Group 11 – The Bid

DiaManage

November 2025

Alistair McConnel

Group 11

Samara Cunningham (H00429775)

Alex Forsyth (H00433036)

Suprita Gajulapalle(H00431391)

Henry Ni (H00406013)

Zain Nawazani (H00409857)

Ismail Soleiman (H00473170)

Table of Contents

List of Figures

1. Requirements Specification

This report introduces a web-based application targeted towards diabetic users. It outlines the foundational frameworks and user requirements for the development of DiaManage, a web-based application commissioned by Alistair McConnell (‘The Client’). The application is primarily designed to assist diabetic users in monitoring and managing their health by setting and tracking personalized wellness goals.

1.1 Purpose

The aim of this project is to develop a web-based application that helps diabetic users track key health indicators and receive personalised suggestions to support a healthier lifestyle. Similarly, the purpose of this project is to promote a healthy lifestyle by enabling users to easily input and track their health data. The application will aid diabetic users in monitoring key health indicators and provide a platform for creating personalised tasks, such as checking blood sugar levels, as well as receiving mandatory activities assigned by healthcare professionals.

This report defines the project’s aims, objectives, and preliminary requirements that will guide the design, development, and evaluation phases throughout the lifecycle of the system. Through detailing the intended functionality, scope, and constraints of the proposed solution, this document will serve as a comprehensive reference for all team members, managers, academic supervisors, and potential stakeholders. In addition to supporting design and implementation activities, it ensures alignment with project goals and user needs. It also facilitates traceability between project objectives, system requirements, and future testing criteria which will be an essential component for both technical and managerial oversight.

This document is intended to be read by all individuals involved in the project, providing a structured basis for collaboration and informed decision-making at every stage of development.

1.2 Scope

The scope of this document encompasses the definition, planning and a specification of our groups proposed web-based application for diabetic users. It covers the project's overall vision, intended users, core and extended features alongside the high-level functional and functional requirements that will guide our design and development process.

Explicitly our document will:

i. Define the aims and objectives of the project, clarifying the problem it seeks to address and the expected outcomes

ii. Identify and describe all target user groups, including diabetic users and healthcare professionals who may interact with the system alongside researchers who may interact with data.
iii. Outline the core functionalities that constitute the Minimum Viable Product (MVP), as well as additional features that may be implemented in future iterations.
iv. Establish preliminary requirements (functional and non-functional) supported by use cases and diagrams where applicable.
v. Discuss the developmental approach, proposed technologies and design considerations that will shape the systems architecture.
vi. Introduce the assessment and validation strategy, detailing how success will be measured and verified through testing and evaluation.
vii. Provide the foundation for the next phase of the project during which the application will be implemented and refined based on this plan.

The scope of the project itself extends to the creation of a versatile and functioning web-based platform that allows diabetic users to log, track and visualise key health indicators (e.g., blood sugar levels, medicine intake, physical activity), and to receive personalised insights and recommendations. The system will also support the assignment and management of health-related tasks from healthcare professionals.

Future versions of the application may integrate advanced features such as aesthetic enhancements, gamification of the user experience and automated feedback mechanisms - these are beyond the MVP but will also be considered in the extended design vision.

1.3 Aim and Objectives

The primary aim of this project is to plan and design the development of a web-based application that empowers diabetic users to monitor, track and manage their health more effectively. The application seeks to promote healthier lifestyle habits by simplifying the process of recording key health indicators and by providing the users with personalized feedback and actionable insights.

In detail, our project aims to:

i. Support diabetes self-management by enabling users to easily input and track key health data such as blood glucose levels, physical activity and medication schedules.
ii. Enhance patient engagement and awareness by presenting health information in a clear, visual and user-friendly format that encourages regular monitoring and informed decision-making
iii. Facilitate communication and collaboration with healthcare professionals by providing a mechanism for professionals to assign mandatory tasks or review user data to support continuous and personalised care.

iv. Lay the groundwork for a scalable and secure digital health platform which can be further developed in later stages to integrate advanced features such as a gamified user experience, automated recommendations, and interoperability.

These aims collectively address the overarching goal of improving the quality of life for diabetic users through accessible digital tools and evidence-based self-management support.

The following objectives define how these aims will be achieved and provide measurable outcomes for assessing the success of the project.

1. **Core Health Tracking Functionality**
    a. Identify and plan features that allow diabetic users to input and monitor key health indicators including but not limited to; blood glucose levels, physical activity, and medication schedules.
    b. Specify the workflows and interactions required to support regular health data entry and progress tracking.
2. **User Interface and Experience Design**
    a. Develop a clear and intuitive interface structure that supports easy navigation and encourages frequent engagement with health data.
    b. Produce preliminary wireframes or interactive mock-ups to illustrate primary screens and navigation flows, ensuring alignment with usability principles.
3. **Personalised Insights and Recommendations**
    a. Define the logic and data requirements needed to generate personalised suggestions based on user-entered health data.
    b. Plan how insights and recommendations will be presented in a clear, actionable, and user-friendly format.
4. **Collaboration with Healthcare Professionals**
    a. Specify system mechanisms that allow healthcare professionals to assign, update, and review mandatory tasks for diabetic users.
    b. Outline how the system will manage, track, and display the completion status of these tasks to support ongoing care.

5. **Data Security, Privacy and Compliance**
    a. Identify necessary security protocols, data handling processes, and storage solutions to protect sensitive medical information.
    b. Ensure all planned system components comply with relevant data protection regulations (e.g., GDPR).

6. **System Architecture and Scalability Planning**
    a. Define a scalable system architecture capable of supporting future extensions such as gamification, automated feedback, and interoperability.

    b. Select appropriate technologies, frameworks, and development methodologies to ensure system maintainability and long-term reliability.

7. **Usability and Evaluation Strategy**
    a. Establish the methods and metrics for evaluating usability, accessibility, and user satisfaction among diabetic users.
    b. Develop traceable validation criteria to ensure alignment between project aims, system requirements, and testing activities.

1.4 Requirements

1.4.1 Functional Requirements

1.4.1.1 Priority 1 (Essential)

These are the core functionalities that form the Minimum Viable Product (MVP). Without these, the application cannot fulfil its purpose.

- **FR1.1**: The system must allow users to create an account with secure authentication (email/password).
- **FR1.2**: The system must allow users to log in and log out securely.
- **FR1.3**: The system must allow diabetic users to input and track key health indicators, including blood glucose levels, medication intake, and physical activity.
- **FR1.4**: The system must provide a dashboard displaying the most recent user health data and trends.
- **FR1.5**: The system must allow users to set, edit and delete personalized health goals (e.g., target blood glucose range, exercise frequency).
- **FR1.6**: The system must allow healthcare professionals to assign mandatory health-related tasks (e.g., monitoring blood sugar) for users.
- **FR1.7**: The system must send reminders to users for mandatory health tasks (e.g., checking blood sugar, taking medication).
- **FR1.8**: The system must allow users to mark health tasks as complete allowing them to track progress.
- **FR1.9**: The system must provide basic visualisations (graphs/charts) of key health indicators (e.g., blood glucose levels over time).

1.4.1.2 Priority 2 (High Priority)

These features are highly desirable and significantly enhance the system's usability, but the application can function without them in the MVP.

- **FR2.1**: The system must allow healthcare professionals to view user health data and activity logs.
- **FR2.2**: The system must provide recommendations based on tracked health data, such as tips for improving blood glucose levels or physical activity suggestions.
- **FR2.3**: The system must allow users to track and log additional health metrics, such as blood pressure, weight, or sleep patterns.
- **FR2.4**: The system must allow users to view their health progress over time, with visualizations comparing current and historical data.
- **FR2.5**: The system must provide a search and filtering option for users to find specific health data or tasks.

1.4.1.3 Priority 3 (Low Priority)

These features would be useful but are not essential for the MVP and can be deferred to later versions of the application.

- **FR3.1**: The system must allow users to connect to external health data sources, such as wearable fitness trackers (e.g., Fitbit, Apple Watch).
- **FR3.2**: The system must allow users to share their health data with family members or caregivers.
- **FR3.3**: The system must have a multi-language option to support users from different regions.
- **FR3.4**: The system must have a mobile application version that mirrors the functionality of the web-based application.
- **FR3.5**: The system must allow users to track and set personal lifestyle goals (e.g., weight loss, exercise targets) in addition to diabetes-specific goals.
- **FR3.6**: The system must allow users to communicate directly with their healthcare professional through the app (e.g., via secure messaging).

1.4.2 Non-Functional Requirements

1.4.2.1 Priority 1 (Essential)

These are fundamental requirements that ensure the system works securely, reliably, and is compliant with standards.

- **NFR1.1**: The system must ensure user data is stored securely, complying with data protection regulations (e.g., GDPR).
- **NFR1.2**: The system must ensure that all user health data is encrypted both in transit and at rest.
- **NFR1.3**: The system must guarantee high availability, with 99.9% uptime.

- **NFR1.4**: The system must comply with relevant healthcare standards and regulations (e.g., HIPAA for US users, GDPR for EU users).
- **NFR1.5**: The system must provide secure login and authentication mechanisms, including password hashing and two-factor authentication.
- **NFR1.6**: The system must provide robust error handling, ensuring that users receive clear feedback in the event of a failure or issue.

1.4.2.2 Priority 2 (High Priority)

These requirements ensure the system is user-friendly and capable of supporting a large number of users focusing on performance and usability.

- **NFR2.1**: The system must load all pages within 5 seconds to provide a fast and responsive user experience.
- **NFR2.2**: The system must be accessible on common web browsers (Chrome, Firefox, Edge, Safari) and on mobile devices (responsive design).
- **NFR2.3**: The system must support a large number of simultaneous users (up to 100 users) without significant degradation in performance.
- **NFR2.4**: The system must be user-friendly and designed with accessibility in mind, meeting WCAG 2.1 AA standards.
- **NFR2.5**: The system must have a clear, intuitive interface that can be easily navigated by diabetic users, including those with limited technical skills.
- **NFR2.6**: The system must provide real-time notifications for reminders and health task alerts.

1.4.2.3 Priority 3 (Low Priority)

These requirements are important for enhancing the user experience or extending the system's capabilities but are not critical to the basic operation of the MVP.

- **NFR3.1**: The system must be capable of supporting integration with third-party services and APIs (e.g., wearable devices, health data services).
- **NFR3.2**: The system must allow for customisation of the user interface (e.g., themes, font sizes, colour schemes).
- **NFR3.3**: The system must be able to provide reports that summarise health progress and trends, exportable in PDF or CSV format.
- **NFR3.4**: The system must support multi-language and multi-currency options, useful for international users.

1.5 UML

1.5.1 Use Case Diagram

The use-case diagram found in *Appendix 6.2.1* provides a high-level visual representation of the main interactions between users and the system. It identifies the two primary actors: Diabetic Users and Healthcare Professionals and outlines the core functionalities available to each. This includes: account creation, login and logout, manual health data entry, goal management, dashboard viewing and task assignments. The purpose of this diagram is to summarise the system's behavioural scope and show how each actor engages with key features of the application.

1.5.2 Textual Use Cases

The textual use case diagrams provide detailed textual descriptions of the system's major user interactions. Each use case outlines preconditions, postconditions, main and alternative flows to clarify how the system behaves under different circumstances. The use cases cover the essential actions of the application, such as:

- New User Registration *(Appendix 6.1.1)*
- User Login *(Appendix 6.1.2)*
- View Dashboard *(Appendix 6.1.3)*
- Input Health Indicators *(Appendix 6.1.4)*
- Set Personal Health Goals *(Appendix 6.1.5)*
- Edit Personal Goals *(Appendix 6.1.6)*
- Delete Personal Goals *(Appendix 6.1.7)*
- Assign Mandatory Tasks *(Appendix 6.1.8)*
- Mark Mandatory Tasks or Personal Goals as Complete *(Appendix 6.1.9)*

These descriptions ensure that each functional requirement is clearly mapped to user workflows.

