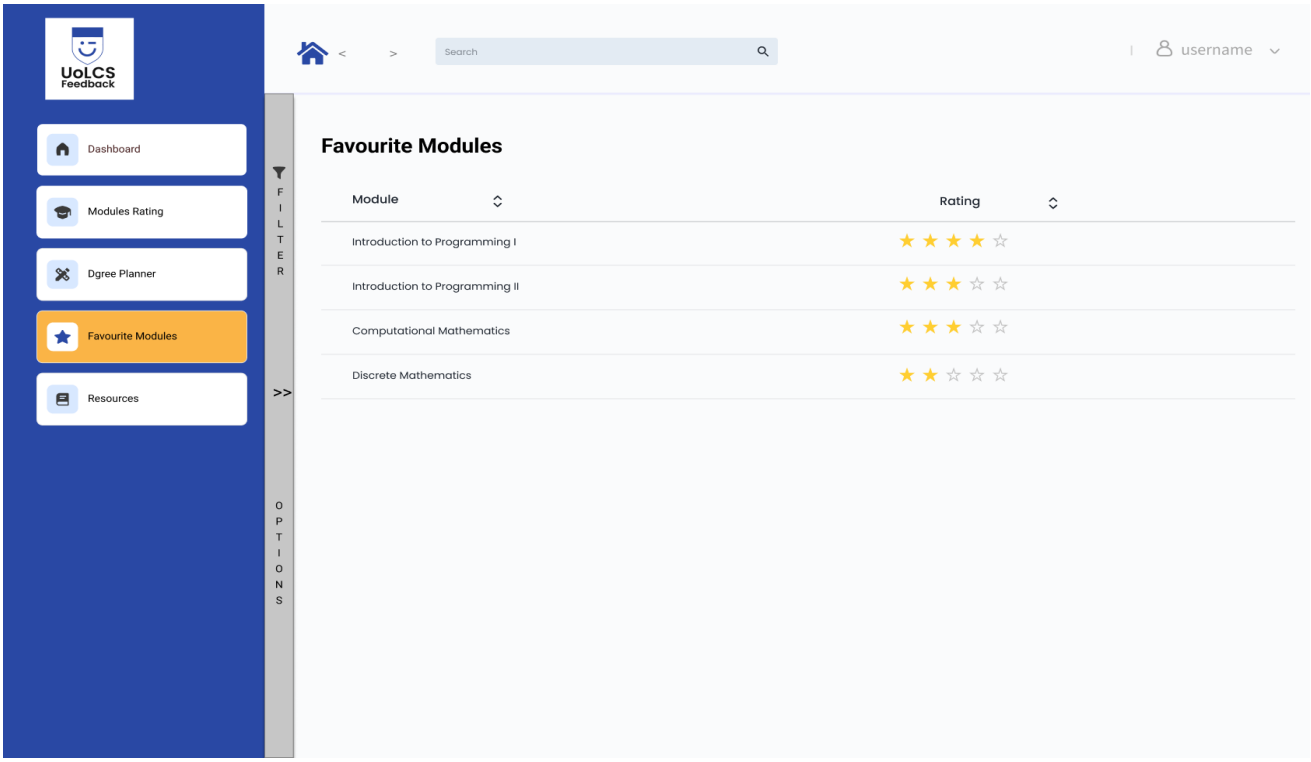


Figure 3.11 – Original hi-fi wireframe for favourite modules page

3.7.Removal of the search functionality

It was decided that the search functionality would be omitted due to time constrains and task prioritisation. Search functionality is a beneficial feature as it promotes the user experience aspect of findability. In a future release/update, users will be able to find specific information by making text searches, which may be helpful to find specific notes, resources, etc.

3.8.Modifications to the resources page

It was originally agreed that useful resources for each module would be added to the resources page. When attempting to conduct this task, it was evident that there are hundreds of resources to provide links for, which is not an efficient way of imbedding the information into the application. In addition, we discovered that there are certain intellectual property right violations by hosting university resources directly in our servers. Information regarding resources already exists in the Resources Enriching Perennial Learners (REPL) website, and this website hosts university resources without infringing intellectual property rights. For this reason, we decided to provide a link to the REPL website for each module (updated wireframe shown in Figure 3.13). A link to the relevant Slack page for each module was also provided for each module, which promotes social media accessibility from the application. The ability to download documents was also removed (area in the bottom right of Figure 3.12) as we are no longer storing university documents within the application.

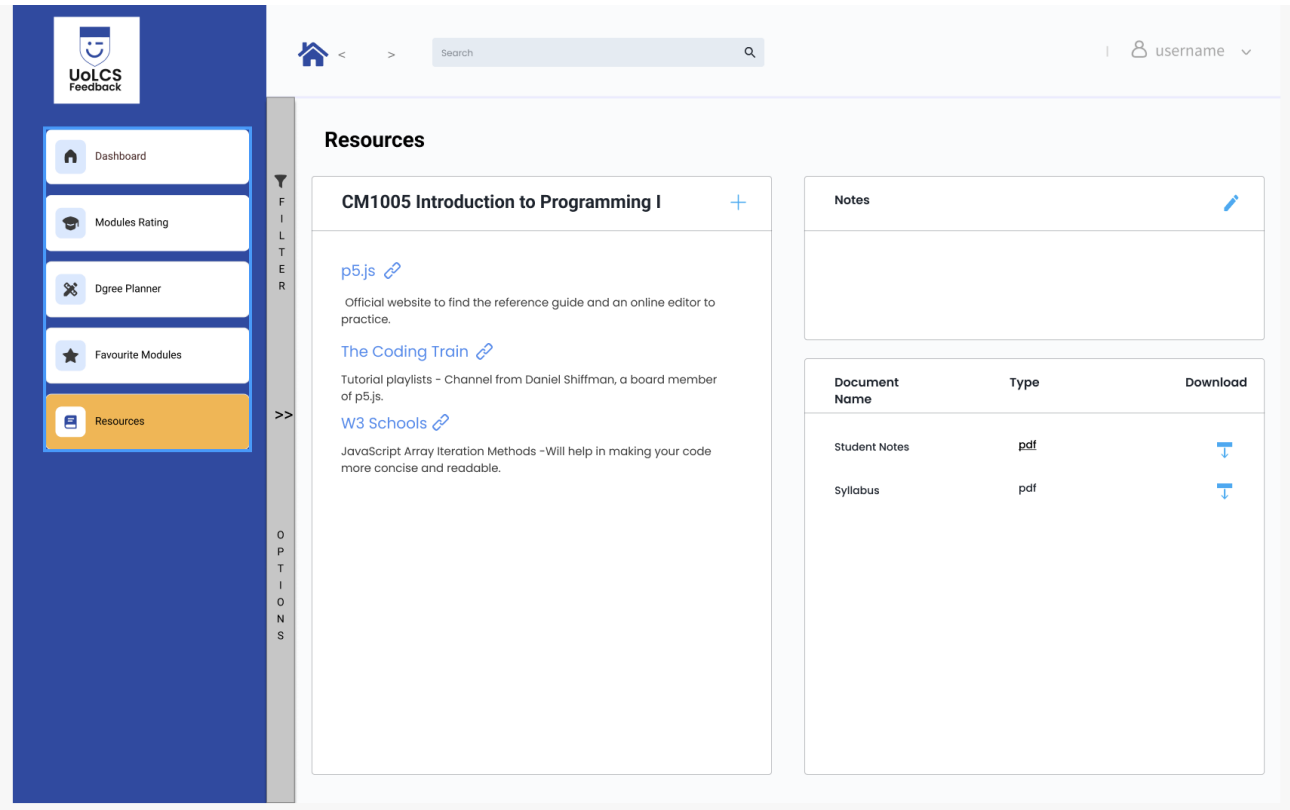


Figure 3.12 – Original hi-fi resources page design

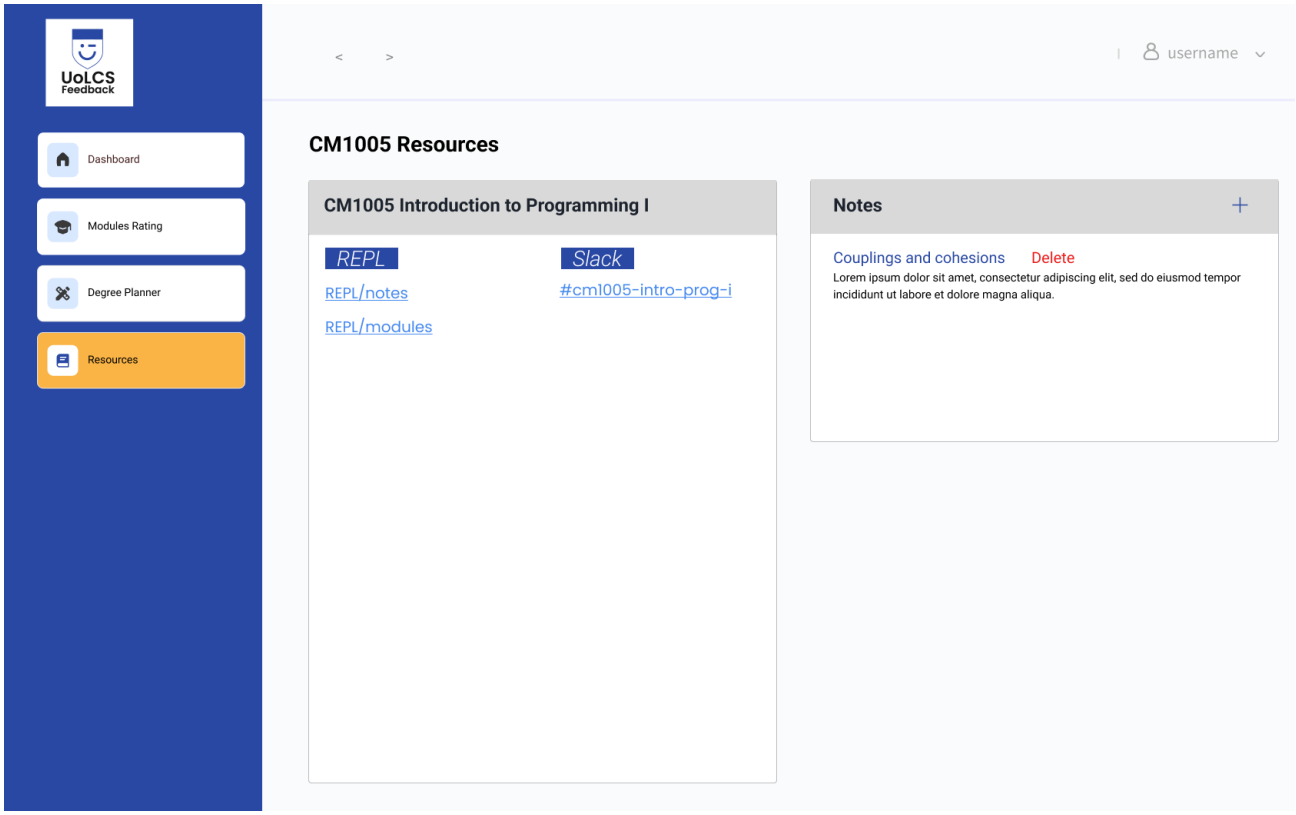


Figure 3.13 – Updated resources page design

3.9.Modifications to the degree planner page

When updating the prototypes for the degree planner page, three features were removed:

- The ability to move modules by semester has been omitted as a feature
- Actual grades will no longer be pulled into the app from coursera
- Drag and drop capability has been removed

The following features were added since the midterm:

- Metrics moved to the top of the screen with better aesthetics
- Ability to add actual and predicted grades next to each module

By assessing application usability, we decided that moving modules by semester will not be greatly beneficial, as users may be more interested in the number of modules remaining and estimated time of completion rather than being pedantic about modules taken by semester. We are no longer pulling actual grade information through an application programming interface (API) connected to coursera, as it may lead to a lot of errors. Coursera is prone to grading errors and wrong grades being released and such issues should not extend into our application. The risk mitigation strategy of removing this feature was captured in the risk register and executed. The drag and drop capability which allows users to rearrange their modules may be useful, but

it will not majorly improve any user experience feature, and it may be implemented in a future release. The updated degree planner page design is found in Figure 3.13.

Aspects of the user interface were improved within the degree planner by moving the metrics to the top. This makes it easier for the users to view this information after making selections. The alignment and placement of the actual and predicted grade fields is convenient for users and may reduce unnecessary scrolling or clicking when making selections and inputting data. This is beneficial as it makes the application easier to use and more visually pleasing.

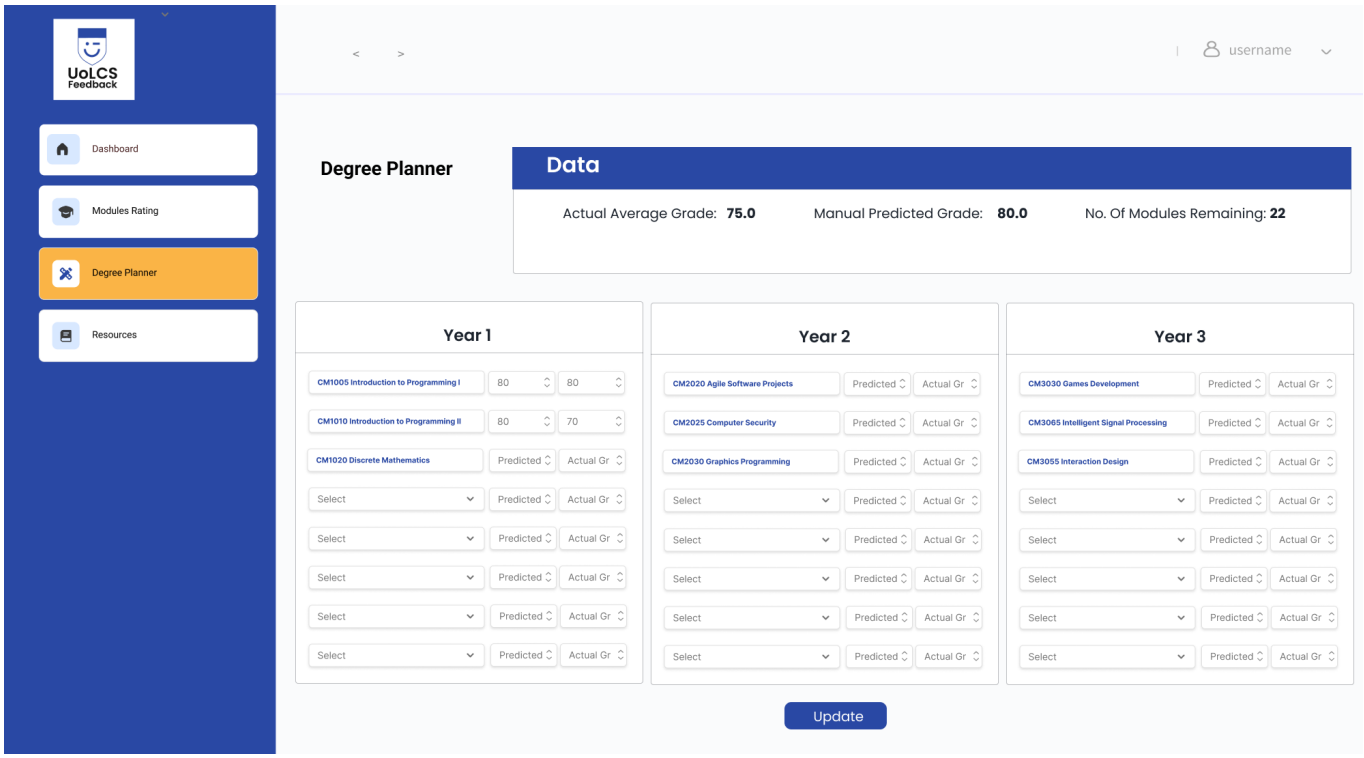


Figure 3.13 – Updated degree planner page

4. Functional requirements, use cases, and change impact assessment

4.1.Functional Requirements

The midterm stated a clear set of functional requirements for the application which were defined based on user research in the form of surveys. Justifications for making changes to the design are shared in section 3, which means that the functional requirements also need to be modified. The original functional requirements are shown in Figure 4.1, and the requirements which have been omitted have been highlighted. There were originally thirty-eight functional requirements, and ten requirements have been omitted based on the justifications provided in section 3.

Func Req ID	Requirement Description	Theme	MoSCoW Priority
FR-001	The web page will have links at the top of the page to allow users to navigate to specific sections of the page.	Main Web page	Should
FR-002	There will be two buttons on the top right of the webpage to allow users to sign in or sign up.	Main Web page	Must
FR-003	There will be a carousel to display images on the page	Main Web page	Should
FR-004	The features section will display four images with descriptions of all four features.	Main Web page	Must
FR-005	The app logo will be displayed on the top left of the screen.	Web Page General	Must
FR-006	Will display a link at the top of the screen to allow the user to navigate back to the main web page	Sign Up Page	Must
FR-007	There will be two buttons on the top right of the webpage to allow users to navigate to the sign up or sign in pages.	Sign Up Page	Must
FR-008	A form will be displayed prompting the user to enter information in the respective fields	Sign Up Page	Must
FR-009	There will be two buttons on the top right of the webpage to allow users to navigate to the sign up or sign in pages.	Sign In Page	Must
FR-010	Will display a link at the top of the screen to allow the user to navigate back to the main web page	Sign In Page	Must
FR-011	A form will be displayed prompting the user to their username and password so that they can sign in to the application. An option also needs to be given to sign in using other methods (such as google sign in, etc).	Sign In Page	Must
FR-012	A navigation bar exists on the left of the page, with icons to navigate to each page of the web application.	Application General	Must

FR-013	A burger bar icon will be positioned above the navigation bar to allow the navigation bar to expand and collapse when clicking the icon.	Application General	Should
FR-014	A filter menu exists on a selection of web pages to allow users to filter items (if applicable to the web page).	Application General	Must
FR-015	A filter bar exists on the left side of the page to allow users to remove filters. It should be positioned to the right of the navigation bar.	Application General	Must
FR-016	A blue and grey colour scheme should be used across all pages of the application in accordance with the visuals provided in the wireframes. A similar shade of blue and grey should be chosen.	Application General	Must
FR-017	An icon that resembles a 'person' will appear at the top right of each screen of the application, allowing users to see their account menu and allowing them to sign out.	Application General	Must
FR-018	A home icon, back icon and forward icon need to exist for each page of the web application, positioned next to the application logo on the top left of the page.	Application General	Must
FR-019	The heading of the respective page will be positioned at the top centre of each page of the application.	Application General	Must
FR-020	A table needs to be displayed providing information about each module and the associated overall ratings for each module.	Module Ratings Page	Must
FR-021	The user needs to be able to hover over and click the text within each table record. This will allow the user to navigate to the module ratings for a specific module.	Module Ratings Page	Must
FR-022	Star icons need to be used within the ratings column, and should be red in colour as seen in the wireframes.	Module Ratings Page	Should
FR-023	Each individual module ratings page will contain additional rating metrics.	Module Ratings Page	Must
FR-024	An area for viewing and leaving comments will be available for individual module ratings pages.	Module Ratings Page	Must
FR-025	An icon should exist on the individual module ratings pages, which allows the user to navigate to the favourite modules page.	Module Ratings Page	Should
FR-026	A table will exist, displaying the name of each module, and the ratings for each module. The user can assign an overall rating for each module.	Favourite Modules	Must
FR-027	The user can click on individual modules and navigate to a page where they can assign ratings for individual metrics.	Favourite Modules	Must
FR-028	All favourite ratings need to feed back into the module rating scores visible on the module ratings pages.	Favourite Modules	Must

FR-029	A table needs to be displayed showing three columns as default for year 1,2 and 3. The user can add more columns by clicking on the column icon on the top right of the page.	Degree Planner Page	Must
FR-030	The table needs to have a total of six rows by default, with the ability to add on more rows (via the rows icon on the top right of the page). Each row represents one semester.	Degree Planner Page	Must
FR-031	The user can add modules by clicking on the plus icon on the top right of the screen. A module selection pop up box will appear, prompting the user to select one or more modules to add to the table	Degree Planner Page	Must
FR-032	The user can click a module and add manual data.	Degree Planner Page	Must
FR-033	Modules which have already been taken will be fixed to the table within the semester they were taken. Key data will also be pulled in from coursera.	Degree Planner Page	Must
FR-034	The user will have the ability to drag and drop modules within different semesters or years.	Degree Planner Page	Must
FR-035	Key course metrics will be displayed at the bottom third of the page, allowing the user to see insightful data associated with their degree performance.	Degree Planner Page	Must
FR-036	A table will exist, containing a list of all modules named by default.	Resources Page	Must
FR-037	The user will be able to hover over each record in the table, and click the record, taking them to the resources page for the chosen module	Resources Page	Must
FR-038	Each dedicated resource page will display information regarding exam dates, readings, past papers and additional resources.	Resources Page	Must

Figure 4.1 – Functional requirements modified by removing highlighted rows

4.2.Use cases

The use cases presented in the midterm will all remain the same, except for the ‘favourite modules’ part of the application, which is being omitted from the application.

4.3.Change Impact Assessment

The changes in the design lead to a very critical question - are we still creating an application which is desired by users if we are choosing to leave out certain features? In order to understand the impact, we must clearly list the features that are being left out. The omitted features, the change descriptions, impact description, and impact scores are show in in Figure 4.2. The table only describes the changes that are considered as negative. This does not mean that further modifications will not be made during the application design and testing, but it does focus on the impact of changes as part of the prototyping phase.

Application Page	Change Description	Impact Description	Impact Score
Landing Page	Footer design modification and change of names from login to sign-in.	This is a low impact change as it would not affect the user's needs. The footer design modification will only enhance the user experience by providing access to social media sites.	Low
Module Ratings	The ability to upvote other user's reviews has been omitted	This feature was not highlighted as a high priority feature when conducting user surveys in the midterm. It is a nice feature to have, but it can be considered as low impact to the needs of users.	Low
General overall application	Burger bar icon is no longer being included in the application to retract the navigation bar	This change will not allow users to retract the main navigation bar, however, this was considered as being a low priority need inside the user surveys.	Low
General overall application	The filter menu is no longer being implemented in the application to filter table items.	This was considered as a desirable feature when conducted user research and surveys. Users would like to filter table items based on specific filter settings.	Medium
General overall application	Search functionality not being added	It is useful for the data to be searchable. For example, if a user has many notes in the resource page, they can easily search them.	Medium
Favourites Modules	This page will no longer exist in the application.	There is no desire for users to have an area within the application for favourites modules. The functionality already exists within other pages of the application.	Low
Resources page	Links to REPL and Slack have been provided instead of providing links to all individual resources within modules	Users will still be able to access all resources. Both REPL and Slack are kept up to date and are in compliance with UK laws.	Low
Degree planner	The ability to move modules by semester has been omitted as a feature	Although it is a nice feature to have, the user can still identify the semester of a module without it.	Low
Degree planner	Actual grades will no longer be pulled into the app from coursera.	Pulling grades from coursera can lead to unnecessary errors, so it is better not to integrate Coursera.	Low
Degree planner	Drag and drop capability has been removed	Users can just select from the dropdown menu, so the drag and drop feature is not needed.	Low

Figure 4.2 – Change impact assessment of omitted features

4.4. Ethical considerations

Further research into ethical considerations was conducted since the midterm, especially since there have been design changes. Our company logo does not contain the same content or font as the University of London logo (Figures 4.3). In addition, the size of the text is larger without explicitly mentioning the University of London, which ensures a lack of infringement of intellectual property rights around the design and content of the logo.