1.5.3 Sequence Diagrams

The sequence diagrams show the step by step flow of information between the user, user interface, backend system and database during key operations. These diagrams show how the system processes user actions over time and help clarify system behaviour for implementation and planning. The sequence diagrams included represent three core processes:

- New User Registration (*Appendix 6.2.2.1)*
- Healthcare Professional Assigning a Task (*Appendix 6.2.2.2)*
- User Marking an Item as Complete (*Appendix 6.2.2.3)*

1.5.4 Activity Diagram

The activity diagram (*Appendix 6.2.3)* shows the typical daily workflow of a Registered Diabetic User when interacting with the system. After opening the application and reaching the dashboard, the user can review their daily checklist and choose from several possible actions. These include entering new biomarker data, marking a task or goal as complete, or viewing statistics and progress graphs on the Statistics page. The model shows that the user may perform multiple actions in any order, returning to the dashboard after each activity. The session ends when the user logs out or closes the application. This diagram provides a high-level behavioural overview of how users could interact with key features of the system during normal use.

1.5.5 State Diagram

The state diagram (*Appendix 6.2.4)* models the lifecycle of a mandatory task or personal goal within the system. Items begin in a created state, either generated by a Healthcare Professional (mandatory task) or the Diabetic User (personal goal). Mandatory tasks move into an assigned state before becoming active. Once visible to the user, all items enter the Incomplete state until the user marks them as complete. A completed item may either be undone (returning to incomplete) or automatically archived after 24 hours. Items may also transition directly to archived if they are deleted by the user or Healthcare Professional. This diagram provides a clear representation of how tasks and goals progress through the system over time.

2. Risk analysis

A range of potential risks were identified within the project, product and business areas. Each risk was assessed based on its likelihood of occurrence and potential impact. Following the assessment, remedial action was established to avoid or minimize these risks.

Risk	Risk Type	Probability	Impact	Remedial Action
Lack of Communication.	People	Low	Serious	The group manager will be informed of communication issues, and additional meetings will be scheduled to improve collaboration.
Illness and Unexpected Absence.	People	High	Tolerable	Having a cohesive group allows members within the team to understand each role and therefore continue if an absence occurs.
Lack of team coordination.	People	Moderate	Serious	Weekly meetings and meeting minutes allow coordination to be documented and discussed. An increase in meetings should take place.
Niche targeted users/ Lack of participants	Product	Moderate	Serious	Faux data allows for testing to be done with generic participants, as well as targeted users.
Missing deadlines	Estimation	Moderate	Catastrophic	The Gantt chart is used to keep track of all deadlines. Increase in workload, communication and organization to recoup.
Skill gaps.	People	Low	Catastrophic	SWOT analysis is used to assign the most competent team members to given tasks. Specialists will coordinate with team members to provide training.
Conflicts within the team.	People	Moderate	Tolerable	Team members will be encouraged to voice

				concerns as early as possible to support collaboration and early mediation. In the event of an escalated conflict, the group manager will be informed.
Lack of documentation.	People	Low	Serious	Procedural documentation, such as meeting minutes, allows for a time-lined analysis of the team's progress. Additionally, shared workspaces allow all team members to participate in documentation.
Test Coverage.	Estimation	Moderate	Catastrophic	The Requirement Specification and Project Plan provides a clear list of prioritized testing necessities.
Lack of software security.	Tools	Moderate	Serious	Software security will include cyber security and stay in line with GDPR. If needed, outside sources, including lecturers, can be consulted.
Software crashes.	Tools	Moderate	Serious	Backup versions of the software will be maintained to restore functionality quickly. Extensive testing is used as a preventative.
File Deletion	Tools	Low	Catastrophic	Backups of the project will be in each team member's workspace as well as in a shared workspace to decrease likelihood.

3. Project decisions and plan

3.1 Team Management

3.1.1 Team Roles

As a team, each member will be informed about every section of the project. However, considering the SWOT analysis and individual skills/preferences, formal roles have been assigned to allow clear understanding of task division.

Henry – Technical Manager (Back-end Developer)

Samara – Reporter (Front-end Developer)

Ismail – Organisational Manager (Full-Stack Developer)

Suprita – Liaison (Front-end Developer)

Zain – Financial Coordinator (Full-Stack Developer)

Alex – Tester (Back-end Developer)

3.1.2 Team Communications

Main communication between the team is conducted through WhatsApp. Additionally, a shared workspace on Notion is used to keep all documentation in one place. This way, a constant update of reports, meeting minutes, to-do lists, etc. is available. The Gantt chart is also included within the notion page to give a clear timeline of tasks. Moving forward, the development will be managed through GitLab, enabling all team members to maintain synchronised code.

3.2 Gantt Chart

As shown *(Appendix 6.3 and 6.4*), the Gantt chart breaks each stage into individual tasks with timelines. The structured breakdown allows for efficient work distribution and gives a clear, visual plan to adhere to. The workload was distributed to allow for an even work balance between all team members. Despite this, the workload between stage 1 and 2 is skewered to utilise strengths and weaknesses of team members.

3.2.1 Stage 1 (The Bid)

Stage 1 "The Bid" focused on analysing the problem, defining the requirements, planning the technical and organisational approach for the project, costing the project and creating the Usability Evaluation.

The team put together a comprehensive requirements specification defining the purpose, scope, aims and functional/non-functional requirements supported by UML diagrams and textual use cases. Following this, a risk analysis was carried out. Key project decisions were made such as technical environments to be used, development methodology and basic system architecture. Initial costings were created through resources and timelines, and initial prototypes were designed through usability evaluation for an early visual.

Overall, these tasks established the strategic and technical baseline for development in Stage 2. The Gantt Chart *(Appendix 6.3)* ensures the team begins implementing with well-defined goals and team roles.

3.2.2 Stage 2

Stage 2 represents the implementation and evaluation stages of the project. It takes the analysis and design concepts from Stage 1 and will result in a functional, tested and user-validated product.

The SCRUM framework was used to design 4 sprints between January and March 2026 (*see Appendix 6.4 and 6.4.1*). Furthermore, an initial planning sprint (*Appendix 6.4.2)* (Sprint 0) takes place in December, before the extended break, to refine further sprints and set up the technical environment. Each sprint begins with a group meeting and has continuous meeting throughout to maintain communication. Sprints have main goals to fully take advantage of the SCRUM process.

Throughout Stage 2, the team will continuously integrate through GitLab, perform buddy coding and conduct continuous testing when appropriate. It is important to note that Stage 2 is subject to change as the specification has yet to be released. Moreover, the report can be a continuous development despite the Gantt chart and the roles can be relaxed when appropriate due to the cohesiveness of the team.

3.2.2.1 Sprint 1 – MVP Development

The first sprint is focused on the core functionality. It will be between the 14th of January 2026 and the 3rd of February 2026 (*see Appendix 6.4.3)* and encompasses basic needs for the application. Essential front-end development such as input fields for necessary data and statistical displays will take place. Furthermore, primary back-end development will occur, such as database schemas and statistical calculations. Any flaws between the connection of the front-end and back-end will also be resolved at this initial stage.

3.2.2.2 Sprint 2 – Extended Features

The second sprint will be between the 9th of February 2026 and the 20th February 2026 (*see Appendix 6.4.4).* Building upon Sprint 1, this will broaden the user interface and features offered within the application. For instance, the doctor and patient task interface and attributes will be fully developed. Some more examples include the development of push notifications, reminders and recommendations given the health data provided. In summary, this represents the last phase of refinement before implementation of security controls and begin rigorous testing.

3.2.2.3 Sprint 3 – Security and Testing

The third sprint will be between the 25th February 2026 and the 10th of March 2026 (*see Appendix 6.4.5).* Cyber security for all aspects of the website will be implemented whilst rigorous testing begins. Each testing stage has been laid out for a structured execution. Cross-testing will take place for a more thorough search. Additionally, Health and Safety, and GDPR concerns will be analysed to ensure the website meets the requirements and standards.

3.2.2.4 Sprint 4 – Usability Test

The fourth will be between the 13th of March 2026 and the 25th of March 2026 (*see Appendix 6.4.6).* It is dedicated to the usability test before the EXPO. The usability test will be refined and ran. Next multiple group meetings will occur and the findings will be used to refine the UI/UX. The sprint will conclude with preparations for the EXPO, ensuring the application is polished.

3.3 Software Decisions

To ensure efficient and maintainable development, the team decided that an existing array of libraries and web frameworks will be used, rather than developing components from scratch. This reduces implementation time.

Front-end Development will target user interaction for both doctors and patients. This includes user interfaces, task related interfaces (e.g. viewing tasks, mandatory task assignments, etc.) and biomarker data visualisation. The front-end layer will be implemented using HTML, CSS and JavaScript.

Back-end Development handles secure login and registration, databases management and data history comparison. The back-end layer will use Java Spring Boot, PostgreSQL and REST API.

Gitlab will be used as a git repository to keep code up to date for all team members. Buddy coding and peer-review coding will be used to ensure risks are kept to a minimum and teamwork across the development is cohesive.

3.4 Resources

The resources table (*appendix 6.6)* is based on each member's availability and parallel courses. An estimation of other courseloads is 30 hours, given 3 other courses for each team member. Additionally, the workload distribution of Stage 1 and 2 is considered (*see Appendix 6.5).* Exact calculations will vary depending on the workload provided by other courses. The distributed effort shows realistic team time management while also ensuring progress toward the project milestones. Stage 2 is approximate at best as exact workload is unknown for Semester 2, at this stage. Tasks conducted by all, are ommited from the tabel, but considered in the calculation of weekly averge hours.

4. Project Costing

The project costing section estimates the total budget required to develop, deploy and maintain the diabetes management web application. Costs are grouped into six main categories: People, Equipment, Training, Testing, Support & Maintenance, and Reserves. All values are given in pounds sterling (£).

4.1 Costing Breakdown

The costing breakdown chart (*Figure 1*) summarises how the total budget is distributed across these categories. It provides a quick visual indication of which areas represent the largest share of the overall cost and helps justify where most of the investment is being made in the diabetes management system. *Figure 1* shows the breakdown of all expenses considered for the project, divided into people, equipment, training, testing, support & maintenance, and reserves. This visual summary links directly to the detailed cost tables in *Section 4.3* and the final totals in *Section 4.4*, helping stakeholders understand how the different elements of the project contribute to the overall budget.

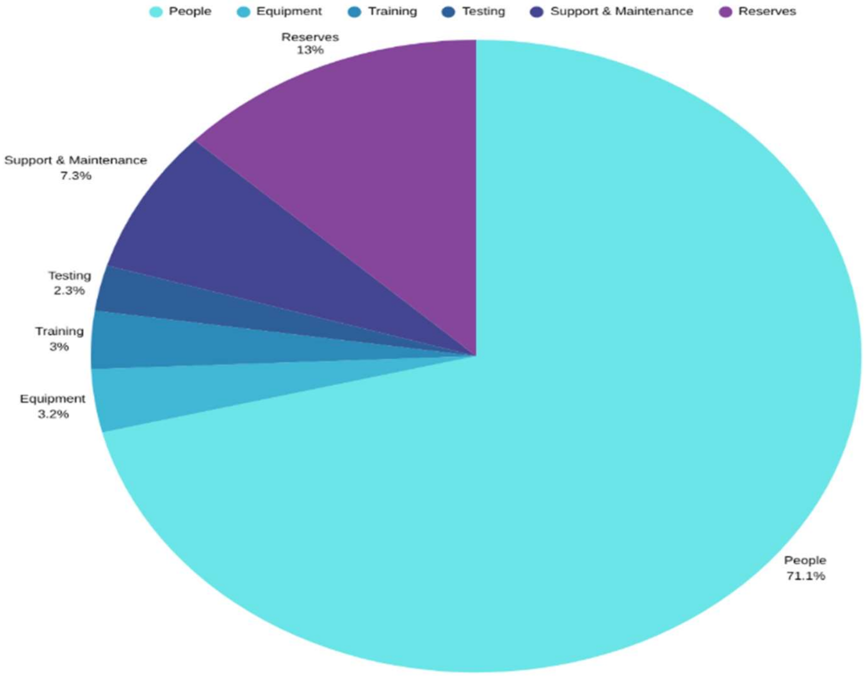


Figure 1- Pie Chart demonstrating the Costing Breakdown

4.2 Cost Analysis

This section explains what is included in each cost category (people, equipment, training, testing, support & maintenance and reserves) and why that spending is needed for the successful delivery of the diabetes management system.

4.2.1 Team Costs

People costs are the largest part of the budget and represent the time spent by the project team. The descriptions below summarise what work is included for each role when their hours are costed in the People category.

- Technical Manager (Henry)
  Oversees the system architecture and back-end implementation, ensuring that the REST API, database and security mechanisms (authentication, authorization, encryption) are designed correctly and meet the non-functional requirements.

- Organizational Manager (Ismail)
  Coordinates planning and scheduling, aligns the team with milestones, manages risks and ensures the project follows the agreed process and documentation standards.

- Financial Coordinator (Zain)
  Responsible for preparing and maintaining the project costing, tracking effort against budget, and helping the team understand how time and resources translate into financial impact. Zain also works as a Full-stack Developer, contributing to both front-end and back-end implementation.

- Full-stack Developer(s) (Ismail & Zain)
  Implement core functionality across both back-end and front-end. This includes user registration and login, biomarker input forms, task management, and integration between the database and the user interfaces. For Zain, full-stack development is a sub-role alongside his main responsibility as Financial Coordinator.

- Front-end Developer(s) (Samara & Suprita)
  Responsible for building the user interfaces for diabetic patients and healthcare professionals. They design and implement pages and components so that screens are responsive, accessible and easy to use, and ensure that charts and graphs correctly visualise biomarkers and trends. These responsibilities align with their main roles as Reporter/Documentation lead (Samara) and Liaison with stakeholders (Suprita).

- Tester (Alex)
  Designs and executes test plans, including functional, integration and regression testing. Ensures that critical functions (e.g. login, data entry, data viewing, notifications) work as expected and that defects are identified and fixed early. Alex also contributes as a Back-end Developer for selected features, which is taken into account in the role-splitting.

4.2.2 Equipment Costs

Equipment costs include the hardware and tools required for project development and deployment.

The Hosting Server / Cloud VM provides the infrastructure for hosting the web application, database and API in a test and deployment environment. This is required for integration testing and for demonstrating the system to stakeholders.

The Development Hardware includes the laptops/desktops used by developers and the tester throughout the project lifecycle. These machines run IDEs, local test databases, build pipelines and testing tools. The Development Tools include software such as IDEs, Git-based version control, collaboration tools and issue trackers. Where possible, free or open-source tools (e.g. GitLab, VS Code, Spring Boot, PostgreSQL) are used to reduce license costs.

4.2.3 Training Costs

Training costs cover the expenses required to ensure the team is proficient in the tools and practices needed to build a safe and reliable medical web application. Investing in training reduces the risk of later rework, poor security decisions or misconfigurations.

Security & Data Protection Training familiarises the team with relevant data protection principles (e.g. confidentiality, secure password storage) and good practices for handling sensitive health data. Development Tool Training provides training on the main frameworks and tools used for coding, database design, testing and project management. This includes training on the chosen web framework, database, version control workflow and CI/CD setup.

4.2.4 Testing Costs

Testing costs include expenses related to validating the performance, security and usability of the system.

Software Functional Testing verifies that each requirement is implemented correctly: account creation, login, biomarker entry, visualization, task assignment, notification handling and so on. Security Testing ensures the system is protected against common security threats such as weak authentication, injection attacks or unauthorized data access. Usability / Integration Testing confirms that the patient and healthcare professional interfaces are intuitive and that all components (front-end, back-end, database, notification service) interact seamlessly.

4.2.5 Support & Maintenance Costs

Suport and Maintenanace costs are the ongoing costs required to keep the system running smoothly after the initial deployment.

Software updates cover bug fixes, compatibility updates and small improvements carried out regularly (e.g. quarterly updates) to keep the system stable and secure. Software enhancements cover larger feature updates requested by stakeholders, such as support for additional biomarkers, improved dashboards, or integration with external health platforms.

Planning for support and maintenance ensures the system remains useful and safe over its lifetime.

4.2.6 Reserves

A contingency reserve is included to cover unforeseen expenses, such as under-estimated development effort, additional testing cycles, or changes in requirements. In this project, reserves are set at 15% of the total cost before reserves.

4.3 Total Cost

The cost calculation translates the qualitative analysis from *Section 4.2* into quantitative estimates. For each cost category, the number of hours or units, the relevant rate and the resulting expected cost are specified. This provides a transparent basis for the overall project budget and makes it clear how each role and resource contributes to the final figure.

4.3.1 Cost Caluclations -People

Appendix 6.7.2 shows how the total people cost of £79,500 is obtained. For each role, it lists the estimated number of hours to be spent on the project, the number of people in that role, and the hourly rate. Multiplying hours, units and rate gives the expected cost per role, and summing these values gives the overall people cost. These hours cover both the main and sub-roles described later in *Section 4.5.*

4.3.2 Cost Calculations - Equipment

Appendix 6.7.3 presents the equipment-related costs. It distinguishes between the cloud-hosted server/VM used for deployment and integration testing, the development machines used by the team, and the development tools. Because open-source technologies are used where possible, the tools themselves have zero licence cost, and the only equipment expenditure is for hosting and hardware, giving a total of £3,600.

4.3.3 Cost Calculations - Training

Appendix 6.7.4 details the training budget. It includes one security/data protection session and one development tools/frameworks session for all six team members. The table shows the per-person cost of each activity and how this scales up to the total training cost of £3,300. This investment ensures that the team shares a consistent understanding of both security practices and the chosen technical stack.

4.3.4 Cost Calculations - Testing

Appendix 6.7.5 summarises the estimated cost of the main testing activities. Each line represents a distinct type of testing by functional, security and usability/integration with a single budgeted amount that covers the effort, tools and any external support required for that activity. Adding these values gives a total testing cost of £2,600, reflecting the importance of thorough verification for a medical web application.

4.3.5 Cost Calculations – Support and Maintenance

Appendix 6.7.6 captures the expected support and maintenance costs for the first year after deployment. The software update line represents four quarterly update cycles, each budgeted at £800, while the software enhancements line provides a lump sum of £5,000 for larger improvements requested by stakeholders. Together, these figures give a support and maintenance budget of £8,200 and show how ongoing work is planned and funded rather than treated as an afterthought.

4.4 Total Expected Cost

The individual category costs from *Section 4.3* are combined to give a single overall budget for the diabetes management system. This total includes all people, equipment, training, testing, and support & maintenance costs, plus a contingency reserve to cover uncertainty in the estimates.

Appendix 6.7.1 summarises these totals and shows how the reserve is derived from the subtotal of the other cost categories. Reserves are calculated as 15% of the subtotal:

$$79{,}500 + 3{,}600 + 3{,}300 + 2{,}600 + 8{,}200 = 97{,}200$$
$$0.15 \times 97{,}200 = 14{,}580$$

This means that the project is expected to require £111,780 in total to develop, deploy and support the system for the initial period. Including the reserve makes the estimate more realistic by explicitly allowing for extra development effort, additional testing or small requirement changes that are likely to occur in practice.

Appendix 6.7.1 brings together all the previous cost categories and shows how they contribute to the final project budget. People costs account for the majority of the total, with smaller but still significant contributions from equipment, training, testing and support & maintenance. A 15% contingency reserve is then added to the subtotal to allow for uncertainty in the estimates. This leads to a total expected cost of £111,780, which forms the basis for budgeting discussions and any contractual agreement with stakeholders.

4.5 Cost Description

The cost description explains how the figures in the costing tables relate to the roles, activities and resources used in the project. It clarifies what is included in each category and how responsibilities are divided within the team.

Appendix 6.8 shows how each team member's time is split between their main role and any sub-roles. Main roles (such as Technical Manager, Organisational Manager, Reporter/Documentation, Liaison with stakeholders, Financial Coordinator and Tester) are the primary drivers of the people costs in the earlier tables. Sub-roles (such as additional development or reporting responsibilities) explain how team members contribute to more than one aspect of the project.

In particular, Henry acts as Technical Manager with additional back-end development responsibilities; Ismail works as Organisational Manager and also as a Full-stack Developer; Zain's main role is Financial Coordinator and he also works as a Full-stack Developer; Samara's main role is Reporter/Documentation lead with additional front-end development tasks; Suprita's main role is Liaison with stakeholders supported by front-end development work; and Alex is primarily the Tester but also contributes as a Back-end Developer.

This demonstrates that the costed hours cover both core responsibilities (e.g. management, development, testing) and secondary activities (e.g. documentation, liaison, financial coordination) that are necessary to deliver a usable and well-managed system.

The project uses hardware owned or provided by the team/university and primarily relies on open-source tools such as Java, Spring Boot, PostgreSQL, HTML, CSS and JavaScript. Because of this, development tool licences are effectively free, and only hosting and hardware are costed. This keeps equipment expenditure relatively low while still providing an appropriate development and deployment environment.

One security/data protection training session and one development tool training programme are budgeted for all team members. These activities ensure a common understanding of frameworks, tools and security practices, which reduces the risk of errors, rework and security weaknesses later in the project. The training costs therefore represent an investment in quality and compliance rather than an optional extra.

Testing costs cover the structured verification of the system, with dedicated budgets for functional, security and usability/integration testing. Given the sensitivity of health-related data and the potential impact of incorrect advice or missing information, it is important that sufficient resources are allocated to thorough testing before deployment.

Regular updates (e.g. every three months) are planned to keep the system secure and up to date. The first update could be offered free as part of the initial contract, while subsequent updates are charged according to this cost model. An additional budget is also reserved for larger enhancements, such as new biomarkers or improved dashboards, ensuring that the system can evolve in response to user feedback and changing clinical needs.

Finally, the total cost is increased by 15% to create a reserve for unforeseen circumstances, such as additional change requests, extra testing or integration work with external systems. This contingency reduces the risk of cost overrun and demonstrates that uncertainty has been explicitly considered in the budget.

4.6 Project Costing Summary

The project costing analysis outlines the financial resources required to develop, deploy and maintain the diabetes management web application. By categorising expenses into people, equipment, training, testing, support & maintenance, and reserves, the budget ensures that every critical activity from secure development and testing to future enhancements is properly funded.

The final expected cost of £111,780 (including a 15% contingency reserve) supports realistic budgeting and planning, and it positions the project for successful completion and long-term sustainability in a healthcare context.

5. Usability evaluation of Mock-ups

5.1 Prototypes

JustInMind was used to design an interactive mock-up prototype of the app. This allows study participants to interact with the app. It was designed on a mobile interface however the final application will be able to scale to any size. *Appendix 6.11.1* shows an overview of general statistics and a daily checklist for users. This will be the home page. *Appendix 6.11.2* demonstrates the statistics page, showing an in-depth analysis of blood sugar, blood pressure and heart rate. The activities page (*Appendix 6.11.3*) shows any prescribed activities and allows the user to add their own tasks.

5.2 Usability Study

5.2.1 Abstract

This study will test the perceived usability of diabetes tracker web application. The goal of this study is to determine the usability of the app and identify areas and features which need improvement.

5.2.2 Introduction

The primary motivation behind this study is to gather feedback on the interactive mock-up (prototype) of our app. This will allow us to identify any problem areas of our app and make informed changes to the final design. Throughout this section of the report the prototype will be referred to as the app.

The aims of the testing are:

- Test the app under controlled conditions and help identify if usability standards have been met
- Gather qualitative and quantitative feedback about the design of the app to determine its usability.
- Gather standardised testing results in the form of a System Usability Scale (SUS).
- Identify if **NFR2.5** (The system must have a clear, intuitive interface that can be easily navigated by diabetic users, including those with limited technical skills) has been met.

The app is designed for diabetic users and health professionals so these would be our ideal population to study. However, our testing is limited to the following sub-group of the general population as required by the ethics statement:

- Heriot-Watt University (HWU) staff members or students
- Not a vulnerable person
- Able to give informed consent

Our testing protocol (*Appendix 6.10*), consent form (*Appendix 6.8*) and survey (*Appendix 6.9*) help enforce these requirements. The testing protocol ensures the correct steps are followed so a participant must sign a consent form before moving on with the testing. Additionally, the survey asks participants if they have understood and signed a consent form, are over 18 years old and are HWU staff or student. If any of these are answered wrong the survey will not allow the participant to continue. If the lead investigator was to identify a user not meeting these requirements the study would be terminated immediately. The participant is also informed of the potential risks involved.