Figure 4.3 – University of London logo vs Student Application Logo

To avoid any misunderstanding and to emphasises that we are not affiliated or endorsed by the university, we have placed a disclaimer within the landing page of the website, stating that we are not affiliated with the university.

Our application also makes use of various assets found inside the static files within the application. Icons are sourced from an organisation named Font Awesome. These are free to use, open source, and general public licence friendly. This means that they can be used for commercial purposes without having to ask for permission. Within the front-end design, we have also used multiple navigation and function icons from the basil icon provided by Craftwork. As the copyright license of these icons allow personal and non-commercial use for free, we found the copyright sufficient for the development of the coursework project. We sourced the hero banner graphic on our landing page from Freepik Company S.L. As stated on the image's copyright license, we are allowed to 'use the image in a commercial setting if giving attribution to the author.'[16] As such, we placed an attribution text on the landing page.

In the resources section of our application, we aimed to provide our users with various types of relevant material for each module of the degree. To achieve this objective, links are provided redirecting the user to the Resources Enriching Perennial Learners (REPL) website. The REPL website not only contains information and links related to many aspects of the degree but also hosts PDF files containing student notes, past exams, and syllabuses. The REPL project is explicitly licensed under the MIT license, so the hosted exam and syllabus files are under the copyright of University of London. We decided to use REPL rather than hosting resources within our application to avoid violating intellectual property rights.

Our application hosts user data such as names, email addresses, and passwords. An important ethical and legal consideration for our project is the handling of user data. We are conforming to the United Kingdom's Data Protection Act of 2018 (which includes the General Data Protection Regulation) by collecting user data fairly and transparently for the stated purpose of assisting students in their pursuit of the degree. We will not collect sensitive information such ethnic, religious, political, or biological attributes of users that would require stronger protection. The Data Protection Act requires the user data to be stored securely, so we have ensured that the data stored in the database is encrypted. The data will be stored for as long as the user maintains the account.

5. Application Design

5.1.Process Design

Some individuals think of process and product as separate entities; however, our team believes that they go hand in hand. An efficient process usually produces a good product, which is why a strong emphasis was given to the process design. Figure 5.1 shows the process flow diagram that was created as part of the midterm. The original process flow diagram modelled the different pages of the application, model decision paths, and identified additional improvements that need to be made (as seen in the sticky notes). For example, by entering the website, the user has the choice to sign-in or sign-up. After performing either task, they enter the main page of the application, and at this point, the user has many potential options when navigating to different pages of the application. When working towards the final design of the application, improvements to the process flow were completed within the first three sprints The completed process flow diagram for the application can be seen in Figure 5.2.

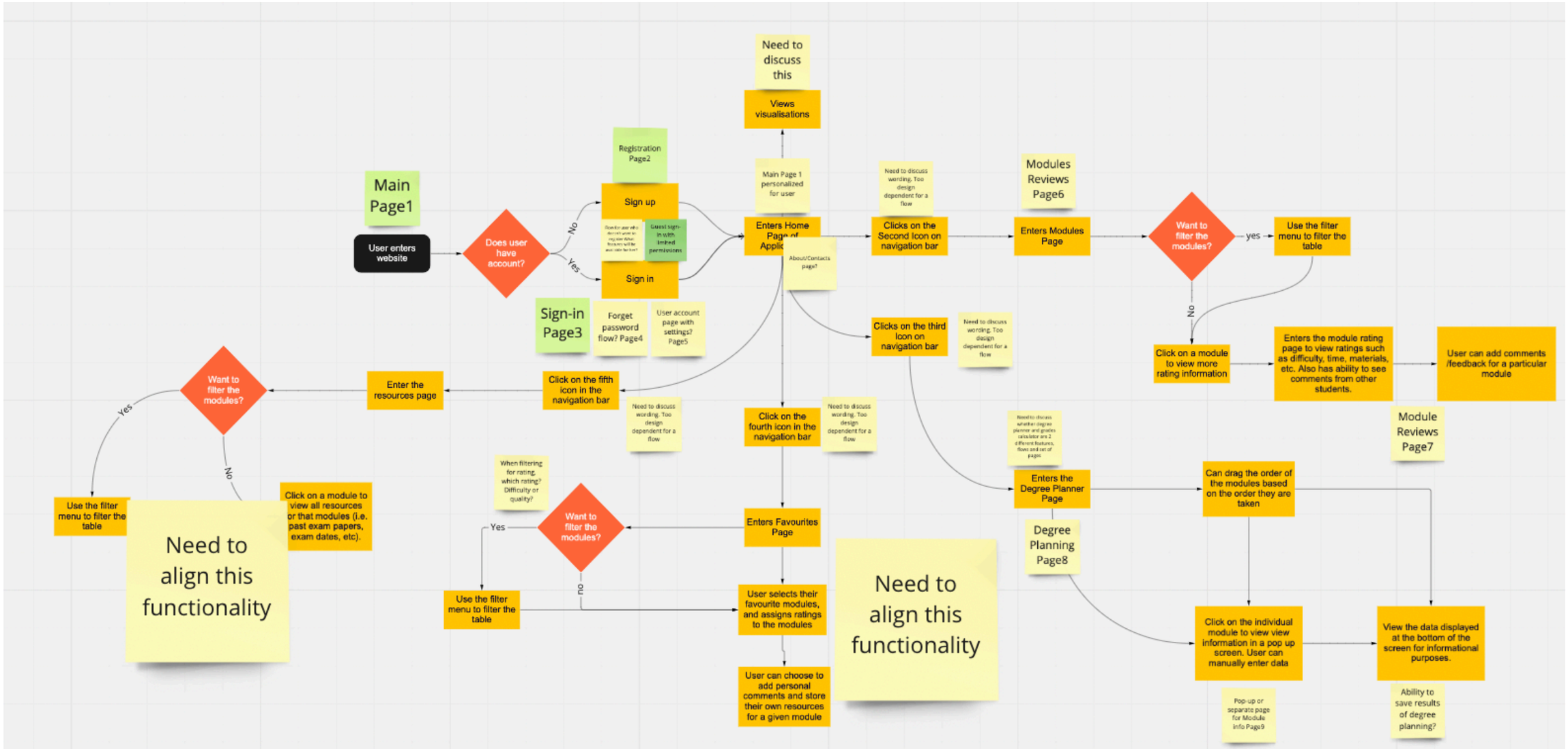


Figure 5.1 – Midterm Process Flow Diagram

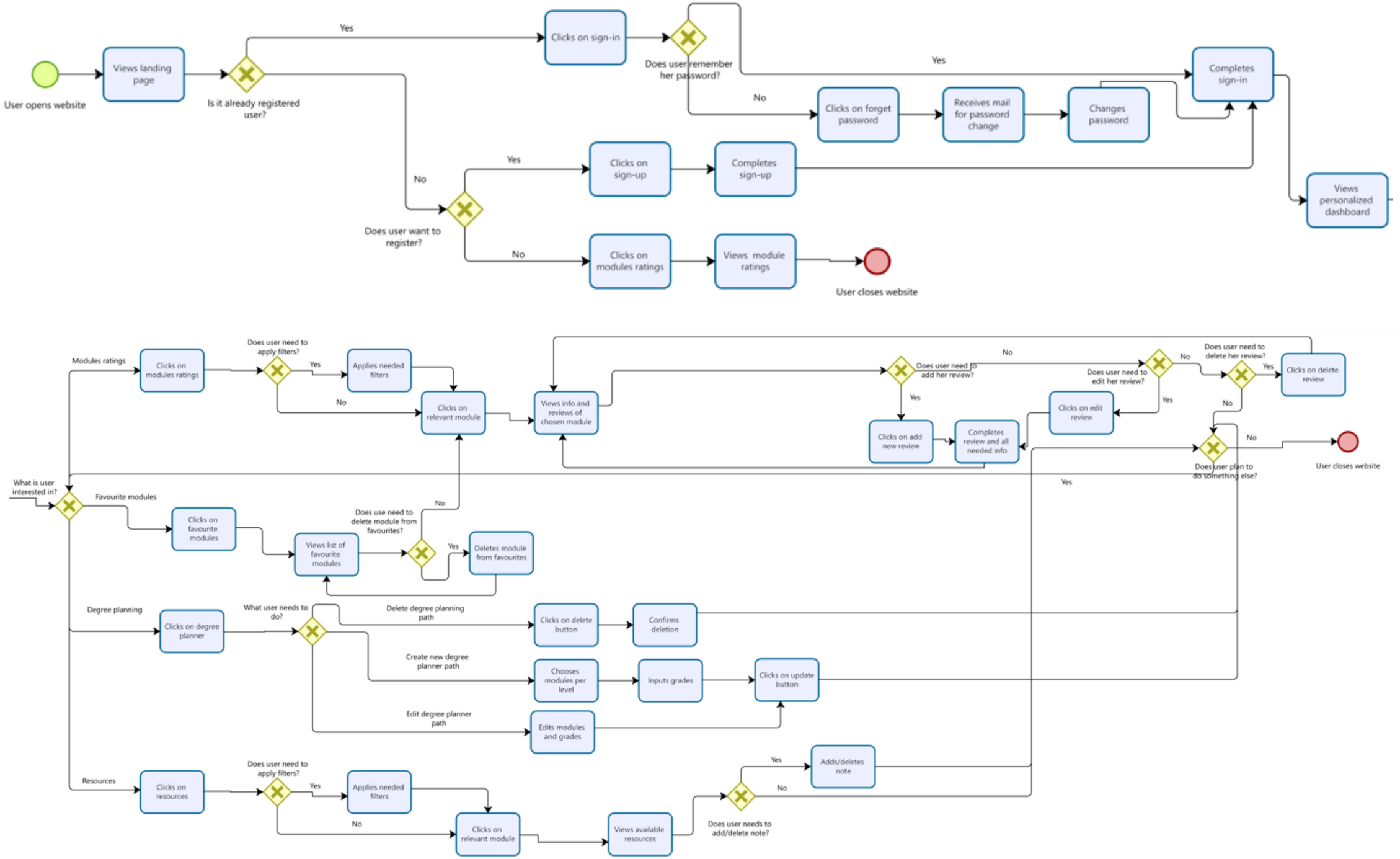


Figure 5.2 – Completed Process Flow Diagram

The original process flow diagram shown in Figure 5.1 states the need to finalise various features such as: forgot password, visualisations in the dashboard, the filter menu, saving results for degree planning, etc. The justifications for adding or removing some of these features have been provided in section 3 as part of the wireframe modifications. The assigned Process Engineer liaised with the UX Designer and the Application Developer in order to define a highly accurate depiction of all process steps during a range of sprints.

5.2.System Packages and libraries

When developing the web application, a range of packages and libraries were used, which are shown in the list within Figure 5.3 (and seen in the requirements.txt file within the application src folder). All the versions utilised have also been stated within Figure 5.3.

```
alembic==1.8.1
blinker==1.4
click==8.1.3
Flask==2.1.3
Flask-Login==0.6.1
Flask-Mail==0.9.1
Flask-Migrate==3.1.0
Flask-SQLAlchemy==2.5.1
Flask-WTF==1.0.1
greenlet==1.1.2
importlib-metadata==4.12.0
itsdangerous==2.1.2
Jinja2==3.1.2
Mako==1.2.1
MarkupSafe==2.1.1
python-dotenv==0.20.0
SQLAlchemy==1.4.39
Werkzeug==2.1.2
WTForms==3.0.1
zipp==3.8.1
```

Figure 5.3 - List of packages and libraries used in the application

Each of the packages/libraries have been described in detail within Figure 5.4 on the next page. Flask comes with a preconfigured database management system named SQLite. You can go to the config.py file to see which database Flask is using.

Alembic	Alembic is a is a lightweight library for data migration employing SQLAlchemy, the ORM framework. Changes are saved in version files which enables simple debugging and tracking.
Blinker	Blinker is a Python library for practical object-to-object signalling. It lets multiple parties to subscribe to events and signal recipients to subscribe to senders.
Click	Click is used to create custom command line interfaces on Python.
Flask	Flask is a lightweight web development framework in Python that allows for powerful yet simple customizations via its plug-in library.
Flask-Login	Flask-Login manages user sessions on Flask.
Flask-Mail	Flask-Mail is a Flask extension that adds email functionality by connecting to an SMTP server and delivering emails to the server.
Flask-Migrate	Flask-Migrate is a Flask extension to handle SQLAlchemy database migrations via Alembic.
Flask-SQLAlchemy	Flask-SQLAlchemy is a Flask extension to enable SQLAlchemy support.
Flask-WTF	Flask-WTF is a Flask extension to integrate WTForms to validate user request data in web forms.
Greenlet	Greenlet is Python extension that enables lightweight coroutine switching.
Importlib-metadata	Importlib-metadata is a Python library to access metadata of installed packages.
Itsdangerous	Itsdangerous is a Python library that provides helpers to transmit encrypted data via HMAC and SHA1 signatures.
Jinja2	Jinja2 is a popular Python templating engine to automatically generate documents from a simple text format.
Mako	Mako is a template library to generate Python code from various text contents.
MarkupSafe	Markupsafe is a Python library to generate escape characters in HTML.
Python-dotenv	Python-dotenv is a Python package to access and set environment variables.
SQLAlchemy	SQLAlchemy is an Object Relation Mapper (ORM) suite that acts as an SQL toolkit for Python.
Werkzeug	Werkzeug is a Web Server Gateway Interface (WSGI) utility library for Python.
WTForms	WTForms is a Python library to handle and validate web form inputs from users.
Zipp	Zipp is a pathlib-adaptable ZIP-file object wrapper for Python.

Figure 5.4 – Description of each package and library used for Flask web application

5.3. Design architecture and structure

It was decided in section 2 that there would a change in the proposed web development framework and structure for application development. This sub-section details the new structure and the new architecture model. When opening the application inside an integrated development environment, it is highly recommended to follow the readme.md file which is found inside the submission file. This provides a step-by-step process to open the application. The contents of readme.md file have been shown in Figure 5.5.

```
## Setup

Clone repo.

Change into main directory.

Create python virtual environment.

> python -m venv .venv

Activate venv.

> . ./.venv/bin/activate

Install depedencies.

> pip install -r requirements.txt

Change into src directory.

Setup database (every time you need to reset the db).

> flask db init && flask db migrate && flask db upgrade

Run server.

> flask run
```

Figure 5.5 – Opening the application

After creating the virtual environment and executing all necessary steps to open the application, a range of files and folders can be seen inside the application src folder. This structure is shown in Figure 5.7, however, before discussing the structure, the application architecture diagram is shown in Figure 5.6.

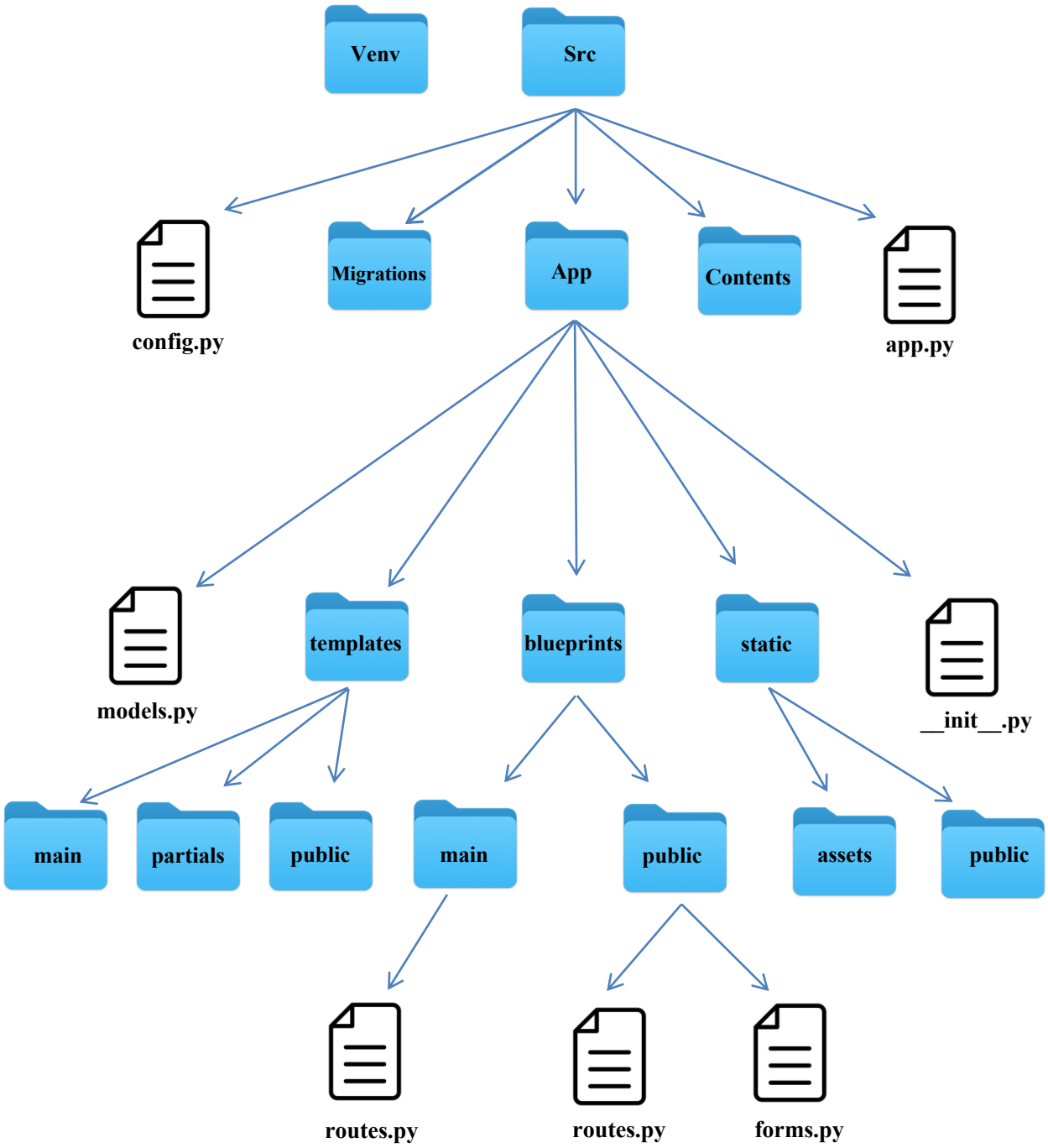
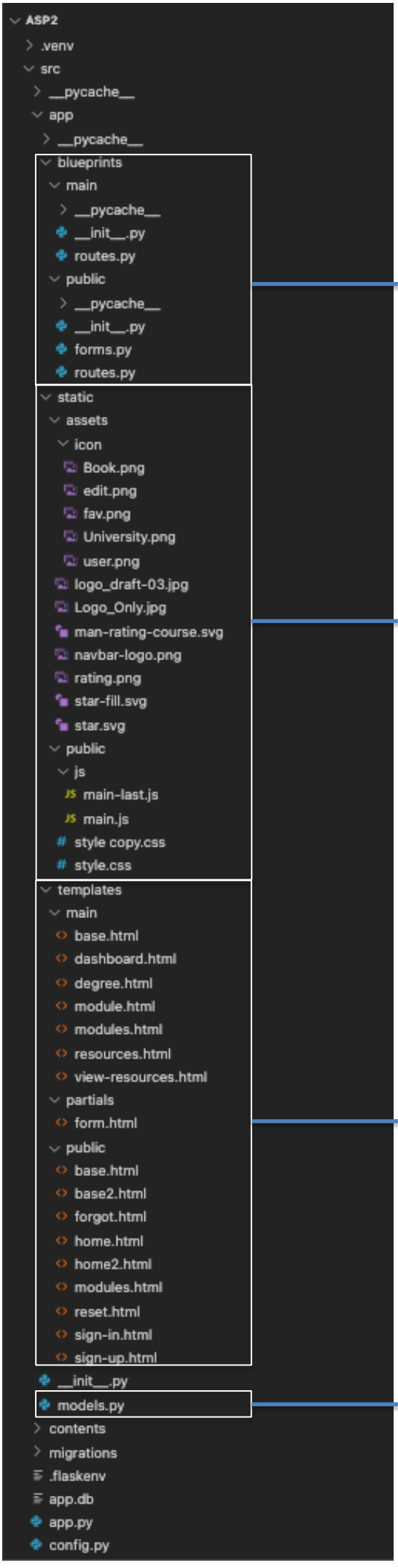


Figure 5.6 – High-level application architecture model

The full structure of the completed application is showed in Figure 5.7. Most of the code for the application is situated inside the app folder inside the src folder. Each file within the app folder is described in detail in the subsequent sections.



The blueprints folder contains a main folder and public folder. The main folder contains routes for web pages once the user signs into the application. The public folder contains routes and a forms.py file for web pages before the user signs into the application.

The static folder contains all files which are static in nature, so they do not change when the application is running, or do not change in response to a user's inputs or actions. Files such as images, logos, icons, CSS files and JavaScript files are all incorporated inside the static folder.

The templates folder contains three folders named main, partials and public. The main folder holds templates for web pages which are accessed after the user signs into the application. The public folder contains templates for web pages that are accessed before the user signs in. The partials folder contains a form which is used across different webpages.

The models.py file allows for the creation of tables within the databases and then stores the data.

Figure 5.7 – Full structure of the application

5.4.Application Development

Figure 5.7 details the application structure, however, this section dives into more detail on each area of the structure and provides insight into the development of the application for each of these components. Each of the completed pages of the application will also be displayed inside this section.