5.2.3 Test Design

The tests were run by a principal investigator on several participants under controlled conditions. The testing protocol (*Appendix 6.10*) highlights this for the investigator and ensures these conditions are met each time the study is run. The location of the testing took place in the study space on the 4th floor of the Post Graduate building on the Heriot-Watt, Edinburgh campus. This can introduce some confounding variables. As this area is open to anyone the noise levels varied throughout testing which could distract the participants. Additionally, the ethics requirement is a specific subset of the general population, this could make it harder to find an ideal population to study. Participants were gathered in a group and interviewed individually one after another, this gives a confounding variable of participants sharing knowledge of the test between interviews, however, the testing protocol asks participants not to share answers. The test design is between as only 1 version of the design is being tested on all participants. This avoids participants learning between attempts. Each participant followed the testing protocol which in summary states each user was asked to sign a consent form then answered a demographic survey. After this they were told briefly about the app. Next, they were given 3 minutes to test the application. After this they completed a few open-ended questions, this was our initial questionnaire which gathered subjective and qualitative data. Then they completed a System Usability Scale (SUS) which is a standardised test which gathered quantitative and subjective data. At the end they were asked if they had any extra comments, this was the exit questionnaire which provided subjective qualitative data.

5.2.4 Data Collection

The data was collected digitally for each participant. This was done by electronically signing a word document consent form (*Appendix 6.8*) and completing the survey (*Appendix 6.9*) using Microsoft Forms. This collected pseudo anonymised data. Each participant was assigned a unique participant ID. This is the only identifying piece of information kept on their test results. A password protected master sheet has been kept by the principal investigators linking consent forms to participant ID's. This allows for us to remove a participants results if they wish to do so. This ensures the right to erasure which is required by GDPR. All personal data is being held in accordance with GDPR legislation. This is done by storing all personal data in password protected and encrypted documents. This is the only place any personal data is stored.

5.2.5 Analysis

5.2.5.1 Demographic

6 participants were gathered for testing. Below is a breakdown of the demographic of these participants:

- All participants were HWU students
- All participants were between the ages of 20-21
- 4 participants study economics, 1 mechanical engineering and 1 computer science
- 5 participants were male and 1 was female
- 2 participants had basic knowledge of computers, 3 had intermediate knowledge and 1 had advanced knowledge.
- All non-diabetic

This demographic limits the generalisability of the study as it is highly homogeneous. All participants are 20-21, Heriot-Watt students and non-diabetic. Additionally, the sample size is only 6 and therefore small. This reduces the ability to draw broader conclusions across different age groups, backgrounds and user types.

5.2.5.2 Initial Questionnaire

The initial questionnaire asked participants about the app's main features, design, and statistics page. Most participants (5/6) correctly identified the main features of the app to be related to diabetes. The other participant identified the app as an "activity tracker". This suggests that the purpose of the app is clear but some additional visual cues could be added to reinforce this. The general feedback on the overall design of the app was positive. However, 3 participants commented on some features not working as expected. These were adding an activity, the search button and scrolling on statistics page. As this was only a prototype not all features were implemented. The feedback on the statistics page was generally positive however, participant 2 found them confusing. This could be because there was no actual data inputted for the prototype. Participant 3 mentioned this in their feedback. Future studies could include realistic faux data to avoid confusion.

5.2.5.3 SUS Analysis

The SUS is a standardised test which asked participants to rank the app based on 10 questions using a scale with the options: strongly disagree, disagree, neutral, agree and strongly agree with the respective values of 1-5. The results from each question of the SUS were plotted on a graph shown in *Figure 2*.

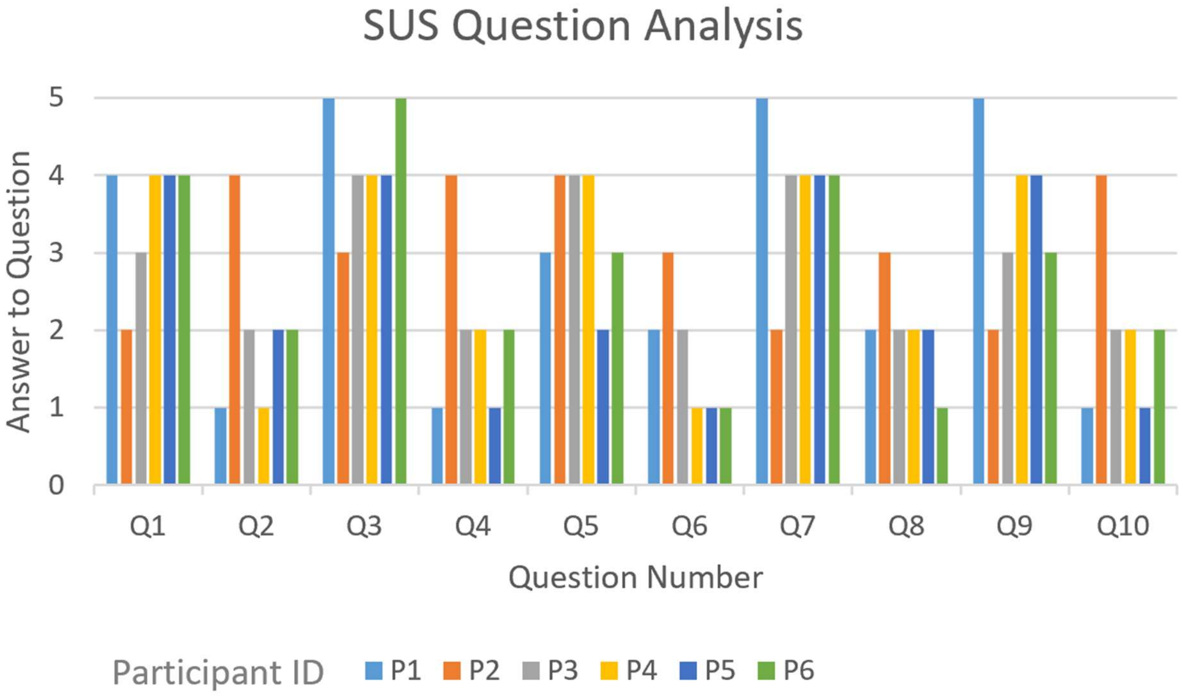


Figure 2 – Graph detailing SUS Results

From *Figure 2* we can see that all questions had varying levels of agreement from each participant. The answers from the SUS were then used to calculate the more useful SUS scores for each participant shown below.

Participant ID	**SUS Score (Out of 100)**
1	87.5
2	37.5
3	70
4	80
5	77.5
6	77.5

This results in a mean score of 71.67 (2 decimal places) with a standard deviation of ±16.12 (2 decimal places). This mean value is above the average SUS score of 68 and places the app to a "good" score. Therefore, this suggests that the app is usable. However, both participant 1 and 2 scores stand out. Their scores are respectively 15.83 points above and 34.17 points below the mean. As participant 2 is greater than 2 whole standard deviations below the mean, this indicates an unusually negative perception of usability compared to the rest of the participants. This led to their results being analysed in more depth to identify potential areas of improvement to design and any confounding variables.

5.2.5.4 Participant 2 (P2)

It was chosen to look at the demographic surveys and identify any areas this participant was an outlier in. P2 was one of only two participants who choose basic computer knowledge in the demographic survey. The other was P1. This data does not seem relevant as the other participant with basic knowledge scored highest in the SUS. However, this question was very subjective. There were no examples or standards for what basic might correlate to. Additionally, P2 was the only female and 21-year-old in the study. However, there is nothing to suggest either of these is the causation of the lower results and therefore must be a coincidental correlation. Therefore, it was chosen to next look at P2's qualitative feedback from the initial and exit questionnaires. For the initial questions they notably stated the graphs were confusing. This could be a contributing factor to the lower SUS score. Additionally, for the exit questionnaire they stated that it took too long to realise they can scroll on the statistics page. This highlights key issues with the statistics page which may be the cause of the lower SUS score and therefore effect the usability of the app.

5.2.5.5 Exit Questionnaire

The exit questionnaire asked if the participant had any comments or suggestions. This question highlighted a few key issues. Participant 2 specifically mentioned scrolling on the statistics page. This could be because each participant was testing a mobile device design using a laptop. This is a confounding variable which was not originally considered. This is backed up by participant 4 who stated, "Using a laptop to test a design of a mobile phone website was confusing". Additionally, participant 5 identified the app didn't feel useful to them as they are not diabetic. However, they did state they can see its usefulness. This was a concern with testing and was identified as a confounding variable. However, we were unable to locate diabetic participants for the study. This could indicate our product is targeted towards a niche market and subset of users.

5.2.6 Conclusion

In conclusion, the testing showed that overall, the app is usable. However, there are areas of the design which can be changed to improve the usability and user experience. The statistics page will need to be redesigned to be more intuitive. Participants identified scrolling on this page to be confusing. In the final design it may make more sense to use arrows to navigate the graphs instead of scrolling as the home page does this and had no issues during testing. This would ensure NFR2.5 is met. This will help ensure the final product is usable and provides a positive user experience.

6. Appendix

6.1 Textual Use Case

6.1.1 Register Account

Use Case ID: 1

Goal: New user creating account to use the system

Primary Actor: New Diabetic User

Secondary Actor(s): Healthcare Professional (via referral code)

Preconditions:

- The user is not currently logged in.
- The user does not already have an existing account under the same email address.

Postconditions:

- A new user account is created and stored in the system.
- The account is either:
  - Linked to a healthcare professional (if a valid doctor code was entered).
  - Created without a doctor link (if no code was entered or the user chose to skip it).

Main Flow:

1. The user navigates to the registration page.
2. The system displays a registration form asking for required details (e.g. name, email, password, basic profile information).
3. The user enters their registration details and submits the form.
4. The system validates the input (e.g. required fields present, password format correct, email format valid).
5. The system checks whether an account with the provided email already exists.
6. The system prompts the user with the option: "Have you been referred by a Healthcare Professional?".
7. The user selects "Yes" and enters the referral code.
8. The system verifies that the code corresponds to an existing Healthcare Professional.
9. The system creates a new user account and links it to the corresponding Healthcare Professional.
10. The system confirms successful registration and redirects the user to the login page.

Alternative Flows:

4a. Input is not valid (e.g. password/ email wrong format, missed required fields).

1. If the user's input is not valid, the system displays a specific error message.
2. No new account is created.
3. System returns to step 2 of the main flow.

5a. Email is already registered.

1. If the email is already associated with an existing account, the system displays an error message (e.g. "An account with this email already exists").
2. No new account is created.
3. System returns to step 2 of the main flow.

6a. User has not been referred by a Healthcare Professional.

1. The user selects “No” when asked if they have a referral code.
2. The system creates a new user account without linking a healthcare professional.
3. System continues to step 10 of the main flow.

8a. Invalid referral code.

1. If the code does not match any registered Healthcare Professional, the system displays an error message (e.g. “Invalid Code”).
2. The system allows the user to:
    a. Re-enter the code.
        i. System returns to step 8 of the main flow.
    b. Skip linking to a Healthcare Professional.
        i. System continues to step 6a.2 of the alternative flows.

6.1.2 User Login

Use Case ID: 2

Goal: Allow a returning user to log into their account

Primary Actor: Registered Diabetic User or Healthcare Professional

Secondary Actor(s): None

Preconditions:

- The user is not currently logged in.
- The user already has an account.

Postconditions:

- The user is successfully logged in and redirected to the dashboard.

Main Flow:

1. The user navigates to the login page.
2. The system displays a login form asking for email and password.
3. The user enters their credentials and submits the form.
4. The system validates the input format (e.g. password not empty, email format valid).
5. The system verifies that the email exists and the password matches the stored credentials.
6. The system logs the user in.
7. The system redirects the user to the dashboard.

Alternative Flows:

4a. Input is not valid.

1. If the user’s input is not valid, the system displays a specific error message.
2. No login occurs.
3. System returns to step 2 of the main flow.

5a. Email not found.

1. If the email does not match any existing account, the system displays an error message (e.g. “No account found with this email”).
2. No login occurs.
3. System returns to step 2 of the main flow.

5b. Incorrect password.

1. If the password does not match the stored password for the given email, the system displays an error message (e.g. “Incorrect Password”).
2. No login occurs.

3. System returns to step 2 of the main flow.

6.1.3 View Dashboard

Use Case ID: 3

Goal: View dashboard (homepage) once the user logs in.

Primary Actor: Registered Diabetic User

Secondary Actor(s): None

Preconditions:

- The user is currently logged in.
- The system has access to stored user data (health indicators, goals, tasks).

Postconditions:

- The dashboard is displayed to the user with up-to-date information.
- No system state is changed (view-only, unless they take further action).

Main Flow:

1. The user is redirected to the dashboard after logging in (or navigates to it manually).
2. The system retrieves the user's most recent health indicator data (e.g. blood glucose, blood pressure, daily activity).
3. The system retrieves the user's personal health goals.
4. The system retrieves any mandatory tasks assigned by the Healthcare Professional.
5. The system retrieves the completion status of previous tasks.
6. The system generates the user's daily checklist (e.g. "Input biomarker data", "Review mandatory tasks", "Check today's statistics").
7. The system generates relevant graphs and visualisations (e.g. glucose trends over time).
8. The system compiles all retrieved information into a dashboard view.
9. The system displays the dashboard to the user.

Alternative Flows:

2a. No health data available.

1. If there are no stored health indicators for the user, the system displays placeholder messages or empty-state prompts (e.g. "No data recorded yet").
2. The dashboard still loads successfully.
3. System continues from step 3 of the main flow.

3a. No personal goals set.

1. If the user has no personal goals, the system displays an empty goals section with an option to create new goals.
2. System continues to step 4 of the main flow.

4a. No assigned tasks.

1. If no mandatory tasks have been assigned by the Healthcare Professional or there is no connection to the Healthcare Professional, the system displays ("No current assigned tasks").
2. System returns to step 5 of the main flow.

6a. Checklist not applicable/ No data yet.

1. If the user has no data or tasks yet, the system generates a simplified checklist or a placeholder message (e.g. "Start by entering your first biomarker reading").

2. System returns to step 7 of the main flow.

7a. Graph generation unavailable.

1. If the system cannot generate graphs, the system displays simple text summaries instead of charts.
2. System returns to step 8 of the main flow.

6.1.4 Input Health Indicators

Use Case ID: 4

Goal: Allow the user to record new biomarker or lifestyle data into the system.

Primary Actor: Registered Diabetic User

Secondary Actor(s): None

Preconditions:

- The user is currently logged in.
- The system is able to store health indicator data.

Postconditions:

- The user's new health indicator data is saved to the system.
- Dashboard and graphs will reflect the updated data on next view.

Main Flow:

1. The user navigates to the "Input Health Data" section of the app.
2. The system displays available categories of health indicators (e.g. blood glucose, medication, blood pressure, physical activity).
3. The user selects a specific category to input data for.
4. The system displays the corresponding input form for that category. Example:
    - Blood glucose page – level & time.
    - Activity page – steps, minutes, calories burned.
    - Medication page – type & dosage.
5. The user enters the required data and submits the form.
6. The system validates the input (e.g. numeric values, required fields, valid ranges).
7. The system stores the new data in the database.
8. The system confirms successful submission to the user (e.g. "Entry saved").
9. Use case ends.

Alternative Flows:

5a. User cancels input

1. If the user decides to cancel the input, the system discards the entered information.
2. Use case ends.

6a. Invalid input format.

1. If the entered data is invalid (e.g. non-numeric glucose level, missing required fields), The system displays an error message identifying the issue.
2. No data is saved.
3. System returns to step 4 of the main flow.

6b. Out-of-range values entered.

1. If the input is in valid format but medically unreasonable (e.g. calories burned = 100 000 kcal), the system warns the user and asks for confirmation or correction.

        a. If the user confirms the entry
            i. System proceeds to step 7.
        b. If the user cancels
            i. System returns to step 4 of the main flow.

7a. Database storage error

1. If the system fails to save the data (e.g. lost connection), the system displays an error message (“Unable to save data. Please try again”).
2. No data is saved.
3. System returns to step 5 of the main program again.

6.1.5 Set Personal Health Goals

Use Case ID: 5

Goal: Allow the user to create personalized health goals.

Primary Actor: Registered Diabetic User

Secondary Actor(s): None

Preconditions:

- The user is currently logged in.
- The system can store goal data in the database.

Postconditions:

- A new personal goal is created and stored.
- The user’s dashboard will reflect this goal on next view.

Main Flow:

1. The user navigates to the “Set Personal Goals” section of the app.
2. The system displays available goal categories (e.g. blood glucose range, steps per day, exercise frequency).
3. The user selects a goal category.
4. The system displays the goal creation form for the selected category.
5. The user enters the goal details (e.g. numeric target, time range, frequency).
6. The user submits the form.
7. The system validates the input (required fields, valid ranges, numerical correctness).
8. The system stores the new goal in the database.
9. The system confirms the successful creation (e.g. “Goal Added”).
10. The use case ends.

Alternative Flows:

4a. User cancels goal creation.

1. If the user cancels during the goal creation, the system discards any entered information.
2. The use case ends.

7a. Invalid goal input.

1. If the goal details are invalid (e.g. non-numeric value, negative target, missing required fields), the system highlights the error and prompts correction.
2. No goal is saved.
3. System returns to step 4 of the main flow.

7b. Out-of-range goal entered.

1. If the goal is valid format but unrealistic (e.g. “run 100km/day”), the system warns the user and asks for confirmation or correction.
    a. If the user confirms.
        i. System proceeds to step 8 of the main flow.
    b. If the user cancels.
        i. System returns to step 4 of the main flow.

6.1.6 Edit Personal Goals

Use Case ID: 6

Goal: Allow the user to modify previously created personal health goals.

Primary Actor: Registered Diabetic User

Secondary Actor(s): None

Preconditions:

- The user is currently logged in.
- The user has at least one personal goal saved in the system.

Postconditions:

- The selected goal is updated in the system.

Main Flow:

1. The user navigates to the “Activities” page.
2. The system displays a list of the user’s existing personal goals.
3. The user selects a specific goal from the list.
4. The system displays options to edit goal or delete goal.
5. The user selects edit goal.
6. The system displays the pre-filled goal editing form with the existing goal details.
7. The user updates the goal details and submits the form.
8. The system validates the updated input (required fields, valid ranges, numeric correctness).
9. The system stores the updated goal in the database.
10. The system confirms successful goal update (e.g. “Goal Updated”).
11. The use case ends.

Alternative Flows:

6a. User cancels edit.

1. If the user cancels while editing, the system discards any changes.
2. Use case ends.

8a. Invalid edited goal input.

1. If the updated goal is invalid (e.g. missing values, negative target, non-numeric numbers), the system displays an error message explaining the issue.
2. No changes are saved.
3. The system returns to step 6 of the main flow

8b. Out-of-range or unrealistic edited goal input.

1. If the updated goal is valid input but unrealistic (e.g. “run 10,000 km/day the system warns the user and asks for confirmation or correction.
    a. If the user confirms.
        i. System proceeds to step 9 of the main flow.

    b. If the user cancels.
        i. System returns to step 2 of the main flow.

9a. Database error during save.

1. If the system is unable to edit the goal due to an internal error, the system displays an error message (“Unable to edit goal. Please try again”).
2. No changes are saved.
3. System returns to step 2 of main flow.

6.1.7 Delete Personal Goals

Use Case ID: 7

Goal: Allow the user to remove previously created personal health goals.

Primary Actor: Registered Diabetic User

Secondary Actor(s): None

Preconditions:

- The user is currently logged in.
- The user has at least one personal goal saved in the system.

Postconditions:

- The selected goal is deleted from the list.

Main Flow:

1. The user navigates to the “Activities” page.
2. The system displays a list of the user’s existing personal goals.
3. The user selects a specific goal from the list.
4. The system displays options to edit goal or delete goal.
5. The user selects delete goal.
6. The system asks the user to confirm the deletion.
7. The user confirms.
8. The system deletes the selected goal from the database.
9. The system displays a confirmation message (e.g. “Goal Deleted”).
10. The use case ends.

Alternative Flows:

6a. User cancels Deletion.

1. If the user selects “Cancel” at the confirmation prompt, the system does not delete the goal.
2. Use case ends.

8a. Database error during deletion.

1. If the system is unable to delete the goal to do an internal error, the system displays an error message (“Unable to delete goal. Please try again.”).
2. System returns to step 2 of main flow.

6.1.8 Healthcare Professional Assigns Mandatory Tasks

Use Case ID: 8

Goal: Allow a Healthcare Professional to assign mandatory health-related tasks to a user.

Primary Actor: Healthcare Professional

Secondary Actor(s): Registered Diabetic User / Patient (indirect)

Preconditions:

- The Healthcare Professional is currently logged in.
- The Healthcare Professional is linked to at least one user.

Postconditions:

- A new mandatory task is created and assigned to the selected patient.
- The assigned patient will be able to view the task and receive reminders.

Main Flow:

1. The Healthcare Professional navigates to the “Assigned Patients” page.
2. The system displays a list of patients linked to the Healthcare Professional.
3. The Healthcare Professional selects a patient from the list.
4. The system displays the selected patient’s profile summary, recent health data and tasks (incomplete and complete).
5. The Healthcare Professional selects “Assign New Task”.
6. The system displays a task creation form (e.g. task name, description, frequency, due date, target values).
7. The Healthcare Professional enters the task details and submits the form.
8. The system validates the input (required fields, valid ranges, valid dates).
9. The system stores the new task in the database.
10. The system confirms successful task creation (e.g. “Task successfully assigned to patient”).
11. The use case ends.

Alternative Flows:

2a. Healthcare Professional does not have any assigned patients.

1. If the Healthcare Professional does not have any assigned patients, the system will display a placeholder message (e.g. “No Current Assigned Patients”).
2. Use case ends.

6a. Healthcare Professional cancels task creation.

1. If the Healthcare Professional cancels, the system discards the entered data.
2. Use case ends.

8a. Invalid task input.

1. If the task details are invalid (missing fields, negative values, impossible dates), the system displays an error message specifying the issue.
2. No task is saved.
3. System returns to step 6 of the main flow.

8b. Unrealistic target or schedule warning.

1. If the task values are valid format but unrealistic (e.g. “Go for a 10,000-mile run”), the system warns the Healthcare Professional and asks for confirmation.
    a. If they confirm
        i. System proceeds to step 9 of the main flow.
    b. If they cancel
        i. System returns to step 6 of the main flow.

9a. Database error while saving task.

1. If the system fails to save the task (e.g. internal error, lost connection), the system displays an error message (“Task could not be assigned. Please try again”).
2. No task is saved.
3. System returns to step 6 of the main flow.

6.1.9 User Marks Mandatory Task or Personal Goal as Complete

Use Case ID: 9

Goal: Allow the Diabetic user to mark either a mandatory task (Healthcare Professional assigned) or a personal goal (Diabetic user created) as complete.

Primary Actor: Registered Diabetic User

Secondary Actor(s): Healthcare Professional (indirect – they can later view the completed status).

Preconditions:

- The Diabetic User is currently logged in.
- The Diabetic User has at least one active (incomplete) item assigned (mandatory task or personal goal).

Postconditions:

- The selected item is marked as complete in the system.
- Mandatory task completion becomes visible to the assigned Healthcare Professional.
- The dashboard will reflect updated completion status.

Main Flow:

1. The user navigates to the Activities page.
2. The system displays a list of all items (mandatory tasks / personal goals).
3. The user clicks the checkbox next to the item they want to mark as complete.
4. The system displays a confirmation message (e.g. “Well done! You completed …”).
5. The system updates the item’s status to “Completed” and reflects the change on the dashboard (Completed item are automatically archived after 24 hours).
6. The use case ends.