5.4.1. Models.py

When developing the application, there was a need to store data for the web application within a database. The models.py file allows for the creation of tables within the database to store the data. Models are the blueprints to create the tables, and models are nothing but classes in python, which are created inside the models.py file. Each time a database schema or model is created, a class must be created inside the models.py file. The code for the models.py file is provided in Figure 5.8 which shows that four classes have been created named User, Module, Rating, and UserNotes. Models are not database tables themselves, but they are the blueprints which are needed to create the tables within the database, and the class stores the relevant variables and functions in order to transact the data. For example, table columns are created in the database from the User class variables named id, email, password, and name with datatypes integer and string respectively. Two functions have also been incorporated to set the user's password and to check the user's password (to check if it matches). This is enabled by the use of the werkzeug.security library. Other libraries that have been imported are flask_login and sqlalchemy.

The class named module allows for the creation of tables within the database named code, name, professor and year. Table constrains, primary keys, and foreign keys have also been assigned to tables within the database. For example, the Rating class has variables named userId and moduleCode which represent two columns within the database table named Rating. The userId is a foreign key to the column named Id, represented within the UserNotes class. The moduleCode is a foreign key to the column named code, represented within the Module Class. Both columns named Id and code are primary keys within their respective tables. The database has been set up in a relational manner, however, the tables have not been normalised. This is a potential future improvement, where the tables can be normalised to third normal form at minimum.

```python
from enum import unique
import json

from flask_login import UserMixin
from sqlalchemy import event, ForeignKey
from werkzeug.security import generate_password_hash, check_password_hash


from app import db, login

@login.user_loader
def load_user(id):
    return User.query.get(id)


class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(500), unique=True, index=True)
    password = db.Column(db.String(500))
    name = db.Column(db.String(500))
    reviews = db.relationship('Rating', backref='user')

    def set_password(self, password):
        self.password = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password, password)



class Module(db.Model):
    code = db.Column(db.String(20), primary_key=True)
    name = db.Column(db.String(100))
    professor = db.Column(db.String(100))
    year = db.Column(db.Integer)
    ratings = db.relationship('Rating', backref='module')
    userNotes = db.relationship('UserNotes', backref='module')

class Rating(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    userId = db.Column(db.Integer, ForeignKey(User.__tablename__ + '.id'), nullable=False)
    moduleCode = db.Column(db.Integer, ForeignKey(Module.__tablename__ + '.code'), nullable=False)
    time = db.Column(db.Integer)
    midterm = db.Column(db.Integer)
    final = db.Column(db.Integer)
    lectures = db.Column(db.Integer)
    usefulness = db.Column(db.Integer)
    pros = db.Column(db.Text)
    cons = db.Column(db.Text)
    helpful = db.Column(db.Integer, default= 0)
    nothelpful = db.Column(db.Integer, default= 0)
    created_on = db.Column(db.DateTime, server_default=db.func.now())


class UserNotes(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    userId = db.Column(db.Integer, ForeignKey(User.__tablename__ + '.id'), nullable=False)
    moduleCode = db.Column(db.Integer, ForeignKey(Module.__tablename__ + '.code'), nullable=False)
    title = db.Column(db.String(500))
    content = db.Column(db.Text)
```

Figure 5.8 – models.py file containing all classes

5.4.2. Templates

Templates contain static data but also contain placeholders where dynamic data is rendered. The Jinja templating engine has been used within the application to render the templates. As seen previously in the templates folder structure (Figure 5.7), there are a range of templates created for each of the pages within the application. The public folder stores all templates for pages which the user accesses before being signed into the application. The main folder stores all templates for pages that the user accesses after signing into the application.

As an example, the code within the modules.html file has been shown in Figure 5.9, which shows the html code alongside the use of specific jinja templating blocks. Blocks such as '% for item %' have been used to iterate through dictionaries and then rendering static information based on the item within the database.

```
{% extends 'main/base.html' %}


{% block main %}

<div class="row">
    <div class="col-md-6">
        <h2>Module Ratings</h2>
    </div>

</div>


<div class="content-wrapper mt-4">
    <div class="row">

        <div class="col-md-12">
            <div class="card spur-card">
                    <div class="card-header">
                        <div class="spur-card-title">Module Ratings</div>
                    </div>
                    <div class="card-body ">
                        <table class="table table-in-card" id="usersTable">
                            <thead>
                                <tr>
                                    <th scope="col">Module Name</th>
                                    <th scope="col">Rating</th>
                                </tr>
                            </thead>
                            <tbody>

                                {% for item in modules %}

                                <tr>
                                    <td><a href="/module/{{item.module.code}}" class="module-name link">{{item.module.name}}</a></td>
                                    <td><span class="stars"> {{item.overallAvg}}</span></td>
                                </tr>


                                {% endfor %}


                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

{% endblock %}
```

Figure 5.9 – Code within the modules.html file

5.4.3. Blueprints and routes.py

The application incorporates blueprints which simplify how the application works and provides a way to register operations in applications. For example, a blueprint can be registered multiple times on the application with different URL rules. Within the blueprints folder, there are routes.py files which incorporate the URL patterns i.e. '/main' or'/logout'. The blueprints/main folder contains a routes.py file which defined the routes for the pages that the user interacts with after signing into the application. A short extract of the code is provided in Figure 5.10. The routes.py file created within the blueprints/public folder has code for all routes associated with the pages of the application that the user interacts with before signing into the application. A short extract of the code is provided in Figure 5.11. Please refer to the source code submission to view the full code. The code for '@bp.route( )' is a python decorator that Flask provides which makes it easy to assign URLs to functions. When a user visits the application at the given route, the function defined underneath the route is executed. For example, within the modules route in Figure 5.10, a function named modules is defined which queries the database for all modules and then appends the module, rating, and overall average to the module list. All the average ratings are then rendered to the main/modules.html page. In a similar manner, multiple functions exist within the routes.py files which perform a range of actions to provide the functionality required by users.

```
from ast import walk
from datetime import date, timedelta
import json
from operator import itemgetter, mod
import os
import random
from flask import redirect, render_template, request, session, url_for, flash
from app.models import Module, Rating, User, UserNotes
from flask_login import login_required, logout_user, current_user
from ..main import bp
from app import db, login
from flask import send_file



@bp.route('/main')
@login_required
def main():
    mods = Module.query.all()
    userId = current_user.id
    allReviews = Rating.query.all()
    myreviews = Rating.query.filter_by(userId=userId)

    modules = []

    for mod in mods:
        ratings = getRating(mod.ratings)
        overallAvg = (ratings['time']['count'] + ratings['midterm']['count'] + ratings['final']['count'] + ratings['lecture']['count']  + ratings['usefulness']['count'])/5
        if(ratings['time']['count'] != 0):
            modules.append({'module' : mod, 'rating' : ratings, 'overallAvg' : overallAvg})

    newlist = sorted(modules, key=itemgetter('overallAvg'), reverse=True)


    return render_template('main/dashboard.html', allmodules = mods ,modules = newlist, myreviews = myreviews, allReviews = allReviews)


@bp.route('/logout')
@login_required
def logout():
    logout_user()
    if session.get('was_once_logged_in'):
        # prevent flashing automatically logged out message
        del session['was_once_logged_in']
    return redirect(url_for('main.main'))


@bp.route('/modules')
@login_required
def modules():
    mods = Module.query.all()
    modules = []
    for mod in mods:
        ratings = getRating(mod.ratings)
        overallAvg = (ratings['time']['avg'] + ratings['midterm']['avg'] + ratings['final']['avg'] + ratings['lecture']['avg']  + ratings['usefulness']['avg'])/5
        modules.append({'module' : mod, 'rating' : ratings, 'overallAvg' : overallAvg})

    newlist = sorted(modules, key=itemgetter('overallAvg'), reverse=True)
    return render_template('main/modules.html', modules = modules)
```

Figure 5.10 – extract of code from routes.py within blueprints/main

```python
from operator import itemgetter
from flask import flash, redirect, render_template, url_for
from itsdangerous import Serializer, SignatureExpired, URLSafeTimedSerializer
from app.blueprints.main.routes import getRating
from flask_login import login_user
from app import db
from app.models import Module, User
from app.blueprints.public import bp
from app.blueprints.public.forms import ForgotForm, ResetForm, SignUpForm, SignInForm
from flask import Flask
from flask_mail import Mail, Message



SECRET = 'NK87KJ8KJ9GTHCJEUYRJ83457J89KDSFJDSIOFJ5674G65DF4G65DF4G5DF'


def insertData():
    db.session.add(Module(code='CM1005', name='Introduction to Programming I', professor = 'Professor Dr. XYZ', year = 1))
    db.session.add(Module(code='CM1006', name='Introduction to Programming II', professor = 'Professor Dr. XYZ', year = 1))
    db.session.add(Module(code='CM1007', name='Computational Mathematics', professor = 'Professor Dr. XYZ', year = 1))
    db.session.add(Module(code='CM1008', name='Discrete Mathematics', professor = 'Professor Dr. XYZ', year = 1))
    db.session.add(Module(code='CM1009', name='How computers work', professor = 'Professor Dr. XYZ', year = 1))
    db.session.add(Module(code='CM1010', name='Fundamentals of computer science', professor = 'Professor Dr. XYZ', year = 1))
    db.session.add(Module(code='CM1011', name='Web Development', professor = 'Professor Dr. XYZ', year = 1))
    db.session.add(Module(code='CM1012', name='Algorithms and Data structures I', professor = 'Professor Dr. XYZ', year = 1))

    db.session.add(Module(code='CM1013', name='Object Oriented Programming', professor = 'Professor Dr. XYZ', year = 2))
    db.session.add(Module(code='CM1014', name='Software design and development', professor = 'Professor Dr. XYZ', year = 2))
    db.session.add(Module(code='CM1015', name='Databases networks and the web', professor = 'Professor Dr. XYZ', year = 2))
    db.session.add(Module(code='CM1016', name='Agile Software Projects', professor = 'Professor Dr. XYZ', year = 2))
    db.session.add(Module(code='CM1017', name='Computer Security', professor = 'Professor Dr. XYZ', year = 2))
    db.session.add(Module(code='CM1018', name='Graphics Programming', professor = 'Professor Dr. XYZ', year = 2))
    db.session.add(Module(code='CM1019', name='Algorithms and Data structures II', professor = 'Professor Dr. XYZ', year = 2))
    db.session.add(Module(code='CM1020', name='Programming with data', professor = 'Professor Dr. XYZ', year = 2))

    db.session.add(Module(code='CM1021', name='Data Science', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1022', name='Databases and advanced data techniques', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1023', name='Machine Learning and neural networks', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1024', name='Artificial Inteligence', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1025', name='Virtual Reality', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1026', name='Games Developement', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1027', name='Advanced Web Developement', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1028', name='Phsical Computing and the Internet of things', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1029', name='3D Graphics and Animation', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1030', name='Mobile Development', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1031', name='Interaction Design', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1032', name='Natural Language Processing', professor = 'Professor Dr. XYZ', year = 3))
    db.session.add(Module(code='CM1033', name='Intelligent Signal Processing', professor = 'Professor Dr. XYZ', year = 3))
    db.session.commit()


@bp.route('/')
def home():
    mods = Module.query.all()
    if(len(mods) == 0):
        insertData()
    return render_template('public/home.html')



@bp.route('/sign-up', methods=['GET', 'POST'])
def sign_up():
    form = SignUpForm()

    # if form.password.data != form.password_confirm.data:
    #   flash('Password and Confirm Password do not match')
    #   return render_template('public/sign-up.html', form=form)

    if form.validate_on_submit():
        user = User(
            email=form.email.data,
            name=form.name.data
        )
        user.set_password(form.password.data)

        db.session.add(user)
        db.session.commit()

        flash('User sign up successful.')

        return redirect(url_for('public.sign_in'))


    return render_template('public/sign-up.html', form=form)
```

Figure 4.11 – extract of code from routes.py within blueprints/public

5.4.4. Static files

As shown previously in Figure 5.7, static files incorporate images, icons, logos, CSS files, and JavaScript files. The logo and various images used across all sites are found within the status file. An example of functions within the JavaScript files are event listeners which wait for an event to occur and then execute a second argument once the event is triggered. The Cascading Style Sheet (CSS) is used to style and layout pages within the application, and defines styles for components such as fonts, colour, size, and spacing,

5.5.Development using Microsoft Azure DevOps

After conducting research and reviewing literature in section three, it was decided that Microsoft Azure DevOps would be used for development operations and to assist with agile methods. One of the key benefits of using the agile method is to release benefits throughout the process rather than releasing benefits at the end of the project. According to agile literature, 'Sprints break down a project into bite-sized chunks. Teams plan a single sprint at a time and adapt future sprints based on the outcome of the previous one.' [17] Microsoft Azure DevOps was used to 'track work with configurable Kanban boards, iterative backlogs, and powerful planning tools.' [18] The Microsoft Azure DevOps user interface is shown in Figure 5.11.

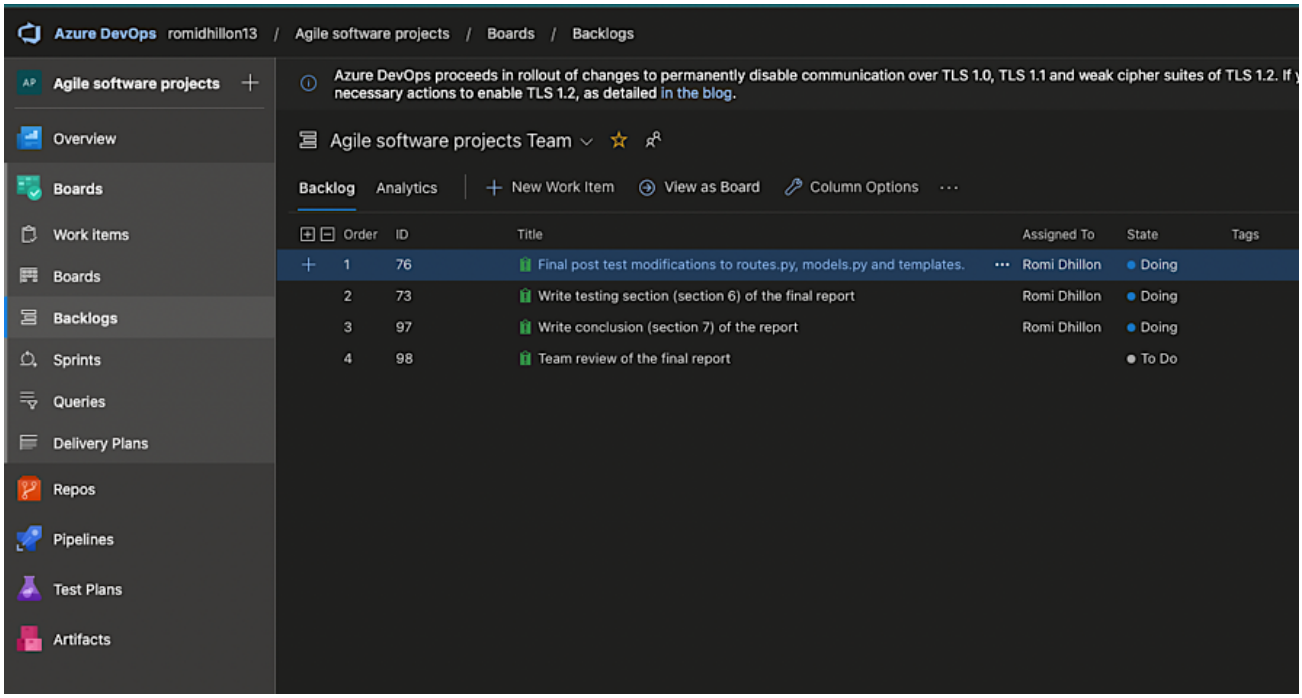


Figure 5.11 – Microsoft Azure DevOps user-interface

A list and description of each DevOps service used as part of the project is provided below:

- Overview: an area to provide a description of the project
- Boards: a Kanban board is used as an agile project management tool to visualise work and reduce work-in-progress. It helps agile teams to produce order in their daily work.
- Backlogs: shows any work items that have been scheduled within a sprint that need to be completed as part of the project.

- Sprints: shows all work items that have been scheduled for specific sprints
- Delivery Plans: allows for the creation of a delivery plan based on sprints
- Repos: GitHub repository integration
- Test Plans: to capture and document all usability and user-interface tests

The development operations conducted as part of boards, backlogs, and sprints are detailed in the subsequent subsections.

5.5.1. Sprints

The length of a sprint can vary based on the duration of the project and the nature of work. The team agreed that each sprint would have a duration of one week and that there would be seven sprints in total. Figures 5.12 and 5.13 shows the sprint details and delivery plan.

Iterations	Start Date	End Date
Agile software projects		
Sprint 1	10/07/2022	17/07/2022
Sprint 2	17/07/2022	24/07/2022
Sprint 3	24/07/2022	31/07/2022
Sprint 4	31/07/2022	07/08/2022
Sprint 5	07/08/2022	14/08/2022
Sprint 6	14/08/2022	21/08/2022
Sprint 7	21/08/2022	28/08/2022

Figure 5.12 – Azure DevOps sprint details

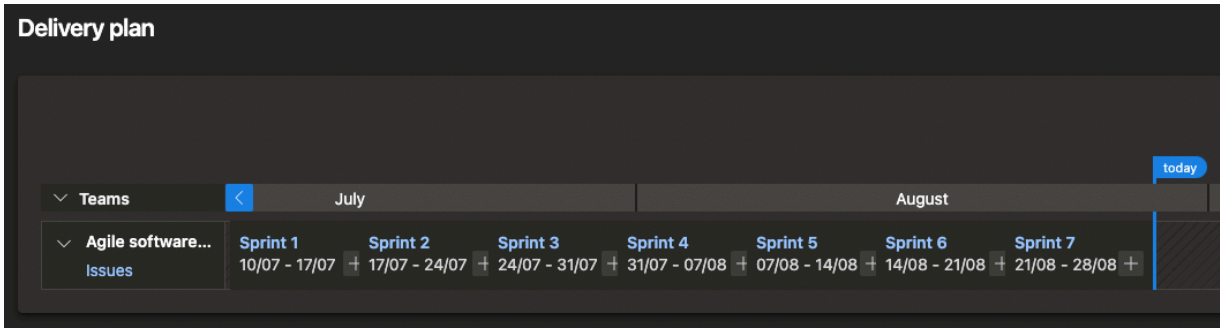


Figure 5.13 – Azure DevOps sprint delivery plan

5.5.1.1. Sprint 1

Figure 5.14 shows the activities for sprint 1 within the Azure DevOps sprints area. The titles of the sprint activities can be seen as separate items, and each activity is assigned to a team member. The state of each activity can be concluded as to-do, doing, or done. Sprint 1 focused on assigning the project/technical lead and the roles and responsibilities of the team. The high-level project plan was also created, and the basic user model was created using Flask. All

requirements from the available tutor video on Coursera were captured, and assets were obtained for the application.

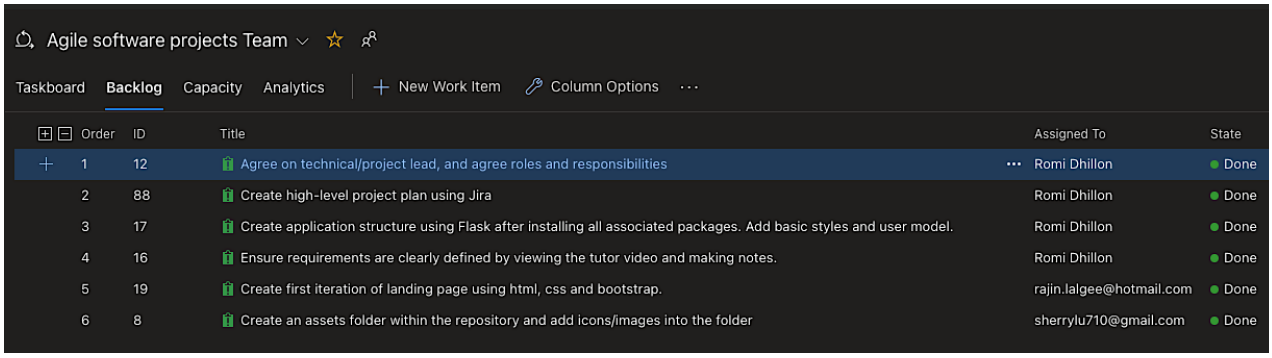


Figure 5.14 – Azure DevOps Sprint 1 tasks

5.5.1.2. Sprint 2

All sprint 2 activities can be seen in Figure 5.15. There was a focus on developing main parts of the application during this sprint such as blueprints, routes and the database (via models.py). User experience and user interface research along with modifications to previous hi-fi wireframes was vital, as the updates to the wireframes were required in time to be implemented inside the application. A literature review was conducted to complement the technologies, methods, tools and techniques that were being used in the project. The process flow diagram was also developed to its first iteration, to ensure that all user activities are executed in the most efficient manner. The team wanted to start the final report early in the project, so the introduction for the report was written within this sprint.