Alternative Flows:

3a. Undo Completion (Item already completed).

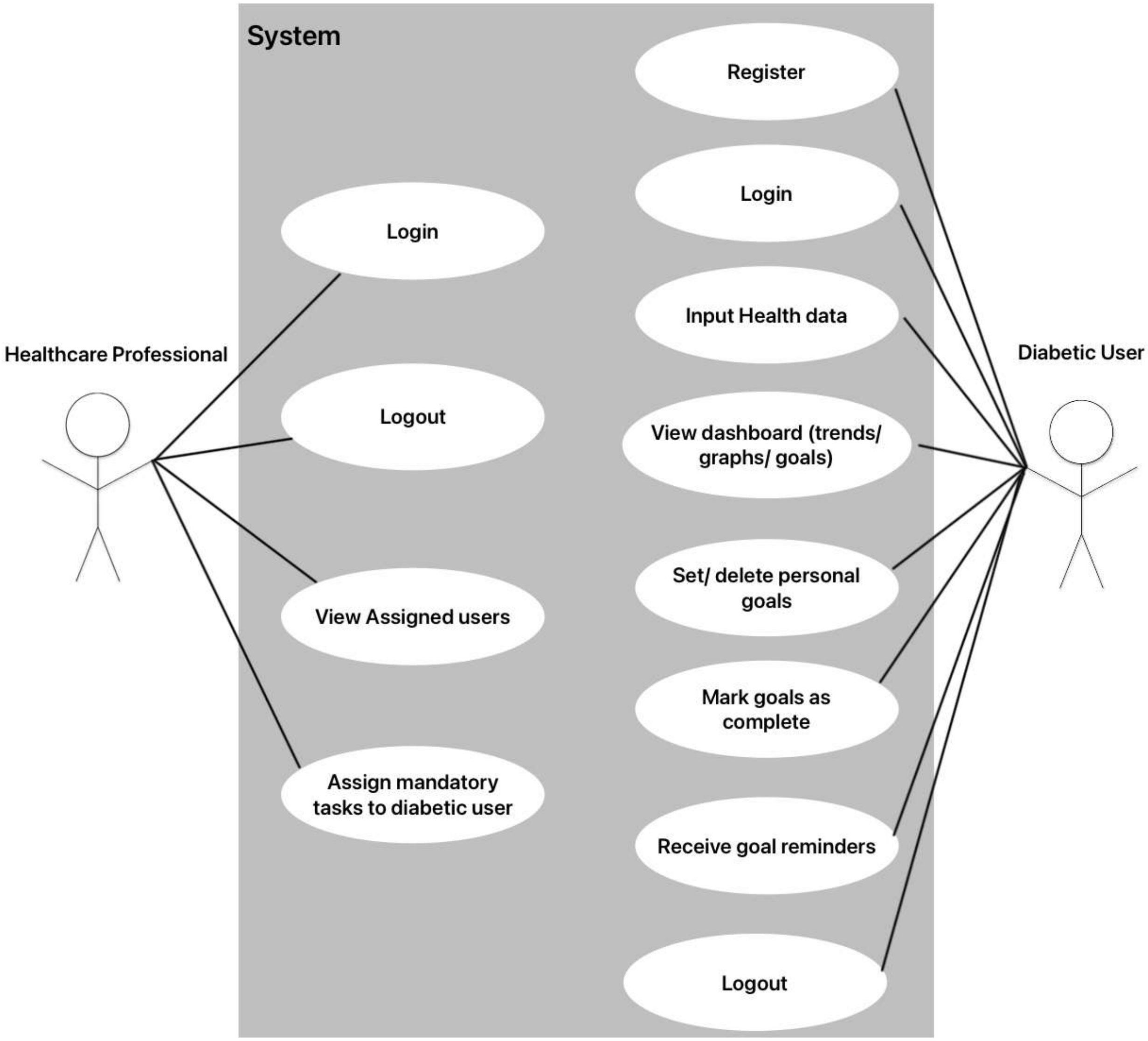
1. If the user clicks the checkbox of an item that was already complete, the system toggles the item back to “Incomplete” (The checkbox becomes unticked).
2. The system updates the dashboard to reflect the undo.
3. The use case ends

5a. Database Error When Updating Status.

1. If the system cannot update the task due to a storage or a network error, the system displays an error message (e.g. “Unable to update item. Please try again.”)
2. No changes are saved.
3. System returns back to step 2 of the main flow.