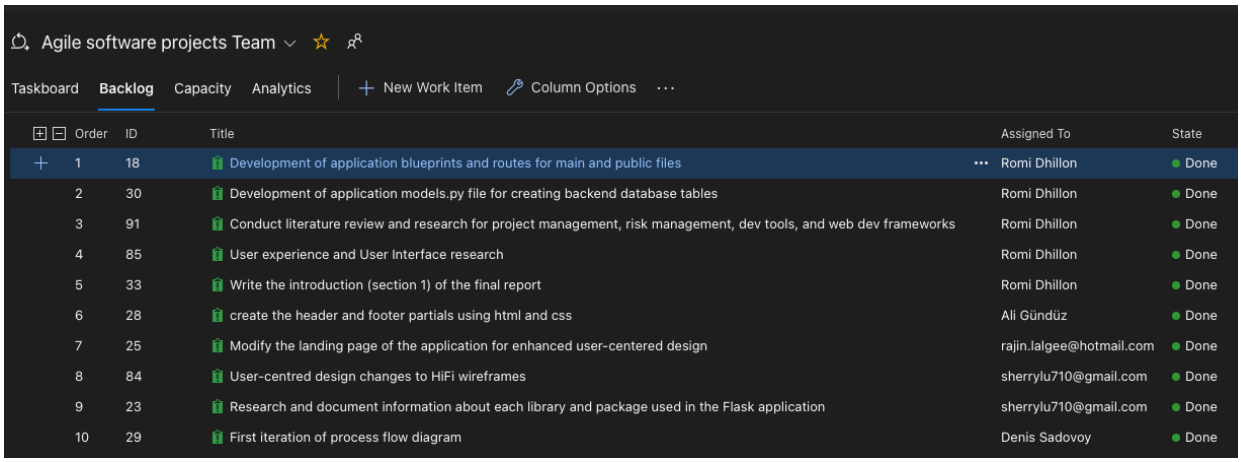


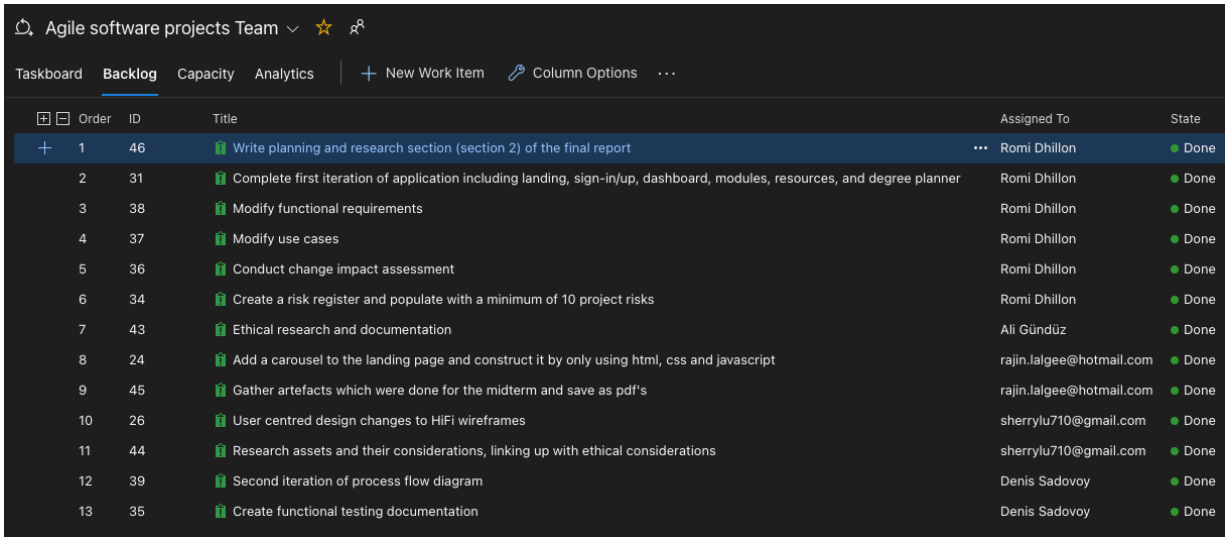
Figure 5.15 – Azure DevOps Sprint 2 tasks

5.5.1.3. Sprint 3

Now that the team had been through two previous sprints, the workload was increased in comparison to previous sprints as seen in Figure 5.16. The second section of the report was written by the end of this sprint after competing modifications to the functional requirements and use cases. A change impact assessment was conducted to assess the impact of changes in the wireframes, design, and functional requirements. A risk register was created to ensure that

all risks are captured and reviewed at the end of future sprint meetings. The first iteration of the application was completed by the end of sprint 3.

The hi-fi wireframes were further developed from the previous sprint, and the design changes were fed back into the application and the process flow diagram.

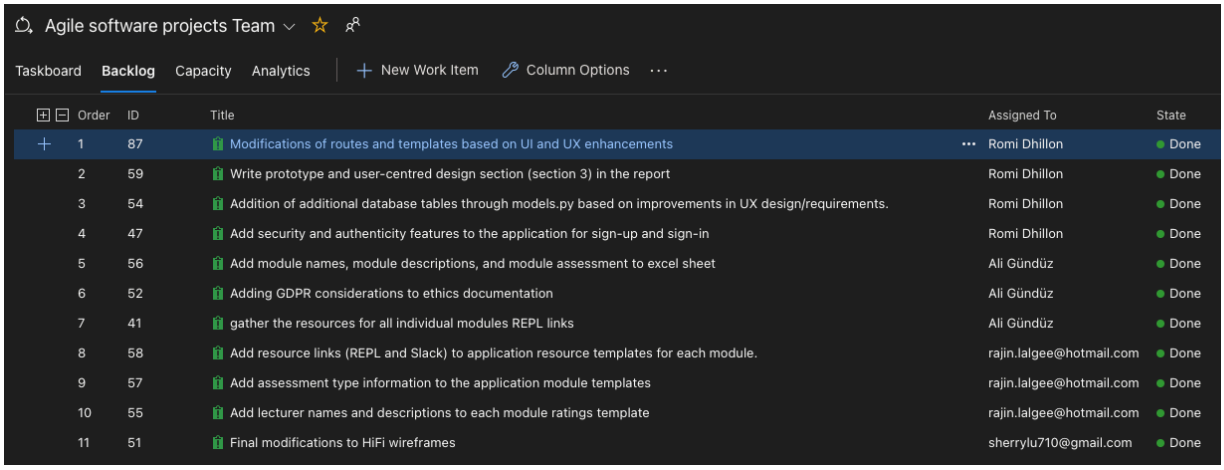


Order	ID	Title	Assigned To	State
1	46	Write planning and research section (section 2) of the final report	Romi Dhillon	Done
2	31	Complete first iteration of application including landing, sign-in/up, dashboard, modules, resources, and degree planner	Romi Dhillon	Done
3	38	Modify functional requirements	Romi Dhillon	Done
4	37	Modify use cases	Romi Dhillon	Done
5	36	Conduct change impact assessment	Romi Dhillon	Done
6	34	Create a risk register and populate with a minimum of 10 project risks	Romi Dhillon	Done
7	43	Ethical research and documentation	Ali Gündüz	Done
8	24	Add a carousel to the landing page and construct it by only using html, css and javascript	rajin.lalgee@hotmail.com	Done
9	45	Gather artefacts which were done for the midterm and save as pdf's	rajin.lalgee@hotmail.com	Done
10	26	User centred design changes to HiFi wireframes	sherrylu710@gmail.com	Done
11	44	Research assets and their considerations, linking up with ethical considerations	sherrylu710@gmail.com	Done
12	39	Second iteration of process flow diagram	Denis Sadovoy	Done
13	35	Create functional testing documentation	Denis Sadovoy	Done

Figure 5.16 – Azure DevOps sprint 3 tasks

5.5.1.4. Sprint 4

All activities for sprint 4 can be seen in Figure 5.17. Further iterations to the application development continued based on the final modifications to the hi-fi wireframes. Additional security and authentication features were added to the application, along with modifications to the database tables. Various front-end modifications were added as final touches to the application, such as modifying lecturer names, module descriptions and resource links. The prototype and user-centred design section (section 3) of the report was completed by the end of this sprint.



Order	ID	Title	Assigned To	State
1	87	Modifications of routes and templates based on UI and UX enhancements	Romi Dhillon	Done
2	59	Write prototype and user-centred design section (section 3) in the report	Romi Dhillon	Done
3	54	Addition of additional database tables through models.py based on improvements in UX design/requirements.	Romi Dhillon	Done
4	47	Add security and authenticity features to the application for sign-up and sign-in	Romi Dhillon	Done
5	56	Add module names, module descriptions, and module assessment to excel sheet	Ali Gündüz	Done
6	52	Adding GDPR considerations to ethics documentation	Ali Gündüz	Done
7	41	gather the resources for all individual modules REPL links	Ali Gündüz	Done
8	58	Add resource links (REPL and Slack) to application resource templates for each module.	rajin.lalgee@hotmail.com	Done
9	57	Add assessment type information to the application module templates	rajin.lalgee@hotmail.com	Done
10	55	Add lecturer names and descriptions to each module ratings template	rajin.lalgee@hotmail.com	Done
11	51	Final modifications to HiFi wireframes	sherrylu710@gmail.com	Done

Figure 5.17 – Azure DevOps sprint 4 tasks

5.5.1.5. Sprint 5

All activities for sprint 5 can be seen in Figure 5.18. The functional testing including usability and user-interface testing were conducted as part of this sprint. Based on the tests, error handling was prioritised, and features were fixed inside the application; features such as forgot password functionality and data storage when leaving the degree planner page. Package and library research captured in the previous sprint was also modified to ensure that it is more succinct. A final assessment of the user interface design was completed as part of this sprint; and section four of the final report was also completed.

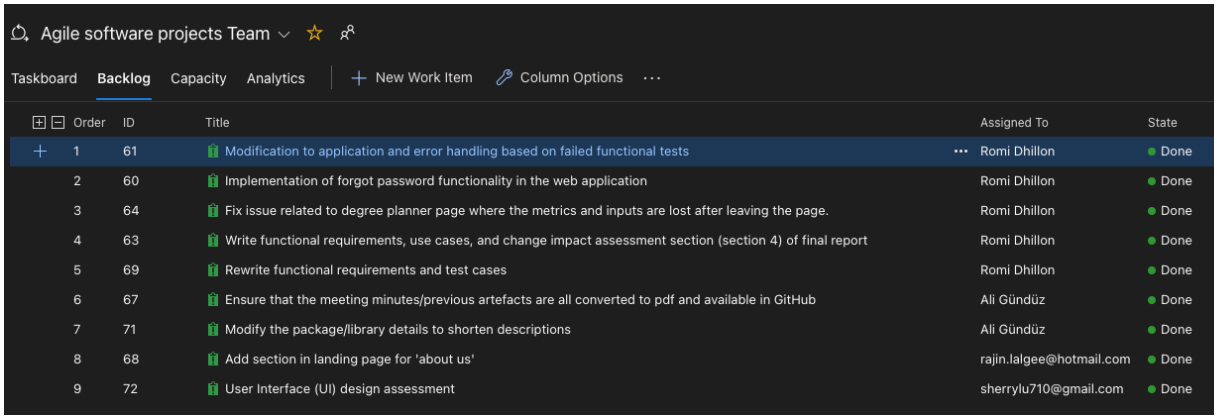


Figure 5.18 – Azure DevOps sprint 5 tasks

5.5.1.6. Sprint 6

All activities for sprint 6 can be seen in Figure 5.19. Functional testing on the failed test cases was performed once more, which led to successful tests for all cases. Minor modifications were made to the front-end features of the application (such as disclaimer messages) and to the wireframes (based on successful functional testing). The application design, development, architecture, and process design were documented in section five of the final report by the end of this sprint.

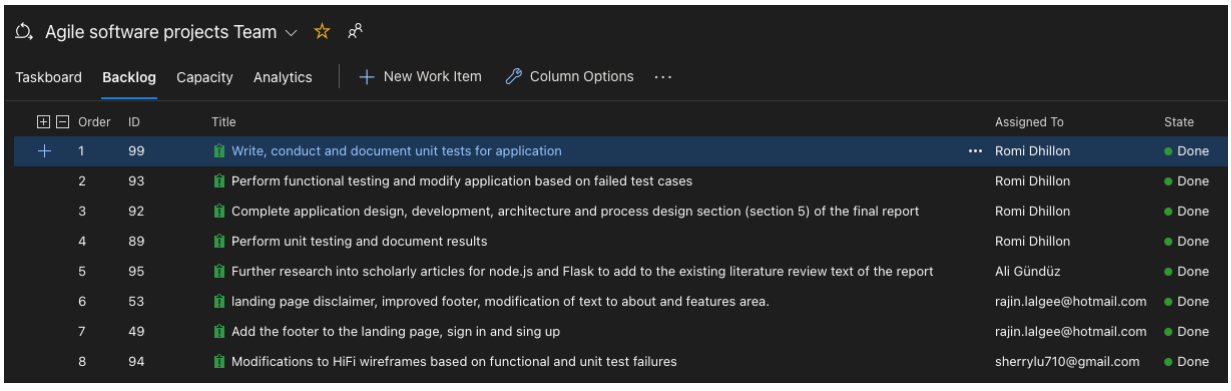


Figure 5.19 – Azure DevOps sprint 6 tasks

5.5.1.7. Sprint 7

Due to the iterative development throughout the project and the heavy load of tasks between sprints 1 and 6, most activities had been completed by sprint 7. The focus on this sprint was to write sections associated with testing and the conclusion within the report (sections 6 and 7). This left ample time for the team to review the report and suggest any modifications.

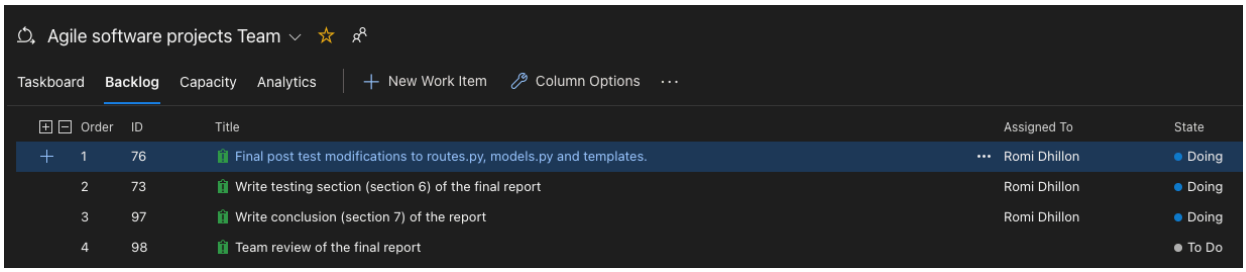


Figure 5.20 – Azure DevOps sprint 7 tasks

5.5.2. Kanban Boards

During each sprint meeting held on Sundays, the Kanban board was used to prioritise and track tasks. An example of the Kanban board during sprint 4 is shown in Figure 4.21. When conducting the sprint 3 meeting, activities for the next sprint are decided and captured on the Kanban board. New items are added to the board and a team member is assigned the activity. The state of the activity is monitored during the week where sprint 4 activities are being conducted. Team members can access the Kanban board and change the state of their task to 'To Do', 'Doing' or 'Done'. The item then moves to the appropriate swim lane. All activities were then reviewed in the sprint 4 meeting. Any activities not completed fell into the backlog.

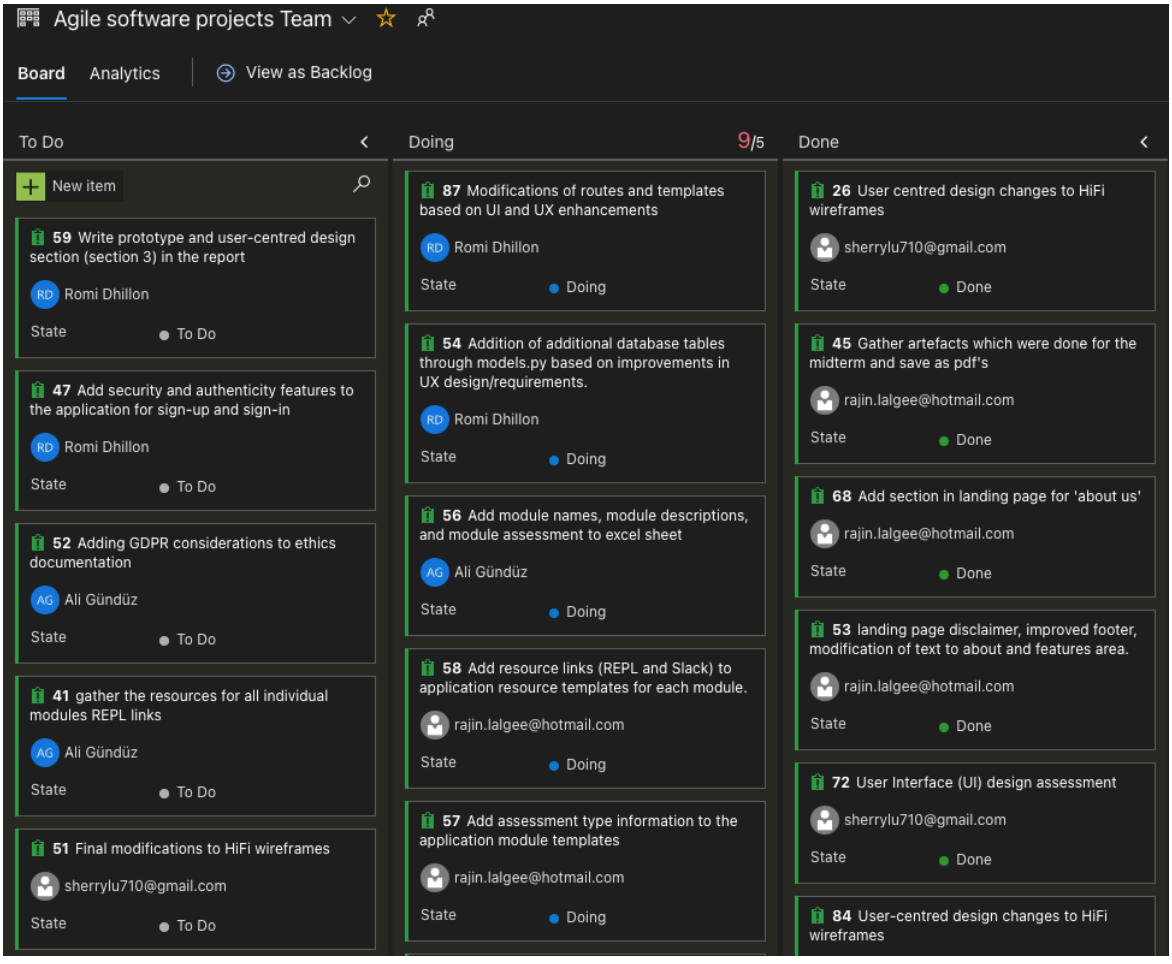


Figure 5.21 – Azure DevOps Kanban board

5.5.3. Backlog

Any activities that are in a state of ‘To Do’ or ‘Doing’ are in the backlog of work. This is independent of sprints and so activities that still need to be completed across different sprints can be found in the backlog. Not all activities that were scheduled within sprints were completed within the sprint. Every effort was made to clear the backlog items as part of future sprints, and one such snapshot of the backlog during the project can be found in Figure 4.22. The snapshot of the backlog is taken during sprint 7, which includes activities that need to be completed during sprint 7, but also included activities that were carried over to sprint 7 from sprint 6. Examples of these activities are the completion of documenting unit tests, final modifications to the wireframes, and aesthetic changes to the footer on the sign-in and sign-up pages.

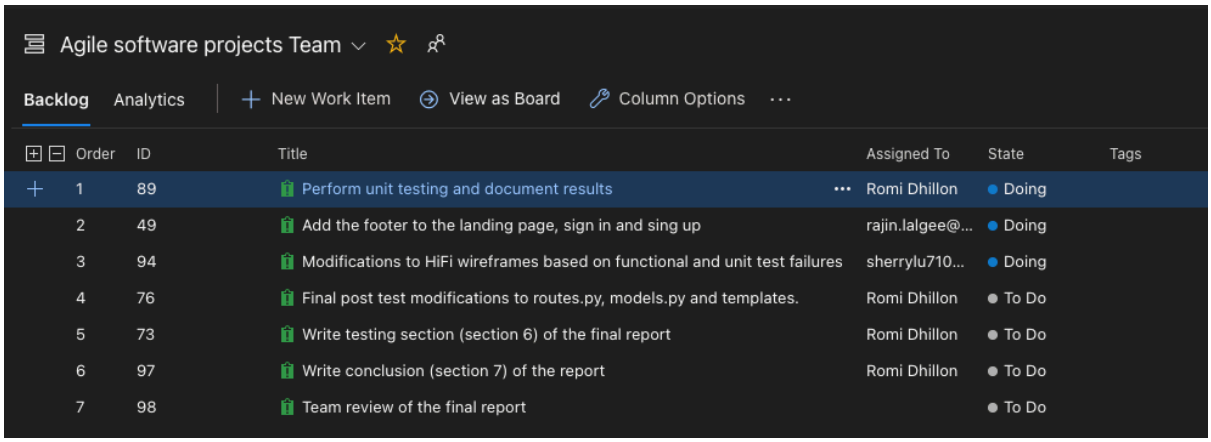


Figure 5.22 – A snapshot of the backlog at one stage in the project

5.6. Development using Git and GitHub

Git and GitHub were used in parallel with Microsoft Azure DevOps to deploy a mechanism for version control. The team used a repository named agilesoftwareprojects on GitHub to upload software and documentation throughout the duration of the project when completing activities for each sprint. Git has features such as creation of branches, pushing, pulling, merging, commit history, etc. Such features allow a collaborative approach whilst ensuring data integrity. An image of the master branch within the repository can be seen in Figure. 5.23, and at the time of taking the screenshot, there were 120 commits within the project. All the source code is found within the src folder within the repository, as seen in Figure 5.24 and 5.25, and all other documents and artefacts are stored in other folders or outside the src folder.