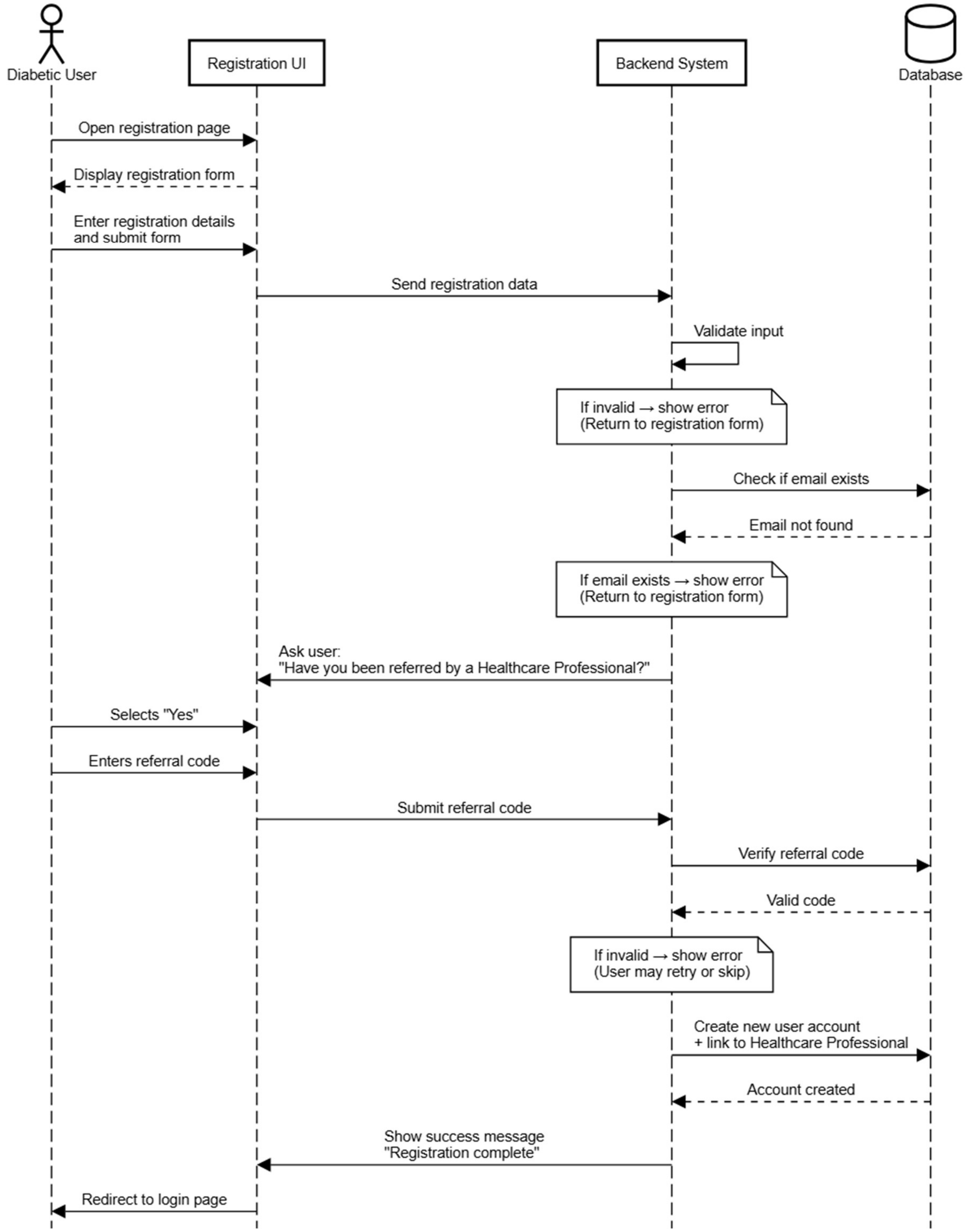
6.2 Diagrams

6.2.1 Use Case Diagram

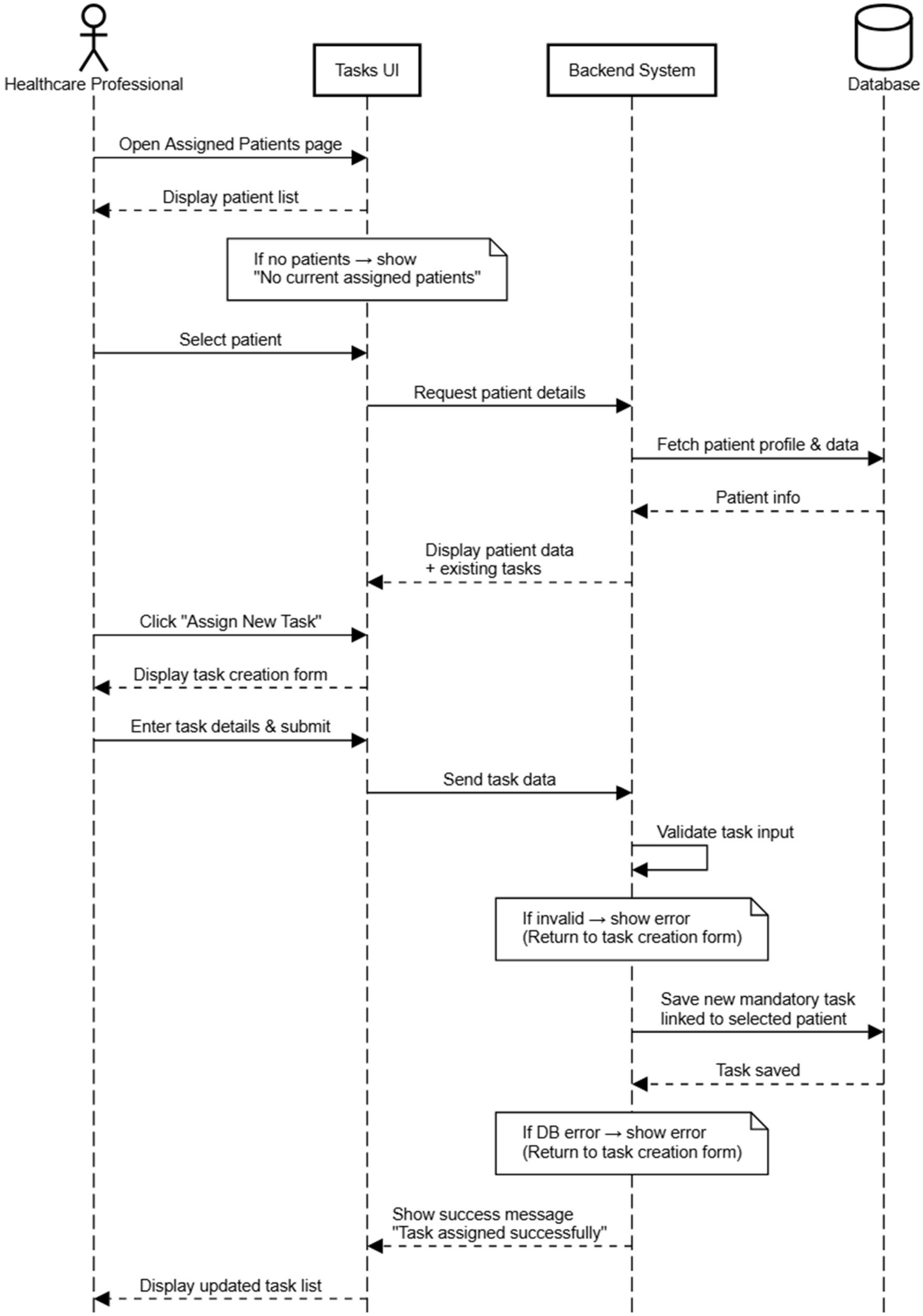


6.2.2 Sequence Diagrams

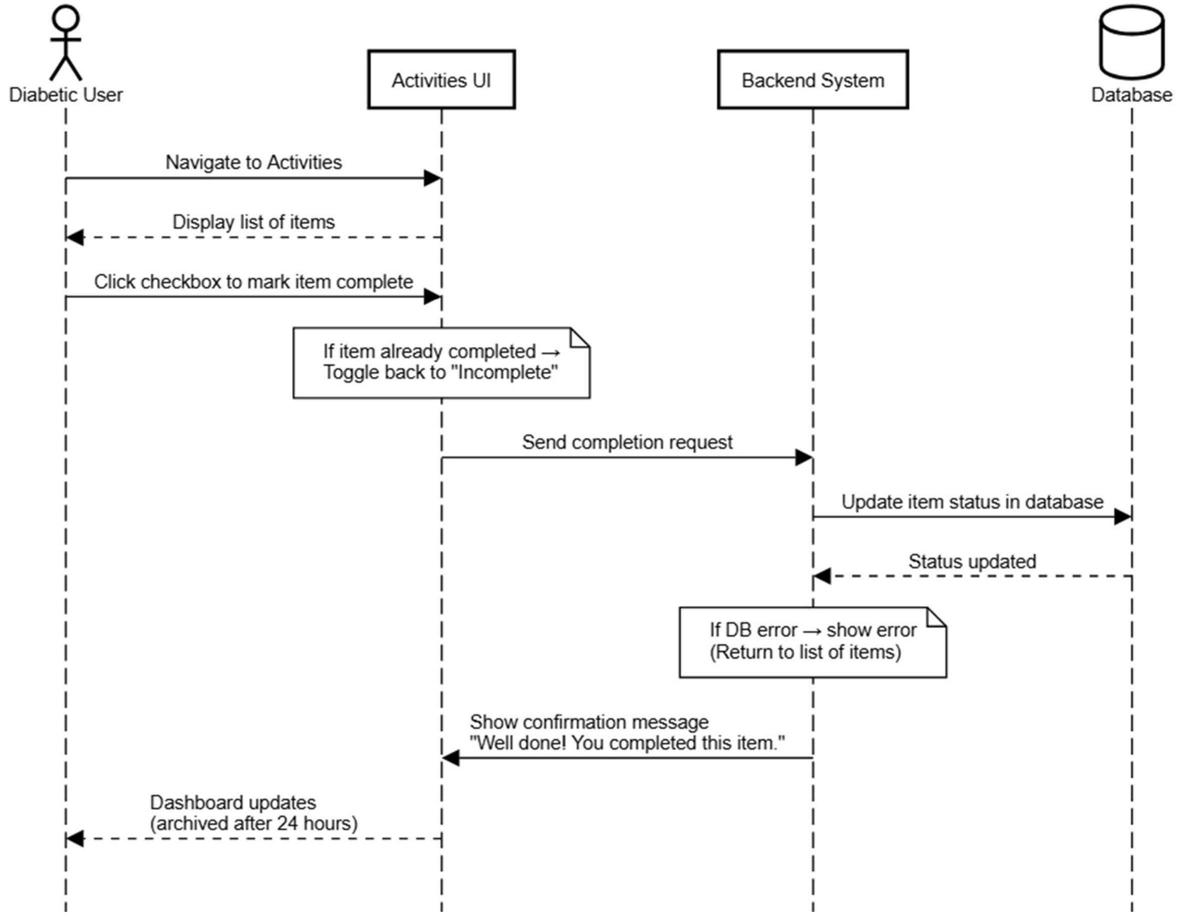
6.2.2.1 New User Registration



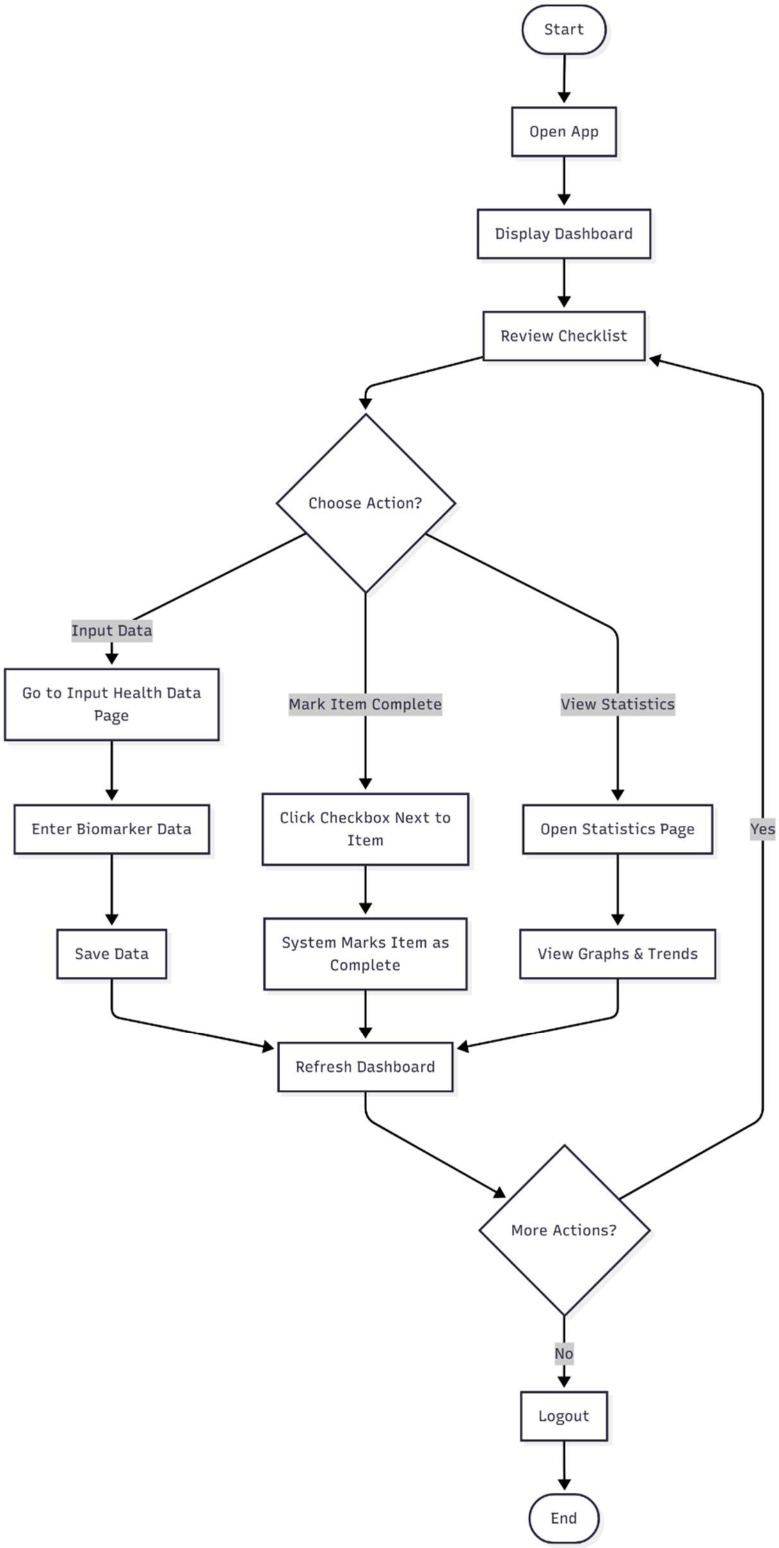
6.2.2.2 Healthcare Professional Assigns Mandatory Task



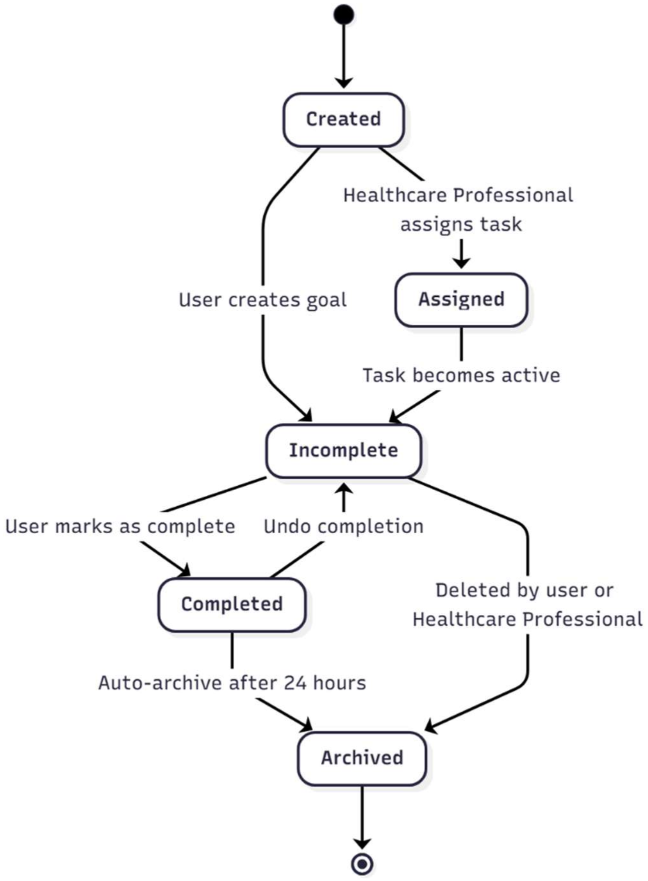
6.2.2.3 User Marks Mandatory Task or Personal Goal as Complete


Diabetic User
Activities UI
Backend System
Database
Navigate to Activities
Display list of items
Click checkbox to mark item complete
If item already completed →
Toggle back to "Incomplete"
Send completion request
Update item status in database
Status updated
If DB error → show error
(Return to list of items)
Show confirmation message
"Well done! You completed this item."
Dashboard updates
(archived after 24 hours)

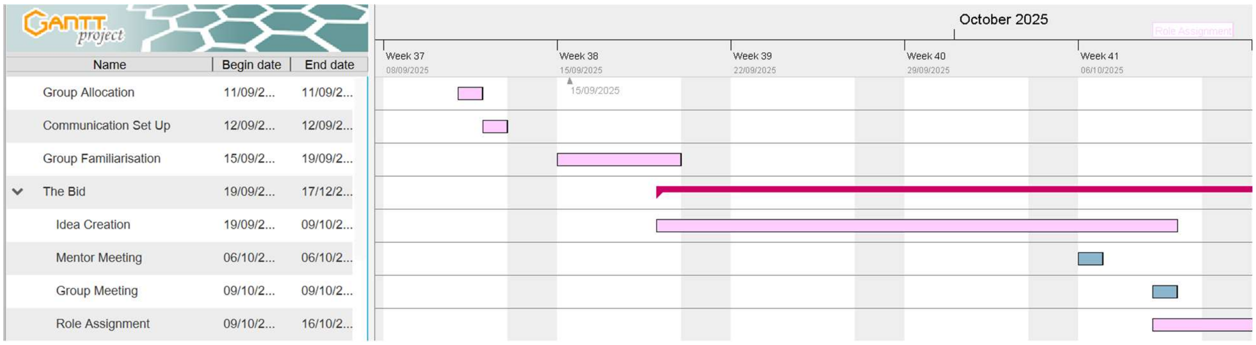
6.2.3 Activity Diagram: Diabetic User Daily Workflow