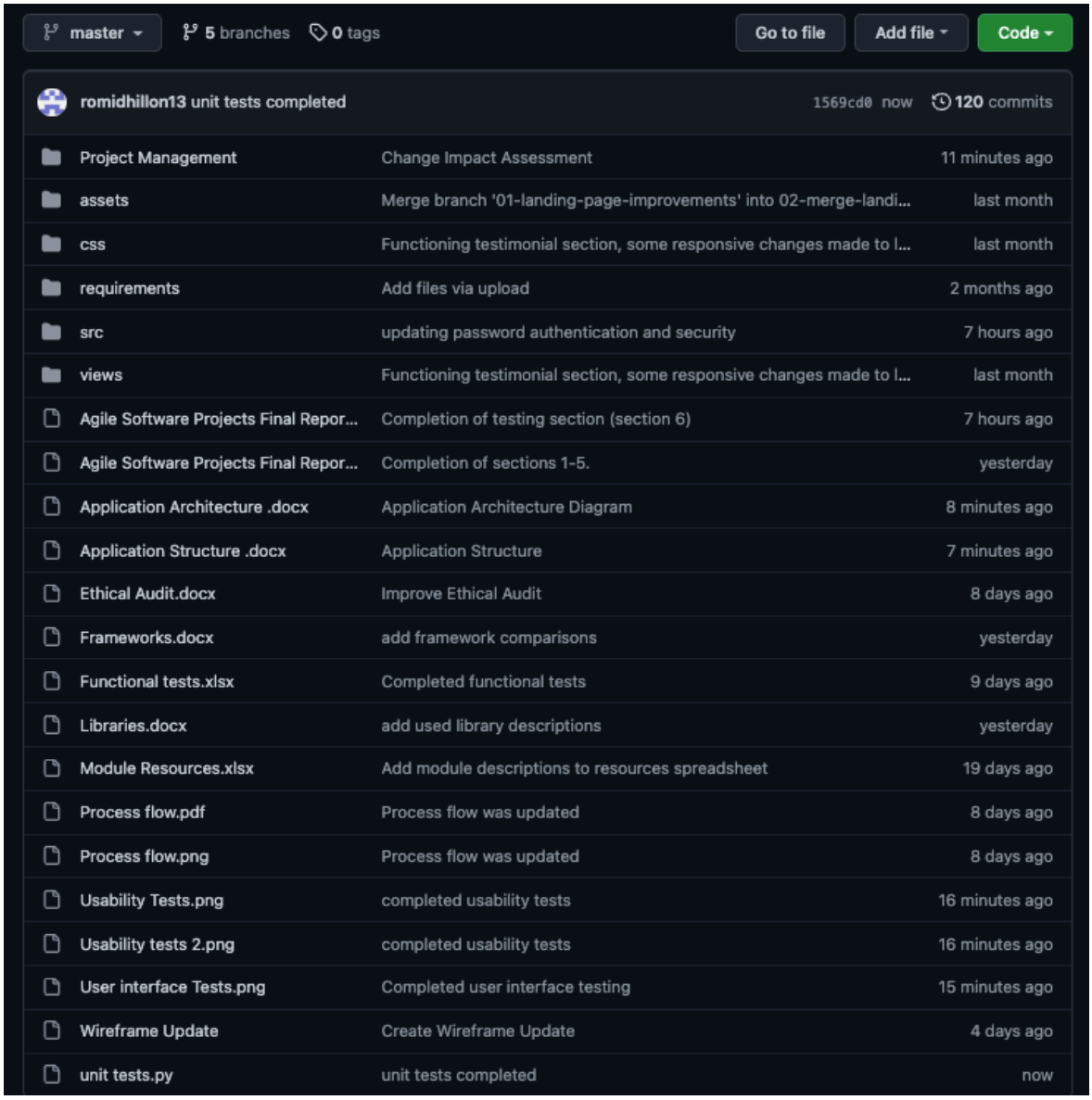


Figure 5.23 – GitHub repository master branch

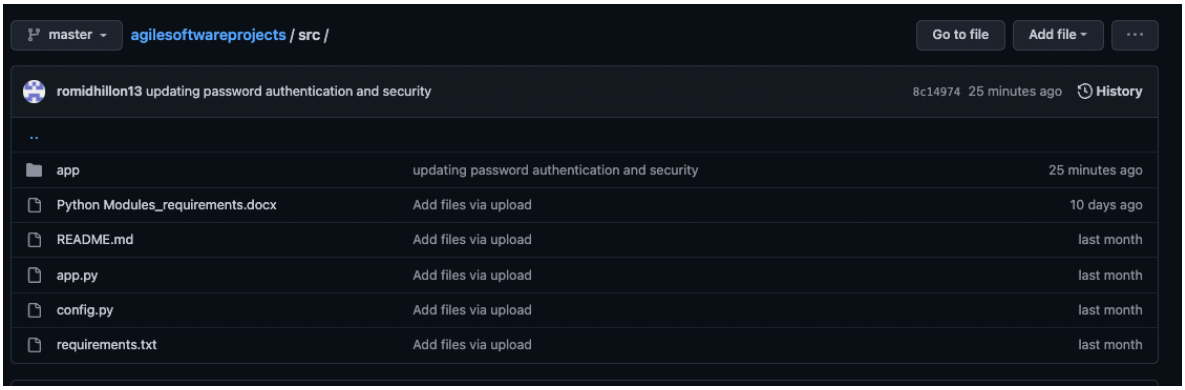


Figure 5.24 – GitHub repository files within the src folder

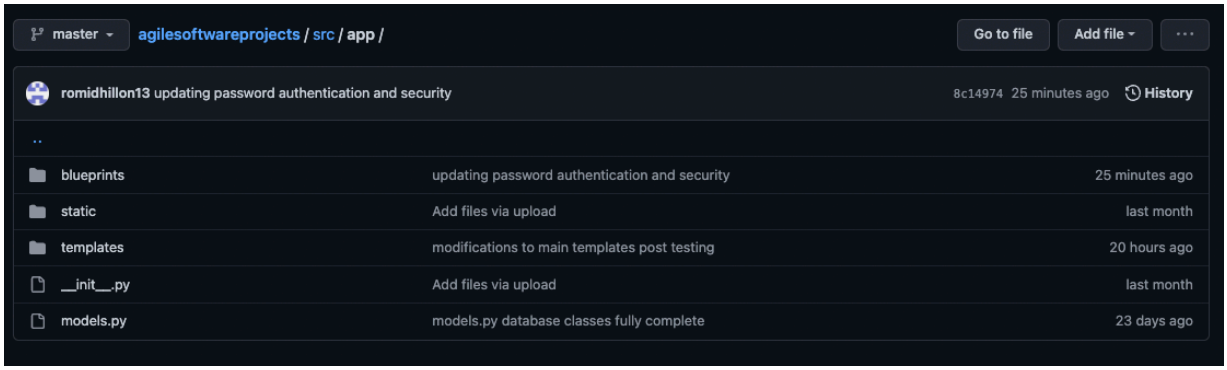


Figure 5.25 – GitHub repository files within the src/app folder

The full commit history is provided in Figure 5.26. Team members were encouraged to commit when there were considerable changes to previously committed versions rather than performing micro-commits. Even without micro-commits, there are a total of 120 commits at the time of writing this report, and each commit has a date and a hash number. The commit history is also provided as a separate file in the submission. A lot of original code development was conducted early on, and the commit descriptions for the developed code is named ‘create files via upload’ or ‘gitignore’. Although a more appropriate commit description should have been given for the commit, a detailed description is still available after clicking on the commit. Figure 5.26 is displayed as an A3 page, so you may need to slightly zoom in to see the image in detail. Please continue to the next page to view Figure 5.26.

Commits on Aug 30, 2022

unit tests completed	romidhillon13 committed 4 days ago	Verified	1569cd0
Application Structure	romidhillon13 committed 4 days ago	Verified	22c88bb
Application Architecture Diagram	romidhillon13 committed 4 days ago	Verified	17dce85
Change Impact Assessment	romidhillon13 committed 5 days ago	Verified	275fc8f
updated project plan with completed milestones	romidhillon13 committed 5 days ago	Verified	75b0ceb
Completed user interface testing	romidhillon13 committed 5 days ago	Verified	83406d2
completed usability tests	romidhillon13 committed 5 days ago	Verified	e341a27
Delete Screenshot 2022-08-30 at 00.56.36.png	romidhillon13 committed 5 days ago	Verified	a224df1
Completed usability tests (Azure DevOps)	romidhillon13 committed 6 days ago	Verified	699dcb4

Commits on Aug 29, 2022

updating password authentication and security	romidhillon13 committed 5 days ago	Verified	8c14974
addition of forgot password functionality	romidhillon13 committed 5 days ago	Verified	bc98437
Completion of testing section (section 6)	romidhillon13 committed 6 days ago	Verified	574c6d7
old report deleted	romidhillon13 committed 6 days ago	Verified	806ac11

Commits on Aug 28, 2022

modifications to models.py post testing	romidhillon13 committed 6 days ago	Verified	7678932
modifications to main templates post testing	romidhillon13 committed 6 days ago	Verified	bfbafb6
modifications to routes.py post functional tests	romidhillon13 committed 6 days ago	Verified	895cb85
modifications to routes.py post functional tests	romidhillon13 committed 6 days ago	Verified	10ae4a7
modifications to routes.py after functional tests	romidhillon13 committed 6 days ago	Verified	cdb7b9e
functional testing modifications/error handling	romidhillon13 committed 6 days ago	Verified	e977d73
Completion of sections 1-5.	romidhillon13 committed 6 days ago	Verified	7320871
add framework comparisons	gnuts committed 6 days ago		a276730
add used library descriptions	gnuts committed 6 days ago		b048d35

Commits on Aug 26, 2022

Create Wireframe Update	sherrii committed 8 days ago	Verified	c3871c4

Commits on Aug 24, 2022

modifications to improve degree planner page	romidhillon13 committed 10 days ago	Verified	58ee8fb
Final report with completion of section 1 and 2	romidhillon13 committed 10 days ago	Verified	f250b26

Commits on Aug 21, 2022

Process flow was updated	dsadovoy committed 13 days ago	Verified	cfc271f
Deleted duplicate of risk tracker	dsadovoy committed 13 days ago	Verified	3dcb862
Improve Ethical Audit	gnuts committed 13 days ago		cc1c055
Add user research artefact	gnuts committed 13 days ago		f3de5cc
Add market research artefact	gnuts committed 13 days ago		bf1445c
Add ideation artefacts	gnuts committed 13 days ago		b5d9c2c
Add MoM artefacts	gnuts committed 13 days ago		d2b8f25

Commits on Aug 20, 2022

Completed functional tests	dsadovoy committed 14 days ago	Verified	bf7a7e5

Commits on Aug 19, 2022

Add files via upload	sherrii committed 15 days ago	Verified	ee63176

Commits on Aug 15, 2022

Updated with tests results	dsadovoy committed 19 days ago	Verified	fe23a8e

Commits on Aug 14, 2022

Create unified ethics section with added repl copyright & user data d...	gnuts committed 20 days ago		9002de4

Commits on Aug 10, 2022

Add module descriptions to resources spreadsheet	gnuts committed 24 days ago		cc2c267
Add assessment type to module resources spreadsheet	gnuts committed 24 days ago		0a87441

Commits on Aug 8, 2022

Add professors to module resources spreadsheet	gnuts committed 26 days ago		68fe67c
Add shared google drive artefact	gnuts committed 26 days ago		cdec060
Add lo-fi artefacts	gnuts committed 26 days ago		959ccfd
Add survey artefacts	gnuts committed 26 days ago		d6ec4b4

Commits on Aug 7, 2022

Remove unnecessary artefacts	gnuts committed 27 days ago		31e5e9c
Add artefacts for personas, surveys and user journey	gnuts committed 27 days ago		6d737d4
Updated Functional tests	dsadovoy committed 27 days ago	Verified	f48c1df
reupload functional tests file	RajinL committed 27 days ago		7bd8122
reupload some files	RajinL committed 27 days ago		c390336

Commits on Aug 6, 2022

models.py database classes fully complete	romidhillon13 committed 28 days ago	Verified	17c117a
backend fully complete and front-end 95% complete	romidhillon13 committed 28 days ago	Verified	5632a83

Commits on Jul 29, 2022

Merge branch '01-landing-page-improvements' into 02-merge-landing-bra...	RajinL committed on 29 Jul		3019fa8

Commits on Jul 27, 2022

Add files via upload	romidhillon13 committed on 27 Jul	Verified	ca79818
Create gitignore	romidhillon13 committed on 27 Jul	Verified	7d97f7c
Add files via upload	romidhillon13 committed on 27 Jul	Verified	a50bbf4
Add files via upload	romidhillon13 committed on 27 Jul	Verified	ba56bad
Add files via upload	romidhillon13 committed on 27 Jul	Verified	2eb619f
Add files via upload	romidhillon13 committed on 27 Jul	Verified	a65f6a9
Create gitignore.mkd	romidhillon13 committed on 27 Jul	Verified	14fd0eb

Commits on Jul 26, 2022

Add files via upload	romidhillon13 committed on 26 Jul	Verified	784cf9e
Add files via upload	romidhillon13 committed on 26 Jul	Verified	c08b9da
Add files via upload	romidhillon13 committed on 26 Jul	Verified	c168078

Commits on Jul 25, 2022

Functioning testimonial section, some responsive changes made to landing	RajinL committed on 25 Jul		861abff

Commits on Jul 24, 2022

minor navbar and review section improvements	RajinL committed on 24 Jul		f840230
max-width navigation menu and hero adjustments	RajinL committed on 24 Jul		a4dbcc3
Added risk tracker	dsadovoy committed on 24 Jul	Verified	2dbdb26
create partials	gnuts committed on 24 Jul		3497cd2
fixes navigation	gnuts committed on 24 Jul		f3fb774

Commits on Jul 17, 2022

Add files via upload	romidhillon13 committed on 17 Jul	Verified	96df90d
Add files via upload	romidhillon13 committed on 17 Jul	Verified	7af9b7a
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	69080e7
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	e6e74cf

Commits on Jul 17, 2022

Delete gitignore	romidhillon13 committed on 17 Jul	Verified	82d7739
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	410b5f2
style.css	romidhillon13 committed on 17 Jul	Verified	26dc98f
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	3535c72
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	1639064
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	696b907
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	d88b431
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	99bd37d
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	76e00b0
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	6e23863
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	48a0bc2
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	1541f92
Delete gitignore	romidhillon13 committed on 17 Jul	Verified	1af5b45
templates added to templates folder	romidhillon13 committed on 17 Jul	Verified	4effcc1
form.html	romidhillon13 committed on 17 Jul	Verified	9188880
Create gitignore	romidhillon13 committed on 17 Jul	Verified	9bdfd00
Create gitignore	romidhillon13 committed on 17 Jul	Verified	d988d97
base.html	romidhillon13 committed on 17 Jul	Verified	81a735f
icons added to assets folder	romidhillon13 committed on 17 Jul	Verified	8eda0b2
Create gitignore	romidhillon13 committed on 17 Jul	Verified	39101e3
images added to assets folder	romidhillon13 committed on 17 Jul	Verified	ddc77b4
Added basic styles and user model, as well as sign up and sign in pag...	romidhillon13 committed on 17 Jul	Verified	cb5e6d9
__init__.py and routes.py	romidhillon13 committed on 17 Jul	Verified	9e46872
Added basic styles and user model, as well as sign up and sign in pag...	romidhillon13 committed on 17 Jul	Verified	cb5e6d9
__init__.py and routes.py	romidhillon13 committed on 17 Jul	Verified	9e46872
Create gitignore	romidhillon13 committed on 17 Jul	Verified	7fc36f3
Create gitignore	romidhillon13 committed on 17 Jul	Verified	8cb8a85
Create gitignore	romidhillon13 committed on 17 Jul	Verified	947cf1b
Create gitignore	romidhillon13 committed on 17 Jul	Verified	d39cc41
Create gitignore	romidhillon13 committed on 17 Jul	Verified	c5f6f60
Create gitignore	romidhillon13 committed on 17 Jul	Verified	0d91ac5
Create gitignore	romidhillon13 committed on 17 Jul	Verified	b9ccde2
Create gitignore	romidhillon13 committed on 17 Jul	Verified	f7b6b9c
Create gitignore	romidhillon13 committed on 17 Jul	Verified	a2cb69a
Create gitignore	romidhillon13 committed on 17 Jul	Verified	ffd42cd
Add files via upload	romidhillon13 committed on 17 Jul	Verified	a95ff46
Create gitignore	romidhillon13 committed on 17 Jul	Verified	eace78b

Figure 5.26 – GitHub commit history (continued on the next page)

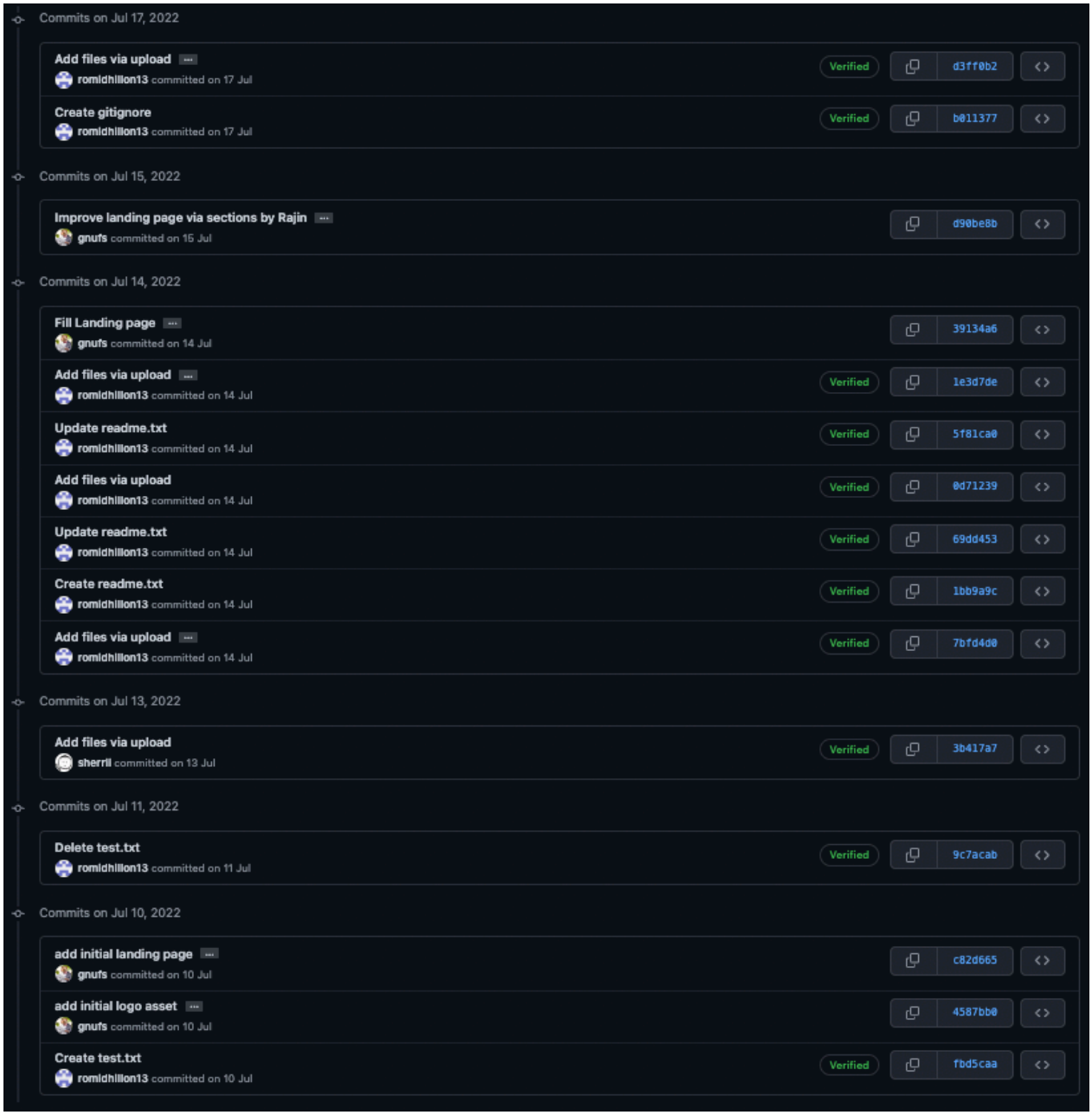


Figure 5.26 – GitHub commit history

6. Testing

6.1.Functional Testing

The purpose of functional testing is to verify that each function of the application performs correctly in the way it was design to perform. An input is provided, and a check is performed to assess if the expected output is achieved. There are many types of functional tests that can be performed, but the main tests conducted as part of this study are user interface testing and usability testing.

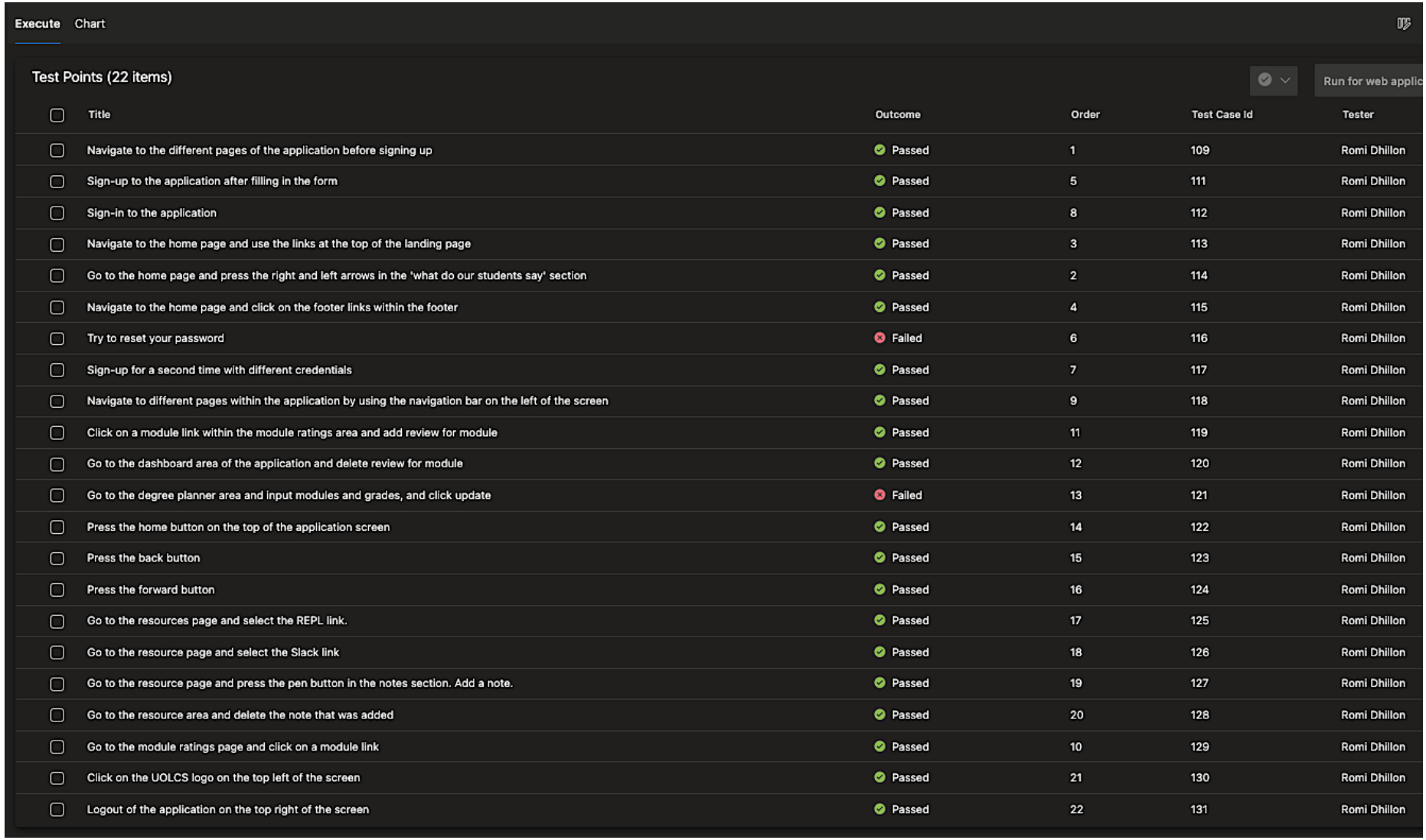
6.2.Justification of testing methods

Usability testing is a black box testing technique, as users have no knowledge of its internal workings. It is possible to capture all expected and unexpected user actions, and so an end-to-end test of all subsystems, database actions, web servers and integrated systems is invertedly conducted. If a user conducts an action that leads to an error, the application will need to be amended, and so such a testing technique does not focus on testing all individual isolated parts (like unit tests) but rather only focuses on the areas that need to be fixed.

Although usability testing can highlight errors associated with the functioning of the application, it does not emphasise the aesthetic features that may repel users. By conducting user interface testing, an opportunity is provided to amend undesirable visual aspects.

6.3.Usability Testing

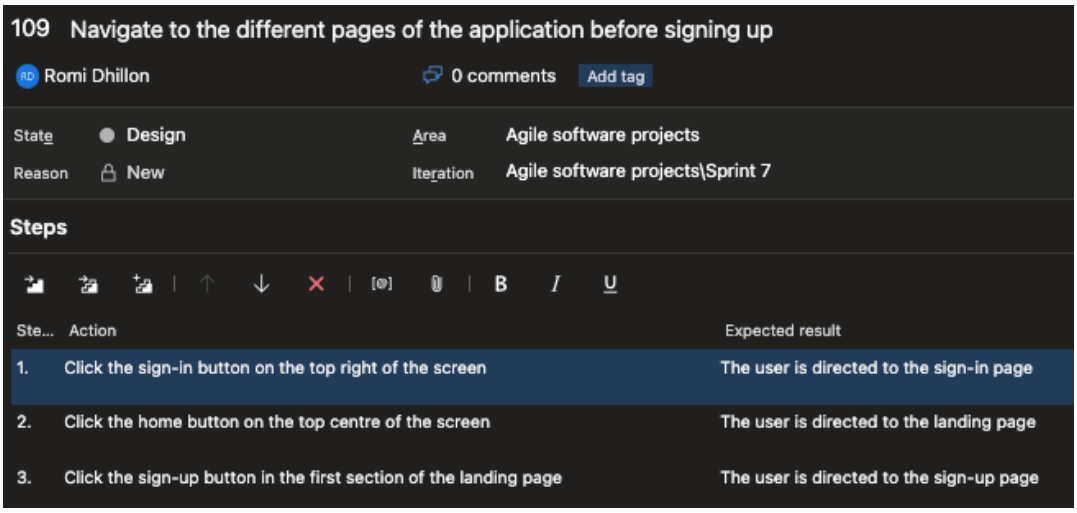
Five individuals with no vested interests in the project were chosen by the team leader from his workplace to test the usability of the application. Microsoft Azure Test Plans was used to write the test cases and a total of 22 test cases were written, each with their own test case id, order number, outcome, tester. The summary of the results is shown in Figure 6.1, which shows that 2 test cases failed out of the 22 that were tested by the application developer.



Title	Outcome	Order	Test Case Id	Tester
Navigate to the different pages of the application before signing up	Passed	1	109	Romi Dhillon
Sign-up to the application after filling in the form	Passed	5	111	Romi Dhillon
Sign-in to the application	Passed	8	112	Romi Dhillon
Navigate to the home page and use the links at the top of the landing page	Passed	3	113	Romi Dhillon
Go to the home page and press the right and left arrows in the 'what do our students say' section	Passed	2	114	Romi Dhillon
Navigate to the home page and click on the footer links within the footer	Passed	4	115	Romi Dhillon
Try to reset your password	Failed	6	116	Romi Dhillon
Sign-up for a second time with different credentials	Passed	7	117	Romi Dhillon
Navigate to different pages within the application by using the navigation bar on the left of the screen	Passed	9	118	Romi Dhillon
Click on a module link within the module ratings area and add review for module	Passed	11	119	Romi Dhillon
Go to the dashboard area of the application and delete review for module	Passed	12	120	Romi Dhillon
Go to the degree planner area and input modules and grades, and click update	Failed	13	121	Romi Dhillon
Press the home button on the top of the application screen	Passed	14	122	Romi Dhillon
Press the back button	Passed	15	123	Romi Dhillon
Press the forward button	Passed	16	124	Romi Dhillon
Go to the resources page and select the REPL link.	Passed	17	125	Romi Dhillon
Go to the resource page and select the Slack link	Passed	18	126	Romi Dhillon
Go to the resource page and press the pen button in the notes section. Add a note.	Passed	19	127	Romi Dhillon
Go to the resource area and delete the note that was added	Passed	20	128	Romi Dhillon
Go to the module ratings page and click on a module link	Passed	10	129	Romi Dhillon
Click on the UOLCS logo on the top left of the screen	Passed	21	130	Romi Dhillon
Logout of the application on the top right of the screen	Passed	22	131	Romi Dhillon

Figure 6.1 – Twenty-two usability test cases captured in Azure DevOps

Each test case shown in Figure 6.1 had a list of steps which needed to be conducted successfully by the test user for the test case to pass. Expected results were also defined for each step in each test case. Steps for all test cases are shown over the next 7 pages in Fig 6.2.



109 Navigate to the different pages of the application before signing up

Romi Dhillon | 0 comments | Add tag

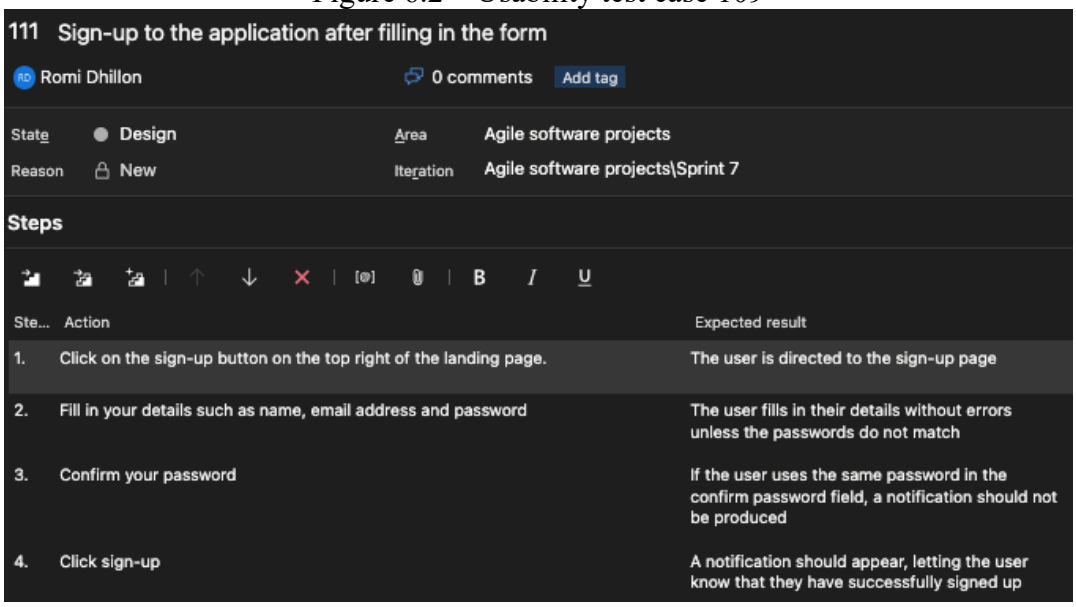
State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Click the sign-in button on the top right of the screen	The user is directed to the sign-in page
2.	Click the home button on the top centre of the screen	The user is directed to the landing page
3.	Click the sign-up button in the first section of the landing page	The user is directed to the sign-up page

Figure 6.2 – Usability test case 109



111 Sign-up to the application after filling in the form

Romi Dhillon | 0 comments | Add tag

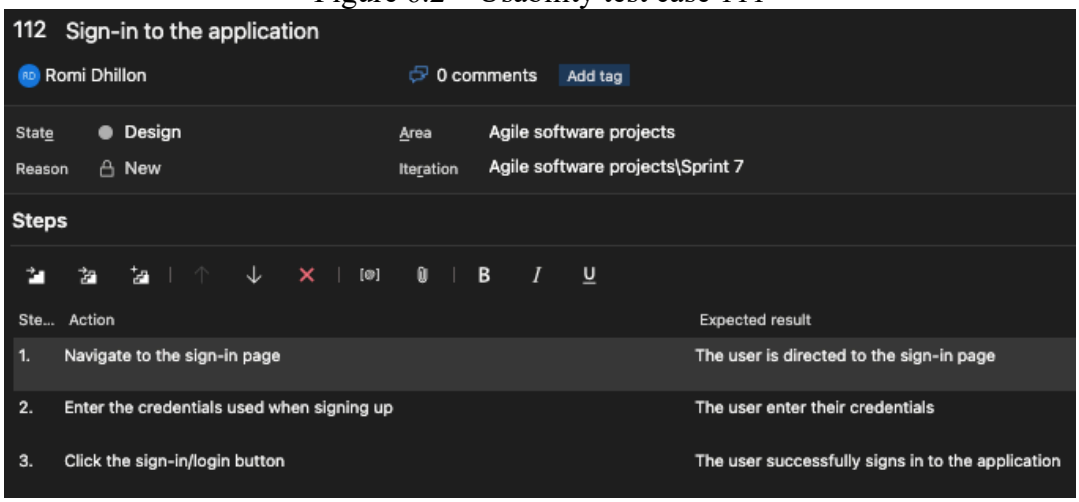
State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Click on the sign-up button on the top right of the landing page.	The user is directed to the sign-up page
2.	Fill in your details such as name, email address and password	The user fills in their details without errors unless the passwords do not match
3.	Confirm your password	If the user uses the same password in the confirm password field, a notification should not be produced
4.	Click sign-up	A notification should appear, letting the user know that they have successfully signed up

Figure 6.2 – Usability test case 111



112 Sign-in to the application

Romi Dhillon | 0 comments | Add tag

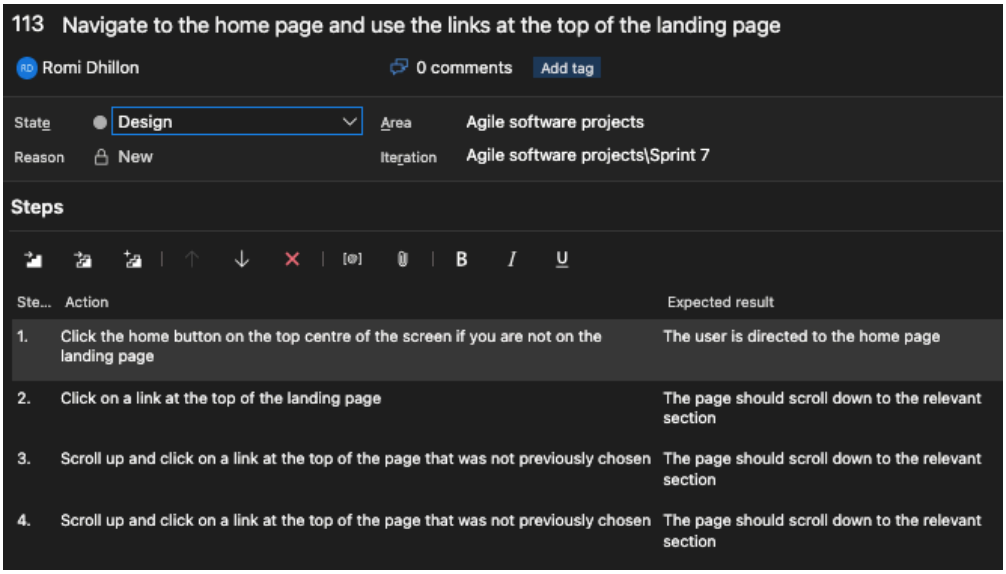
State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Navigate to the sign-in page	The user is directed to the sign-in page
2.	Enter the credentials used when signing up	The user enter their credentials
3.	Click the sign-in/login button	The user successfully signs in to the application

Figure 6.2 – Usability test case 112



113 Navigate to the home page and use the links at the top of the landing page

Romi Dhillon | 0 comments | Add tag

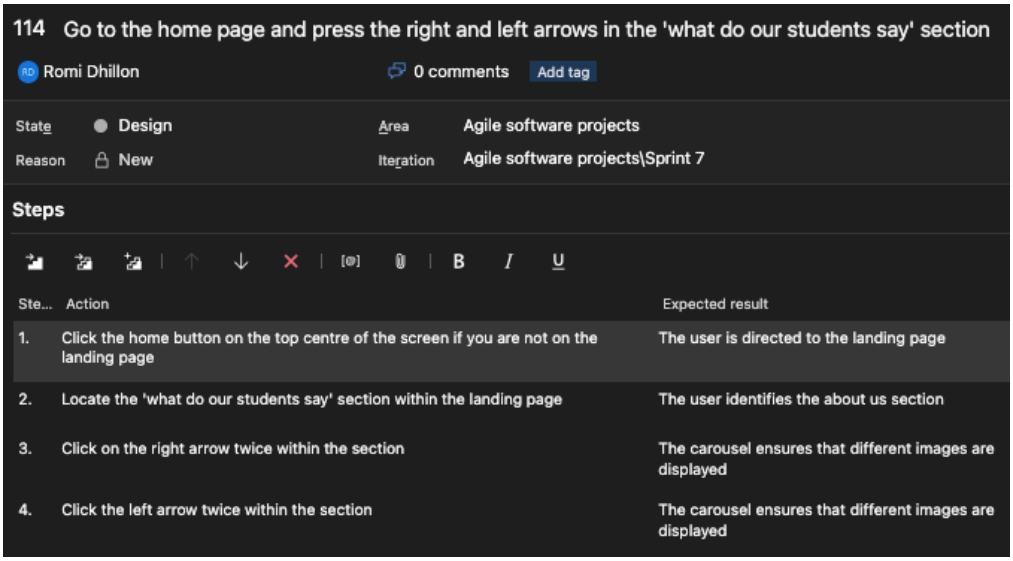
State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Click the home button on the top centre of the screen if you are not on the landing page	The user is directed to the home page
2.	Click on a link at the top of the landing page	The page should scroll down to the relevant section
3.	Scroll up and click on a link at the top of the page that was not previously chosen	The page should scroll down to the relevant section
4.	Scroll up and click on a link at the top of the page that was not previously chosen	The page should scroll down to the relevant section

Figure 6.2 – Usability test case 113



114 Go to the home page and press the right and left arrows in the 'what do our students say' section

Romi Dhillon | 0 comments | Add tag

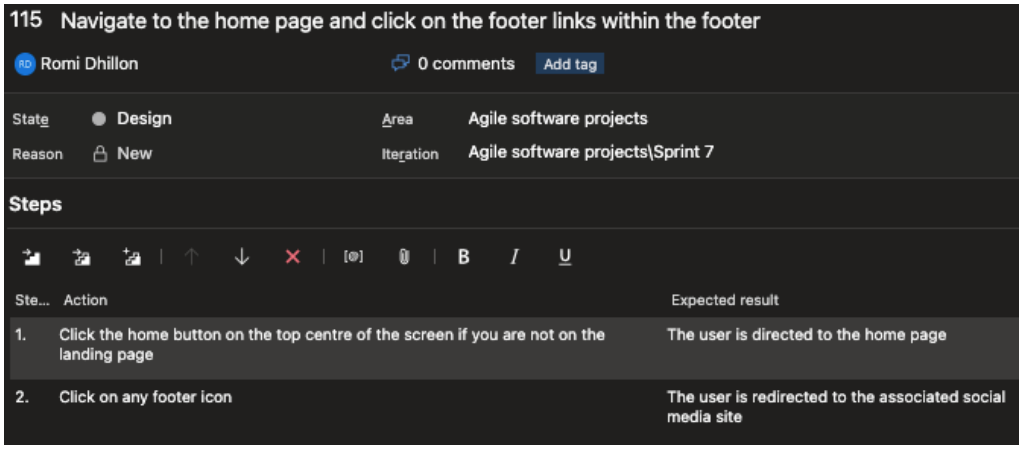
State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Click the home button on the top centre of the screen if you are not on the landing page	The user is directed to the landing page
2.	Locate the 'what do our students say' section within the landing page	The user identifies the about us section
3.	Click on the right arrow twice within the section	The carousel ensures that different images are displayed
4.	Click the left arrow twice within the section	The carousel ensures that different images are displayed

Figure 6.2 – Usability test case 114



115 Navigate to the home page and click on the footer links within the footer

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Click the home button on the top centre of the screen if you are not on the landing page	The user is directed to the home page
2.	Click on any footer icon	The user is redirected to the associated social media site

Figure 6.2 – Usability test case 114

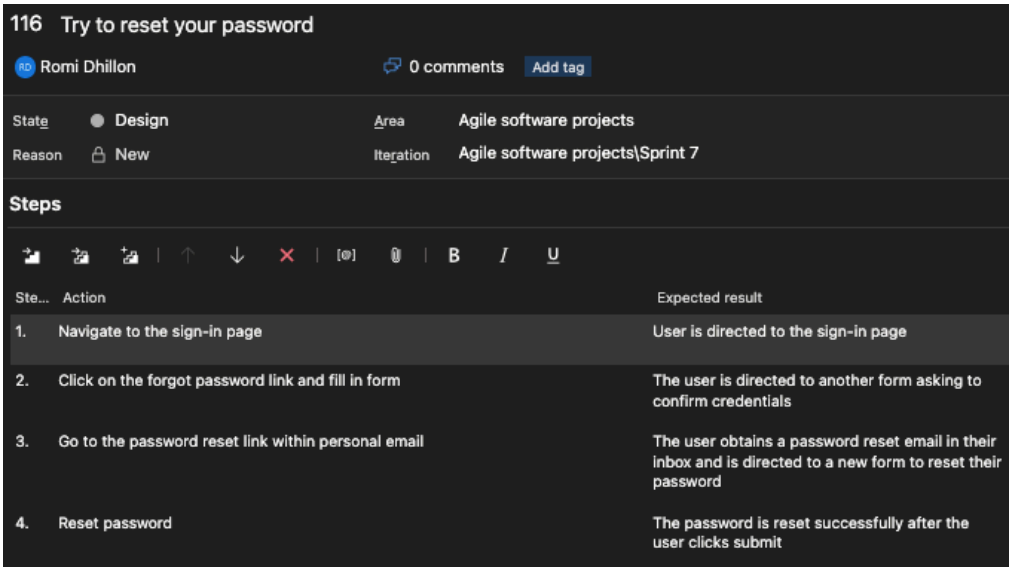


Figure 6.2 – Usability test case 116

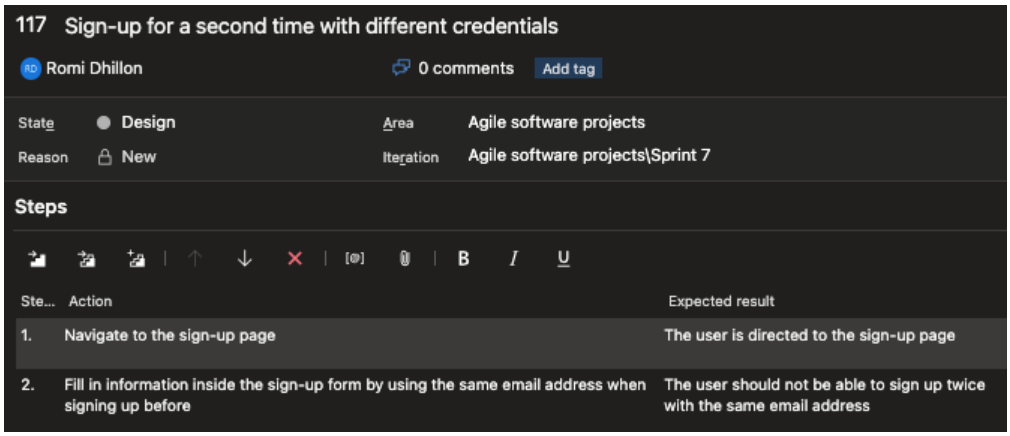


Figure 6.2 – Usability test case 117

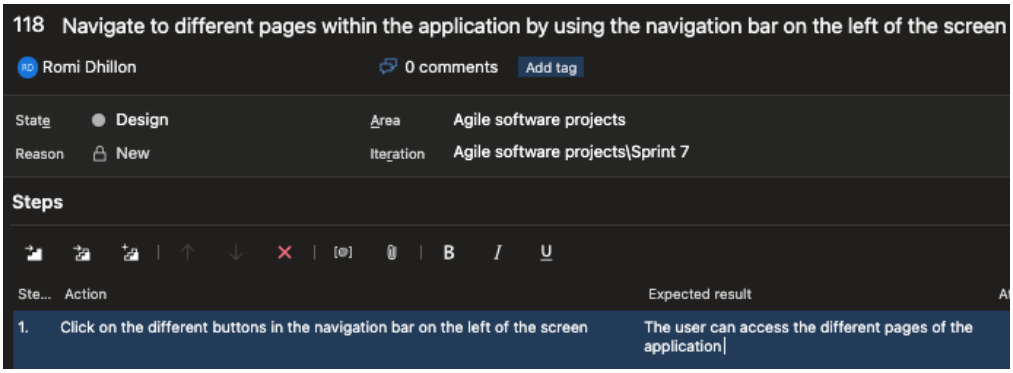
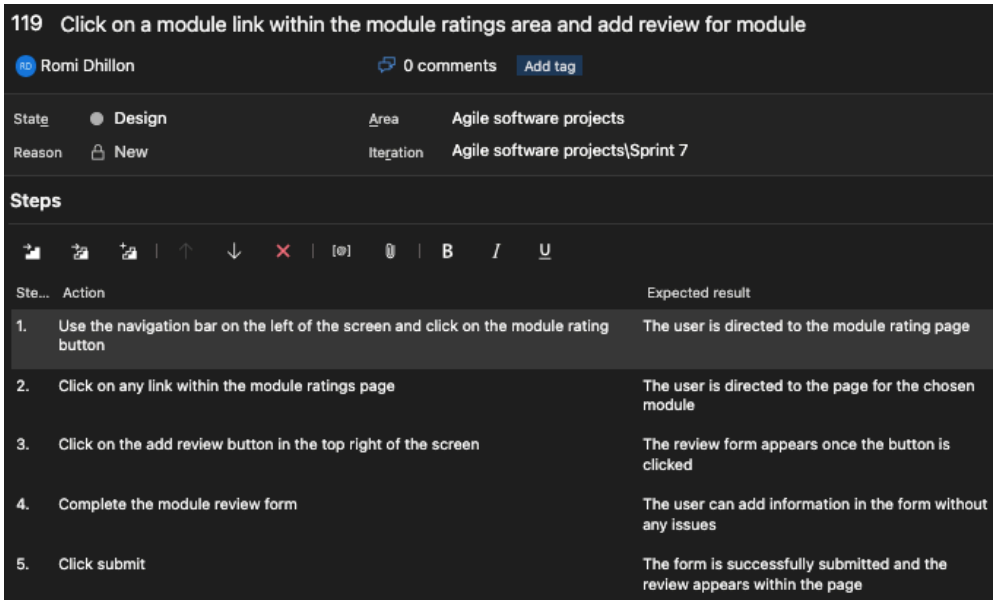


Figure 6.2 – Usability test case 118



119 Click on a module link within the module ratings area and add review for module

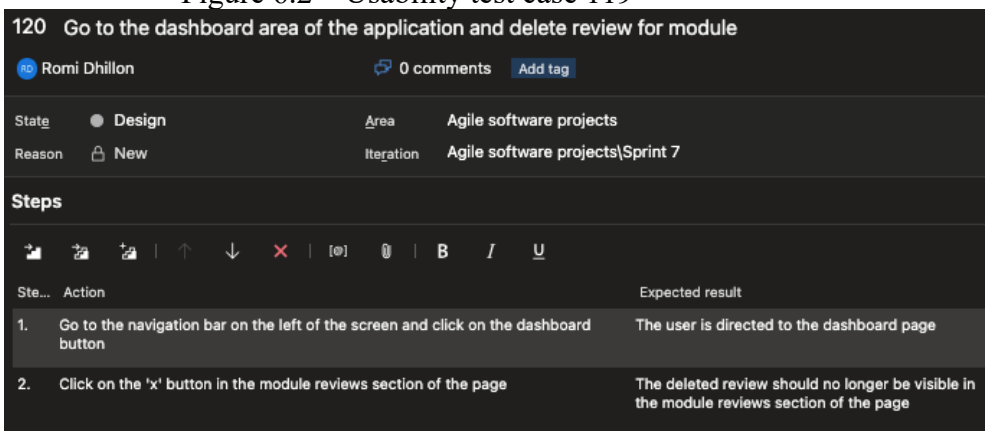
Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects
Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Use the navigation bar on the left of the screen and click on the module rating button	The user is directed to the module rating page
2.	Click on any link within the module ratings page	The user is directed to the page for the chosen module
3.	Click on the add review button in the top right of the screen	The review form appears once the button is clicked
4.	Complete the module review form	The user can add information in the form without any issues
5.	Click submit	The form is successfully submitted and the review appears within the page

Figure 6.2 – Usability test case 119



120 Go to the dashboard area of the application and delete review for module

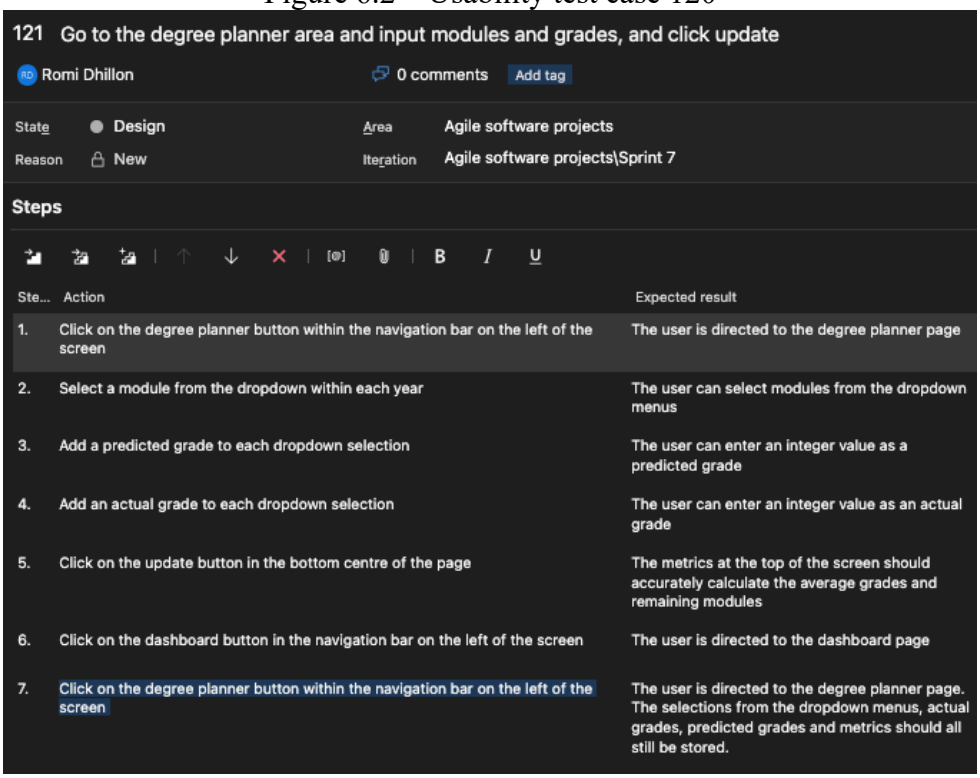
Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects
Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Go to the navigation bar on the left of the screen and click on the dashboard button	The user is directed to the dashboard page
2.	Click on the 'x' button in the module reviews section of the page	The deleted review should no longer be visible in the module reviews section of the page

Figure 6.2 – Usability test case 120



121 Go to the degree planner area and input modules and grades, and click update

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects
Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Click on the degree planner button within the navigation bar on the left of the screen	The user is directed to the degree planner page
2.	Select a module from the dropdown within each year	The user can select modules from the dropdown menus
3.	Add a predicted grade to each dropdown selection	The user can enter an integer value as a predicted grade
4.	Add an actual grade to each dropdown selection	The user can enter an integer value as an actual grade
5.	Click on the update button in the bottom centre of the page	The metrics at the top of the screen should accurately calculate the average grades and remaining modules
6.	Click on the dashboard button in the navigation bar on the left of the screen	The user is directed to the dashboard page
7.	Click on the degree planner button within the navigation bar on the left of the screen	The user is directed to the degree planner page. The selections from the dropdown menus, actual grades, predicted grades and metrics should all still be stored.

Figure 6.2 – Usability test case 121

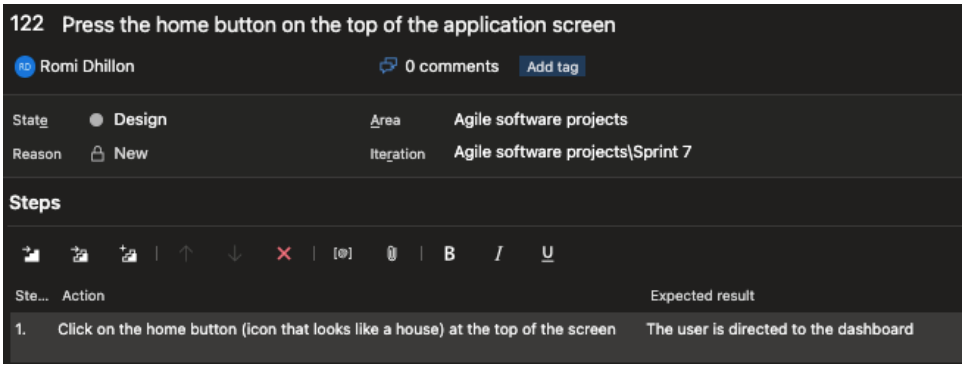


Figure 6.2 – Usability test case 122

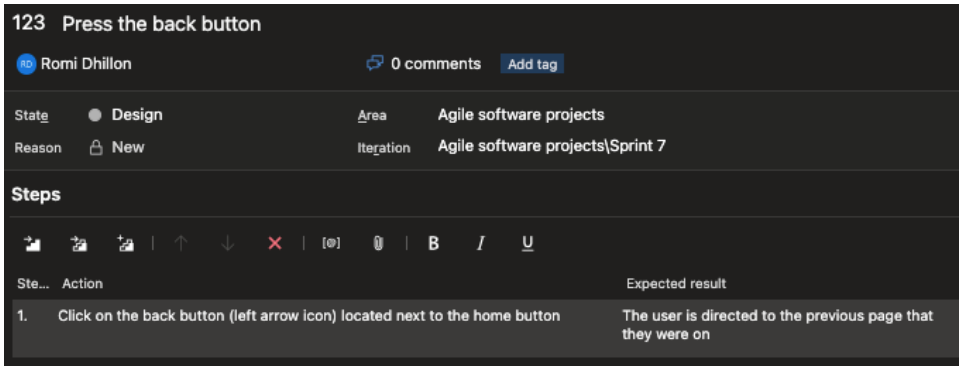


Figure 6.2 – Usability test case 123

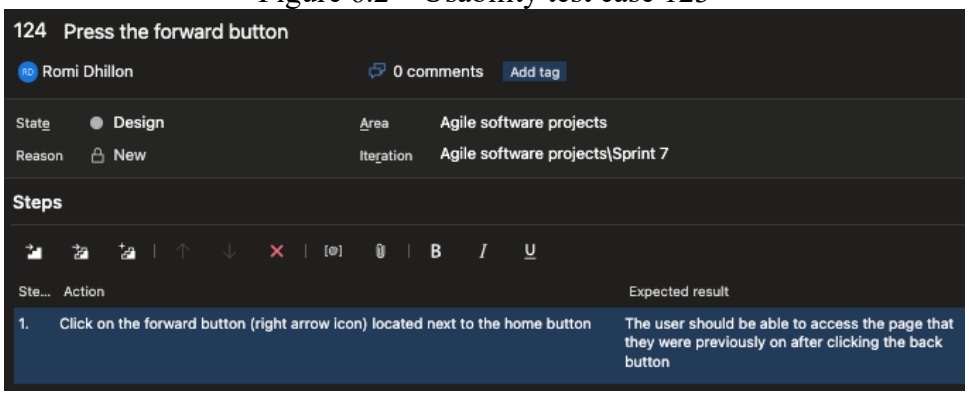


Figure 6.2 – Usability test case 124

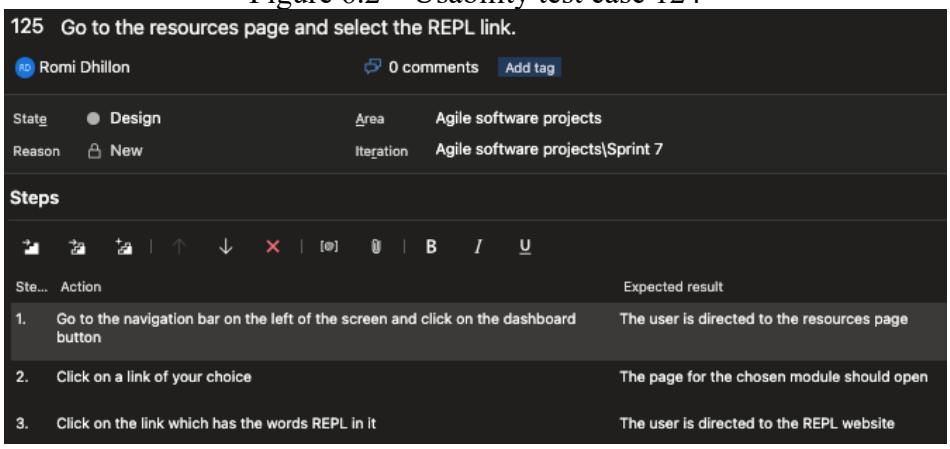


Figure 6.2 – Usability test case 125

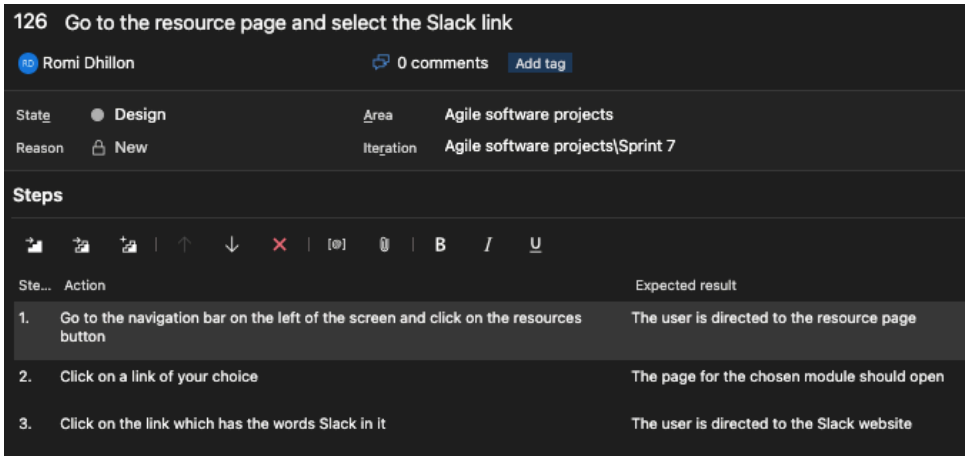


Figure 6.2 – Usability test case 126

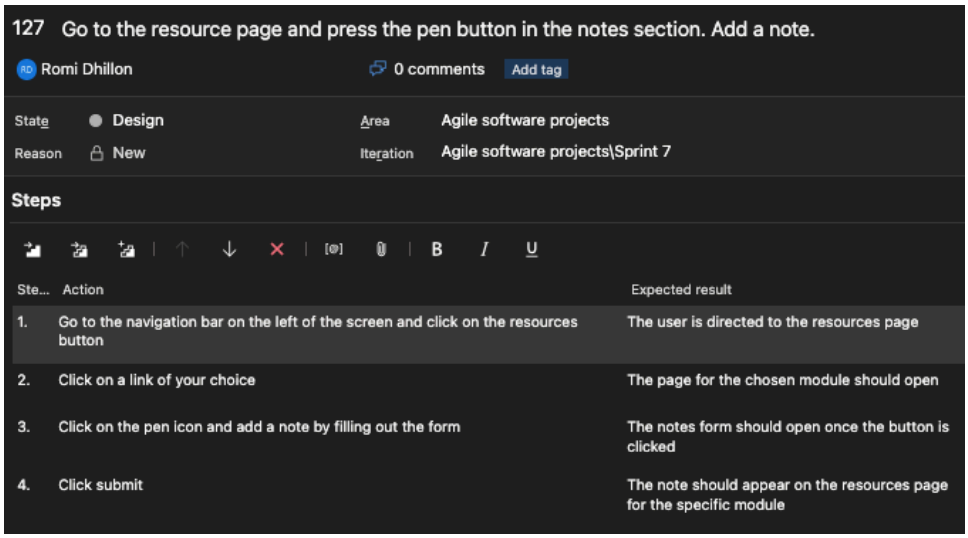


Figure 6.2 – Usability test case 127

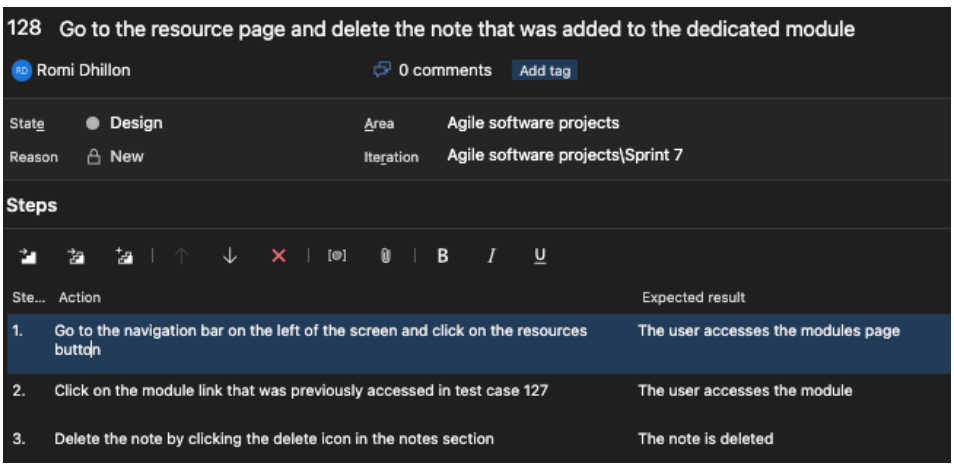


Figure 6.2 – Usability test case 128

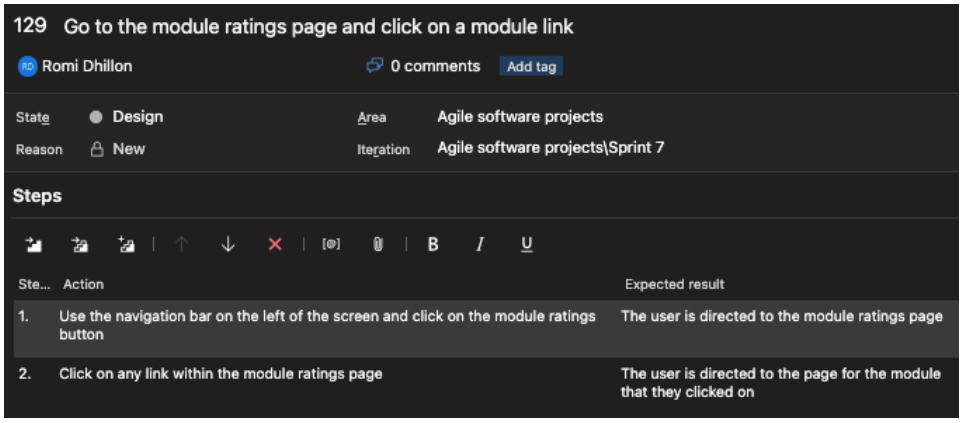


Figure 6.2 – Usability test case 129

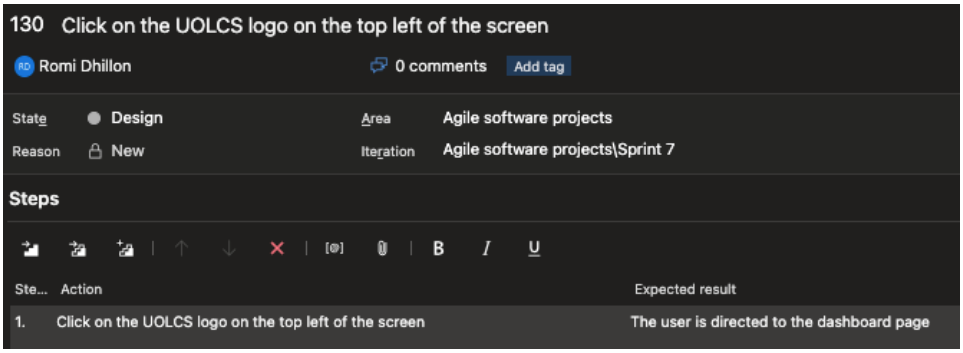


Figure 6.2 – Usability test case 130

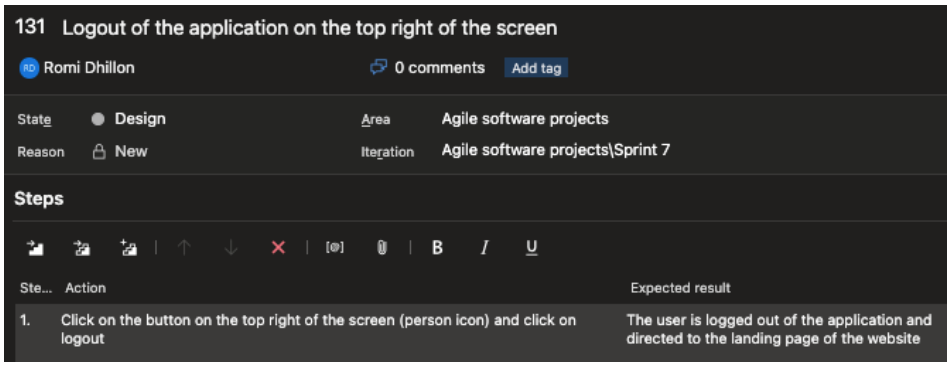


Figure 6.2 – Usability test case 131

6.3.1. Modifying the code to fix failed usability tests

Test cases 116 and 121 failed testing in the first round of testing. Modifications were made to the application in order to fix these usability issues. Test case 116 failed after the user clicked on the forgot password link and did not receive an email to reset their password. Microsoft Outlook blocked the connection due to their security rules. The issue was resolved by purchasing a professional business email address and obtaining email server SMPT credentials

under a 1-year subscription. Once the code was updated with the new information, test case 116 was repeated by the test user and the test case passed.

For test cases 121, the test user navigated to the degree planner page from the navigation bar. They selected the module choices from the dropdown menus under each year and inputted a manual and predicted grade. The metrics were calculated successfully, however, the test case failed as the information was not stored once the user left the page and entered the page again.

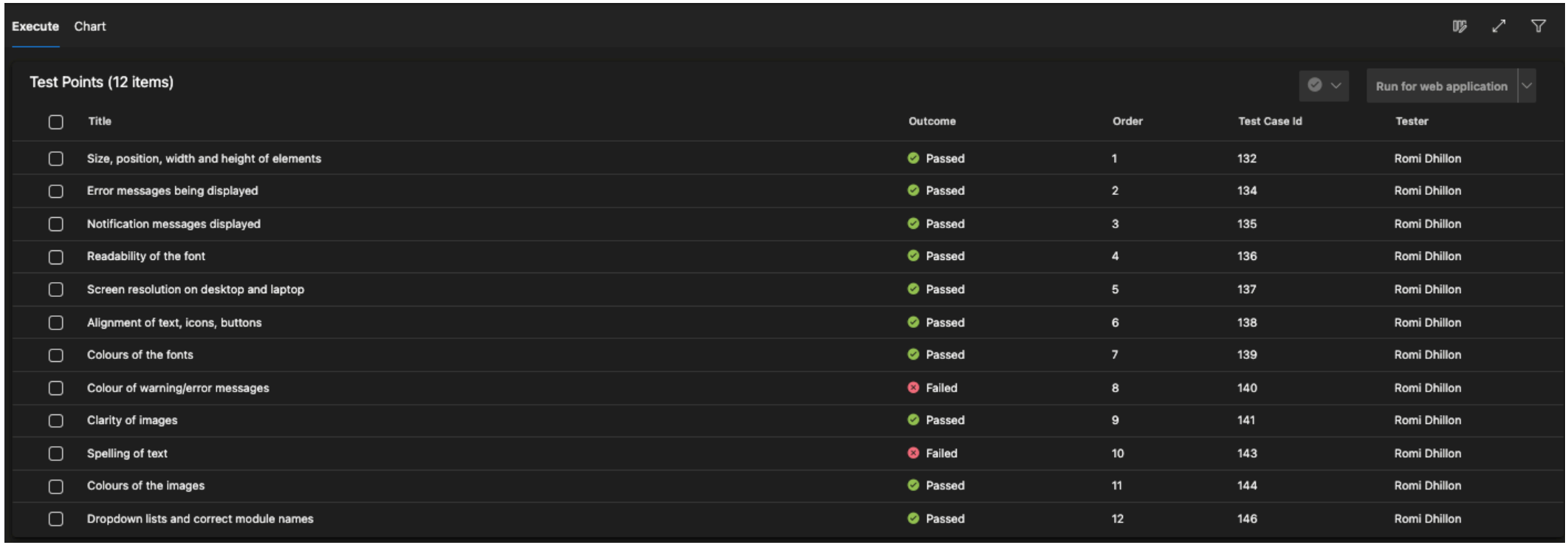
In order to obtain a successful outcome for test case 121, user module choices were saved via a new model, then loaded whenever the user navigated to the degree planner page. A new model named UserModule was created in the main models.py file which has foreign keys to the Module class. Backward relations were created within the Module class named usermodules to make it easier to access all the user's modules. The next step was to restructure the degree planner route within the routes.py file. Originally, the post request conducted all the calculations, and then rendered them on the degree planner page leading to a GET request being executed when leaving the page. This led to the data not being stored when the user navigated to other pages. The GET and POST requests for the degree planner route were modified so that the data was captured and stored on the page even if the user navigated to a different page.

6.4.User interface Testing

The purpose of user interface testing is to identify any errors or inconveniences in the application design when using the graphical user interface. This testing technique is mainly focused on the design of elements. Microsoft Azure DevOps was used to perform write and document the test cases and test results. A total of 12 test cases were written which are shown in Figure 6.3. The outcome of each test case, the test case id, and the name of the application tester are also shown. Test cases 140 and 143 both failed out of the twelve tests.

As aspects of design can be subjective and based on personal tastes, a greater number of test users were invited to conduct user interface testing. A total of ten professional work colleagues from industry were assigned the task of testing the application by the team leader, and none of the individuals had a vested interest in the application.

Each test case and steps are shown in Figure 6.4 over a span of four pages. Each test had a list of steps which needed to be conducted successfully by the test user for the test case to pass. An expected result was also defined for each step in each test case.



Title	Outcome	Order	Test Case Id	Tester
Size, position, width and height of elements	Passed	1	132	Romi Dhillon
Error messages being displayed	Passed	2	134	Romi Dhillon
Notification messages displayed	Passed	3	135	Romi Dhillon
Readability of the font	Passed	4	136	Romi Dhillon
Screen resolution on desktop and laptop	Passed	5	137	Romi Dhillon
Alignment of text, icons, buttons	Passed	6	138	Romi Dhillon
Colours of the fonts	Passed	7	139	Romi Dhillon
Colour of warning/error messages	Failed	8	140	Romi Dhillon
Clarity of images	Passed	9	141	Romi Dhillon
Spelling of text	Failed	10	143	Romi Dhillon
Colours of the images	Passed	11	144	Romi Dhillon
Dropdown lists and correct module names	Passed	12	146	Romi Dhillon

Figure 6.3 – Twelve user-interface test cases captured in Azure DevOps

132 Size, width, and height of elements

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects
Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Navigate to different pages of the application and assess the size, width, and height of the elements within the application	The user is content with the size, width and height of elements within different pages of the application

Figure 6.4 – User Interface test case 132

134 Error messages being displayed

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects
Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Go to the sign-in page of the application	The user is directed to the sign-in page of the application
2.	Enter your email address and a wrong password	An error message should appear at the top right of the page, stating that the wrong password has been entered and to try again. The user should find the message to be clear.

Figure 6.4 – User Interface test case 134

135 Notification messages displayed

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects
Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Navigate to the sign-in page and sign-in to the application	A notification message should appear on the top right of the screen stating that the user has successfully logged into the application
2.	Click the person icon at the top right of the screen and click logout	A notification message should appear on the top right of the screen stating that the user has successfully logged out of the applicationA

Figure 6.4 – User Interface test case 135

136 Readability of the font

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Read any text from the landing page	The user should be able to easily read the text
2.	Sign-in to the application, and navigate to any page within the application	The user should be able to navigate their chosen page
3.	Read any text from the chosen page	The user should be able to easily read the text within the application

Figure 6.4 – User Interface test case 136

137 Screen resolution on desktop and laptop

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Open the application on your desktop of laptop	The webpages should be displayed to the right resolution on desktops and laptops

Figure 6.4 – User Interface test case 137

138 Alignment of text, icons, buttons

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Review the text, icons and images on different pages of the application	The user should not feel that the images, icons or text is misaligned or uneasy to look at

Figure 6.4 – User Interface test case 138

139 Colours of the fonts

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Navigate to different pages of the application and give your opinion on the fonts seen	The user should not feel uneasy when viewing the fonts used within the application. The user should not be driven away by the choice of fonts.

Figure 6.4 – User Interface test case 139

140 Colour of warning/error messages

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Navigate to the sign-up page of the application	The user is directed to the sign-up page
2.	Fill in the details within the form and choose a different password in the confirm password field in comparison to the password field.	The user should give positive feedback on the choice of red font for the error messages

Figure 6.4 – User Interface test case 140

141 Clarity of images

Romi Dhillon | 0 comments | Add tag

State: Design | Area: Agile software projects

Reason: New | Iteration: Agile software projects\Sprint 7

Steps

Ste...	Action	Expected result
1.	Locate images by navigating to different areas of the website and application and review the clarity of the images.	The user should be able to view the images clearly

Figure 6.4 – User Interface test case 141

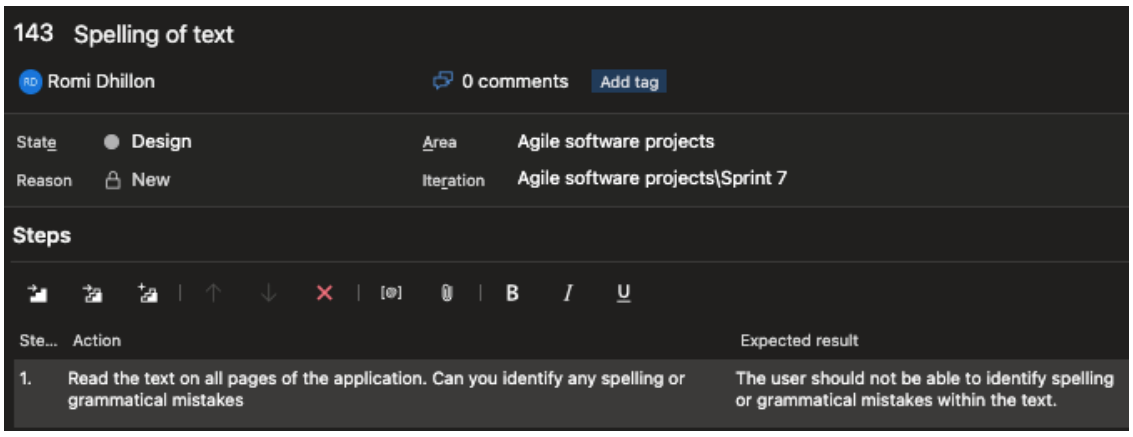


Figure 6.4 – User Interface test case 143

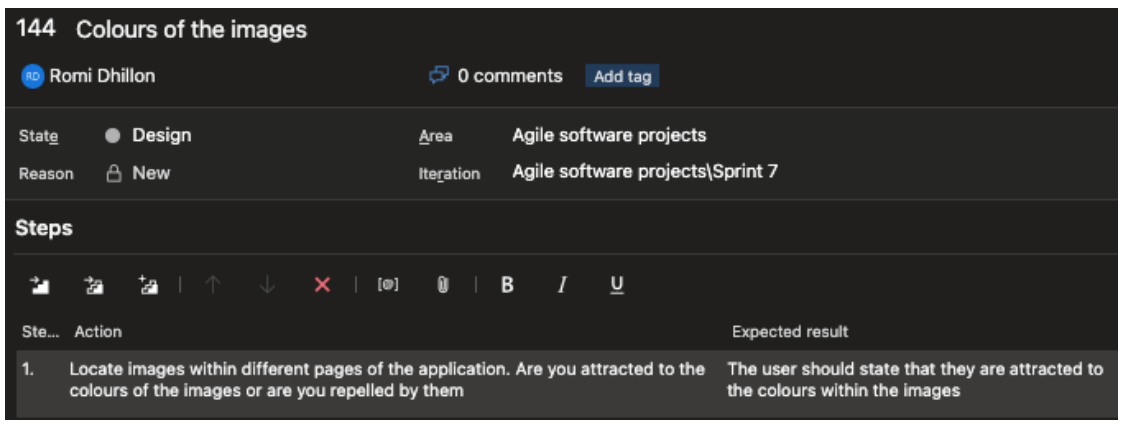


Figure 6.4 – User Interface test case 144

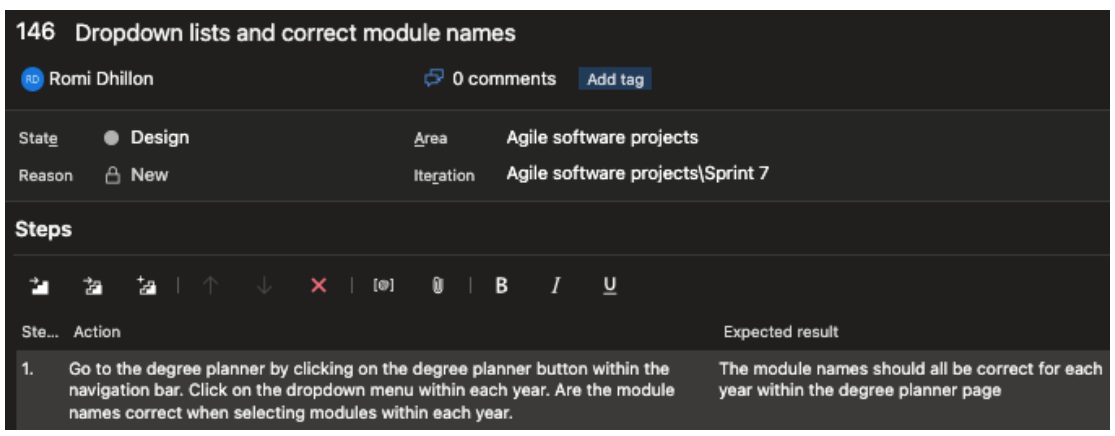


Figure 6.4 – User Interface test case 146

6.4.1. Modifying the code to address failed user interface tests

Test cases 140 and 143 failed during user interface testing. Test case 140 failed as users stated that the shade of red used was not alarming enough and was too dark. The shade of red was changed to a brighter shade and the test case passed on the second attempt.

Test case 143 failed as various users identified spelling mistakes on different pages of the web application. This was fixed by team members performing multiple visual inspections of the text on different pages and correcting the spelling mistakes. The test case passed on the second attempt.

6.5.Revisiting the project plan and risk register

After concluding all the tests and producing all the artefacts, the high-level project plan was revisited, and thirteen out of fourteen tasks were completed. The final task is to review and submit the report, which will also be completed shortly.

The risk register was continually reviewed and updated at the end of each sprint meeting once a week. The final version of the risk register is seen in Figure 6.5, which shows that all tasks have a status of closed with comments about how they were closed out.

0		Category
Risk Categories:		
Strategic		S
Legal and Regulatory		L
Political/Economic		P
Environmental		E
Technical/Operational/Infrastructure		T

Figure 6.5 – Final iteration of risk register

No	Date last reviewed	Risk description	Category	Risk Owner (Accountable)	Inherent Risk			Mitigation Actions	Status	Comments on Status	Target Closure
					Impact	Likelihood	Exposure				
1	22-Jul-22	There is a risk that all of the points within the marking rubric will not be fulfilled as part of the report	S	Romi Dhillon	High	Low	Med	Ensure that each activity within the marking rubric is focused on when planning activities on a weekly basis as part of sprints.	Closed	All points within the marking rubric have been fulfilled.	20-Aug-22
2	22-Jul-22	There is a risk that the report will not have an adequate flow from start to finish, especially when adding multiple different artefacts.	T	Romi Dhillon	High	Low	Med	Get started with the report early and ensure that it takes the reader through a journey	Closed	The report was completed by prioritising the journey and flow.	20-Aug-22
3	22-Jul-22	There is a risk that different team members have different expectations regarding some of the pages within the application	S	Romi Dhillon	Low	Med	Low	Conduct additional team meetings so that all team members have aligned expectations associated with the content of each pages.	Closed	All expectations have been discussed and aligned.	05-Aug-22
4	22-Jul-22	There is a risk that there will not be enough time to conduct all of the different types of testing required within the project.	S	Romi Dhillon	Med	Low	Low	Complete the first iteration of application development by 1st August. This will leave ample time for testing.	Closed	All tests have been completed within the required timelines.	01-Aug-22
5	10-Jul-22	There is a risk that different team members will not be able to attend meetings due to time differences (due to location)	S	Romi Dhillon	Med	Med	Med	Meetings are to be conducted at a time/day once a week which suits everyone. This decision will be made on slack on a weekly basis.	Closed	Most of the team members attended each meeting every Sunday.	22-Jul-22
6	01-Jul-22	There is a risk that the same collaborative tools used in the midterm are not suited for software development	T	Romi Dhillon	Med	Med	Med	Ensure that all team members collaborate using Git and Azure DevOps.	Closed	Literature review and research conducted on these tools. These tools have been used throughout the project.	10-Jul-22
7	10-Jul-22	There is a risk that some aspects of functionality within the application are too ambitious. There are time constraints within the project, and there may not be enough time to implement honourous functionality.	T	Romi Dhillon	Med	Med	Med	Use a web development framework which is suitable. Also decide on features which can be taken out without compromising usability.	Closed	First iteration on target for completion by the end of sprint 3 (31st July)	10-Aug-22
8	10-Jul-22	There is a risk that team members have other modules and other commitments outside of work, leaving less time to dedicate to the project.	S	Romi Dhillon	Low	High	Med	Sprint task tracking will be conducted on DevOps and regular updates will be achieved in weekly team meetings.	Closed	Sprints and weekly team meetings were conducted to get regular updates.	01-Sep-22
9	10-Jul-22	There is a risk of COVID on a global scale causing health and economic impacts.	P & E	Romi Dhillon	Med	Med	Med	If team members are affected, then other team members will support the team member and pick up the additional workload.	Closed	Some team members have been affected by COVID, and other team members have pitched in to suport.	01-Sep-22
10	10-Jul-22	There is a risk of intellectual property and ethical breaches taking place when created the application.	L	Romi Dhillon	High	Low	Med	Ensure that ethical aspects are properly researched and do not use assets which may breach intellectual property rights.	Closed	Researched and imbedded in report.	10-Aug-22
11	10-Jul-22	There is a risk that the application will be enhanced without updating the prototypes first.	T	Romi Dhillon	Low	Med	Low	Ensure that the UX Designer engages with the application developer.	Closed	UX Designer engaged with the application designer throughout the project.	15-Aug-22

7. Conclusion

We first started the study by focusing on market research and conducting a literature review of project and risk management, development tools, and web development frameworks. This allowed for decisions to be made about the type of tools and frameworks that we want to use to develop the application and artefacts. The roles and responsibilities were clearly defined, and a high-level project plan and risk register were created to define key high-level activities and to log risks.

Various updates were made to the hi-fi wireframes from the midterm, applying user experience and user interface research when updating the wireframes to better suit users. Each change was explained in detail, highlighting the reasons for the modifications. Critical evaluation took place throughout the report, and the benefits and disadvantages of design changes were critically assessed when discussing updates and as part of the change impact assessment.
Functional requirements from the midterm were updated based on the design changes, and a critical review of ethical considerations was given along with decisions to avoid intellectual property infringement.

The application design was first initiated with an optimisation of the process flow, continuing its development based on the model from the midterm. The application architecture and structure were presented with clear models and diagrams, leading to a detailed description of key components of the application design (i.e. models.py, or routes.py). The application development process was detailed to a granular level, discussing each of the seven sprints, agile methods such as the Kanban board, and the GitHub repository.

The last section of the report exhibited usability testing and user interface testing, displaying each test case, action steps and expected outcomes. The steps taken to correct failed tests were discussed by discussing the steps taken to amend the code. The project plan and risk register were also revisited to discuss and display the final iteration of both documents.

Key points related to the change in design decisions, direction, and approach have been presented throughout the report. A few tasks could have been conducted to improve the study further. For example, a few user surveys were conducted as part of the midterm to get feedback on user preferences. Additional user surveys could have been conducted early on in this study to gain clarification on some of the user-centered design changes that were made as part of the design process.

Future software releases will focus on incorporating two features that were omitted as part of the design. The first is the filter functionality, which allows users to filter table data. The second is the addition of the search functionality, allowing users to search for text across different pages in the application. Both features will improve the user experience in the future.

Please continue to the next page.

8. Contribution Percentages

To ensure a fair process, each individual member of the team was given equal opportunity to contribute to the project throughout its duration. One team member named Eisha Tirazia was given multiple opportunities to contribute to the project and she did not contribute at all; she rarely engaged with the team. The images below show a slack conversation with Eisha on 1st September which prove these claims. In her own words, she states that 'it's her fault for not being able to play her part at all' as seen in Figure 8.1.

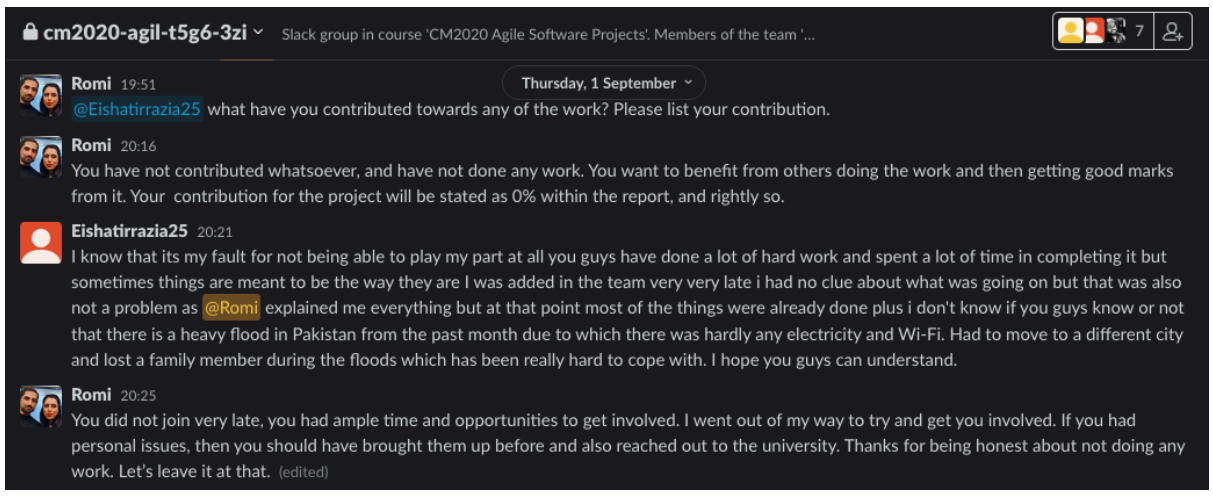


Figure 8.1 – Conversation with Eisha Tirazia on 1st September

All team members were asked their opinions on Eisha Tirazia's involved in the project on 3rd September. The responses from all team members can be seen in Figure 8.2

9. References

[1] apm, "Difference between agile and waterfall approaches to project management," 22 06 2022. [Online]. Available: https://www.apm.org.uk/resources/find-a-resource/agile-project-management/difference-between-agile-and-waterfall-approaches/?gclid=Cj0KCQjw0oyYBhDGARIsAMZEuMsflk2YxO7ZLw27yMsR8oiHIxhtdGCLkSAhsnpEy3LwrEZwKmxsT0AaAomcEALw_wcB. [Accessed 06 08 2022].

[2] Forbes, "Agile Vs. Waterfall: Which Project Management Methodology Is Best For You?," 22 January 2022. [Online]. Available: https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/. [Accessed 05 August 2022].

[3] PMI, "Agile versus Waterfall," 05 May 2022. [Online]. Available: https://www.pmi.org/learning/library/agile-versus-waterfall-approach-erp-project-6300. [Accessed 15 August 2022].

[4] P. Fitsilis, "Comparing PMBOK and Agile Project Management soware development processes," Academia, Greece, 2018.

[5] C. Starr, "Risk management, assessment, and acceptability," *Risk management, assessment, and acceptability,* vol. 5:2, 2005.

[6] OCRM, "Risk Management," 27 March 2022. [Online]. Available: https://www.ocrmglobal.com/blog/what-is-risk-management. [Accessed 18 August 2022].

[7] Cognizant, "DevOps Best Practices: Combine Coding with Collaboration," Cognizant 20-20 Insights, New York , 2021.

[8] EY, "EY and Microsoft Alliance," 21 June 2022. [Online]. Available: https://www.ey.com/en_uk/alliances/microsoft?WT.mc_id=10818322&AA.tsrc=paidsearch. [Accessed 17 August 2022].

[9] Stackify, "Top Source Code Repository Hosts: 50 Repo Hosts for Team Collaboration, Open Source, and More," 02 September 2021. [Online]. Available: https://stackify.com/source-code-repository-hosts/. [Accessed 20 August 2022].

[10] Slintel, "Source Code Management," November 2021. [Online]. Available: https://www.slintel.com/tech/source-code-management/github-market-share#:~:text=Github%20has%20market%20share%20of,Services%20with%201.07%25%20market%20share.. [Accessed 21 August 2022].

[11] P. S. Lokhande and F. A. Aslam , "Efficient Way Of Web Development Using Python And Flask.," *International Journal of Advanced Computer Research ,* vol. 6, pp. 18-19, 17 February 2015.

[12] M. S. Bonney, , M. de Angelis and D. Borgo, "Development of a digital twin operational platform using Python Flask," pp. 9-10, 07 01 2022.

[13] N. Chhetri, "A Comparative Analysis of Node.js (Server-Side JavaScript)," *Culminating Projects in Computer Science and Information Technology,* pp. 40-42, March 2016.

[14] usability.gov, "User Experience Basics," May 2022. [Online]. Available: https://www.usability.gov/what-and-why/user-experience.html. [Accessed 26 August 2022].

[15] usability.gov, "User Interface Design Basics," June 2022. [Online]. Available: https://www.usability.gov/what-and-why/user-interface-design.html. [Accessed 26 August 2022].

[16] Freepik Company S.L, "Organic flat feedback landing page template," 2022. [Online]. Available: https://www.freepik.com/free-vector/organic-flat-feedback-landing-page-template_13862317.htm. [Accessed 07 August 2022].

[17] wrike , "PROJECT MANAGEMENT GUIDE," 29 January 2022. [Online]. Available: https://www.wrike.com/project-management-guide/faq/what-is-a-sprint-in-agile/. [Accessed 13 August 2022].

[18] Microsoft , "Azure DevOps," August 2022. [Online]. Available: https://azure.microsoft.com/en-us/services/devops/. [Accessed August 2022].