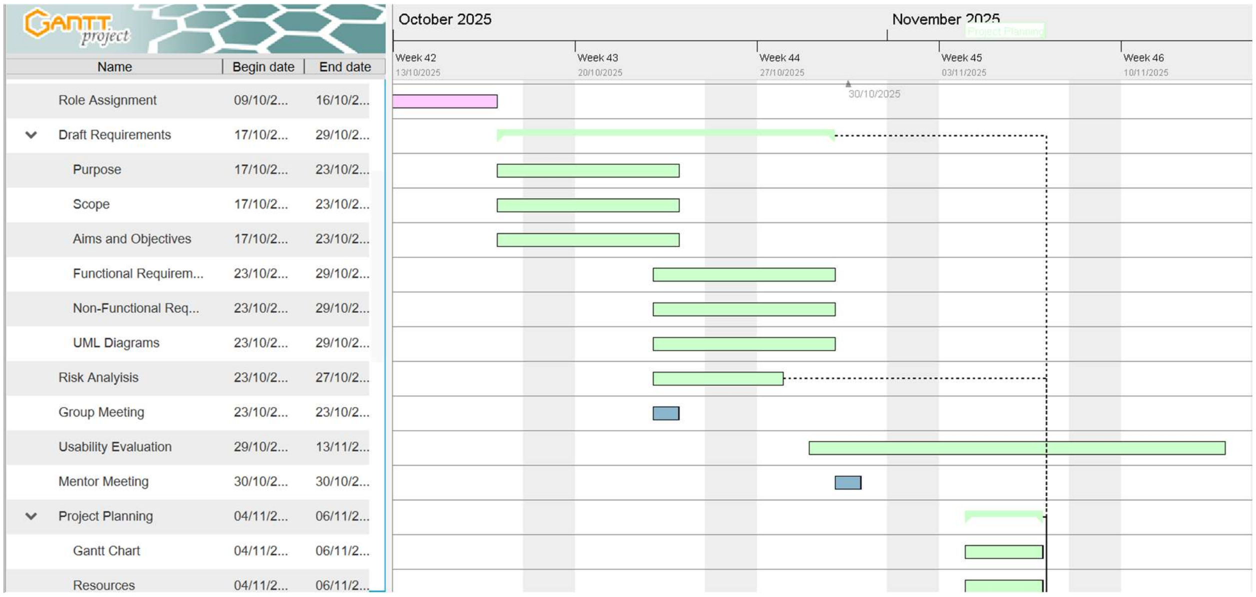
6.2.4 State Diagram: Lifecycle of a Mandatory Task or Personal Goal



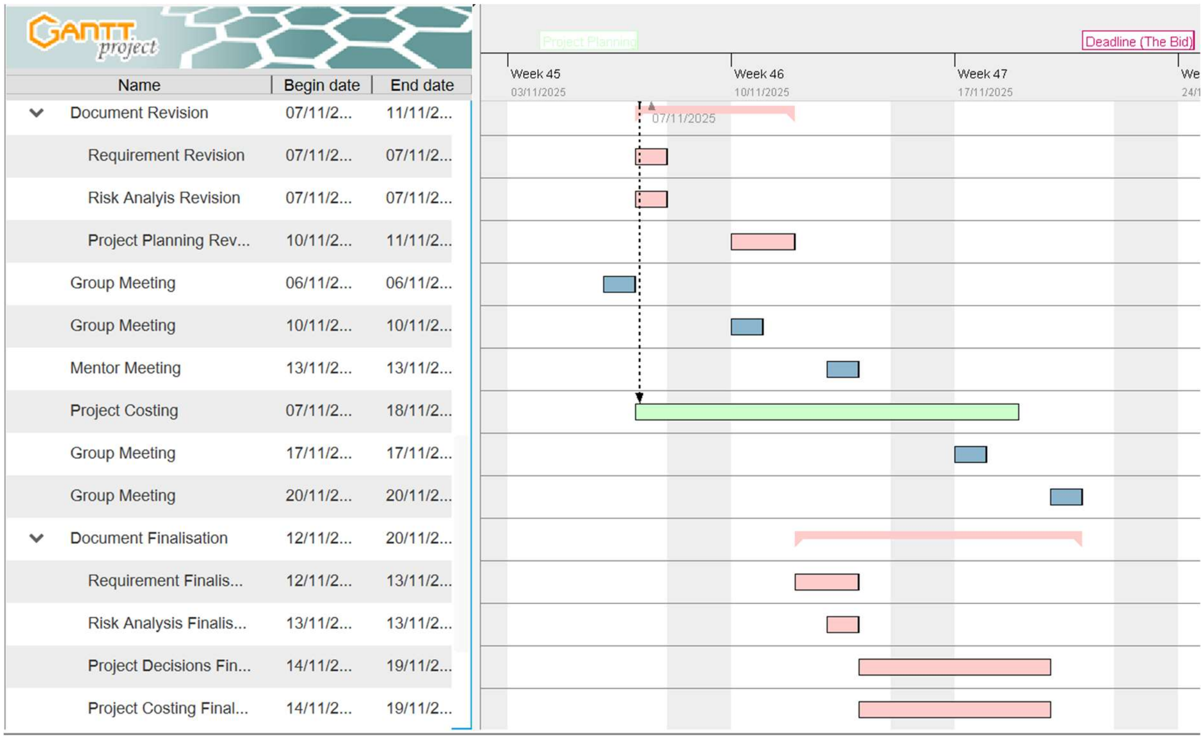
6.3 Stage 1 Tasks (Gantt Chart)



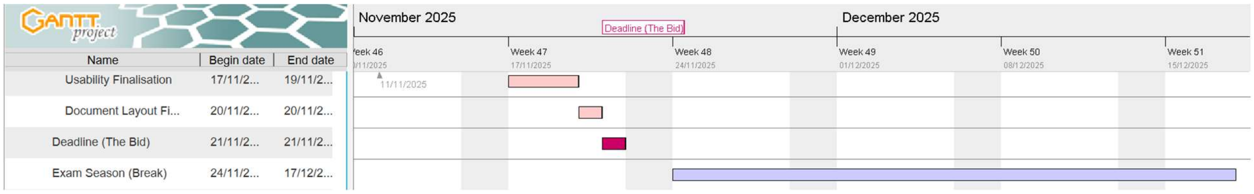
Name	Begin date	End date
Group Allocation	11/09/2...	11/09/2...
Communication Set Up	12/09/2...	12/09/2...
Group Familiarisation	15/09/2...	19/09/2...
The Bid	19/09/2...	17/12/2...
Idea Creation	19/09/2...	09/10/2...
Mentor Meeting	06/10/2...	06/10/2...
Group Meeting	09/10/2...	09/10/2...
Role Assignment	09/10/2...	16/10/2...



Name	Begin date	End date
Role Assignment	09/10/2...	16/10/2...
Draft Requirements	17/10/2...	29/10/2...
Purpose	17/10/2...	23/10/2...
Scope	17/10/2...	23/10/2...
Aims and Objectives	17/10/2...	23/10/2...
Functional Requirem...	23/10/2...	29/10/2...
Non-Functional Req...	23/10/2...	29/10/2...
UML Diagrams	23/10/2...	29/10/2...
Risk Analyisis	23/10/2...	27/10/2...
Group Meeting	23/10/2...	23/10/2...
Usability Evaluation	29/10/2...	13/11/2...
Mentor Meeting	30/10/2...	30/10/2...
Project Planning	04/11/2...	06/11/2...
Gantt Chart	04/11/2...	06/11/2...
Resources	04/11/2...	06/11/2...



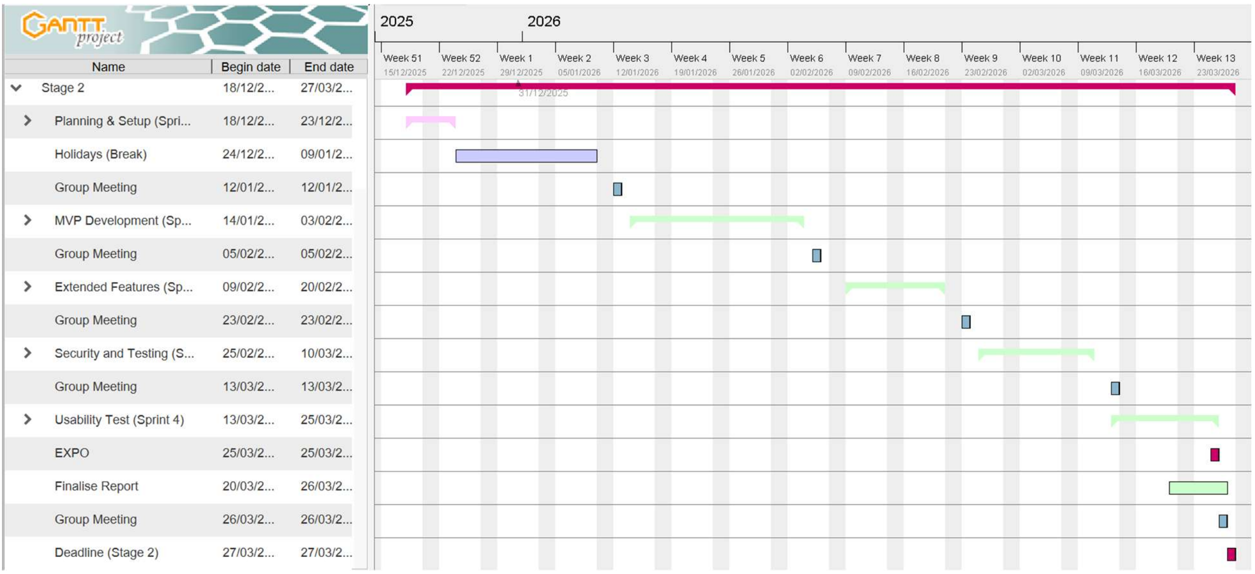
Name	Begin date	End date
Document Revision	07/11/2...	11/11/2...
Requirement Revision	07/11/2...	07/11/2...
Risk Analyis Revision	07/11/2...	07/11/2...
Project Planning Rev...	10/11/2...	11/11/2...
Group Meeting	06/11/2...	06/11/2...
Group Meeting	10/11/2...	10/11/2...
Mentor Meeting	13/11/2...	13/11/2...
Project Costing	07/11/2...	18/11/2...
Group Meeting	17/11/2...	17/11/2...
Group Meeting	20/11/2...	20/11/2...
Document Finalisation	12/11/2...	20/11/2...
Requirement Finalis...	12/11/2...	13/11/2...
Risk Analysis Finalis...	13/11/2...	13/11/2...
Project Decisions Fin...	14/11/2...	19/11/2...
Project Costing Final...	14/11/2...	19/11/2...



Name	Begin date	End date
Usability Finalisation	17/11/2...	19/11/2...
Document Layout Fi...	20/11/2...	20/11/2...
Deadline (The Bid)	21/11/2...	21/11/2...
Exam Season (Break)	24/11/2...	17/12/2...

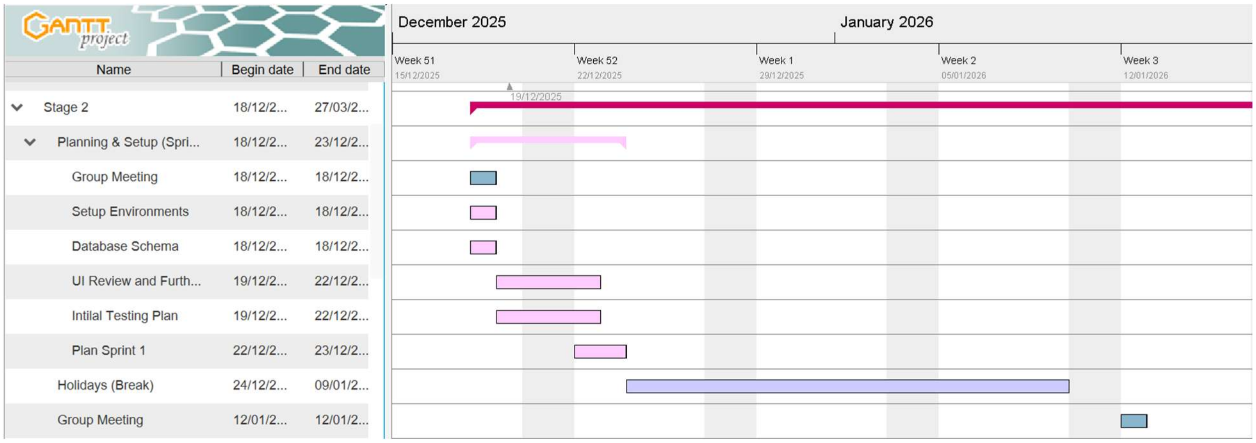
6.4 Tasks for Stage 1 (Gantt Chart)

6.4.1 Stage 2 (Overview)



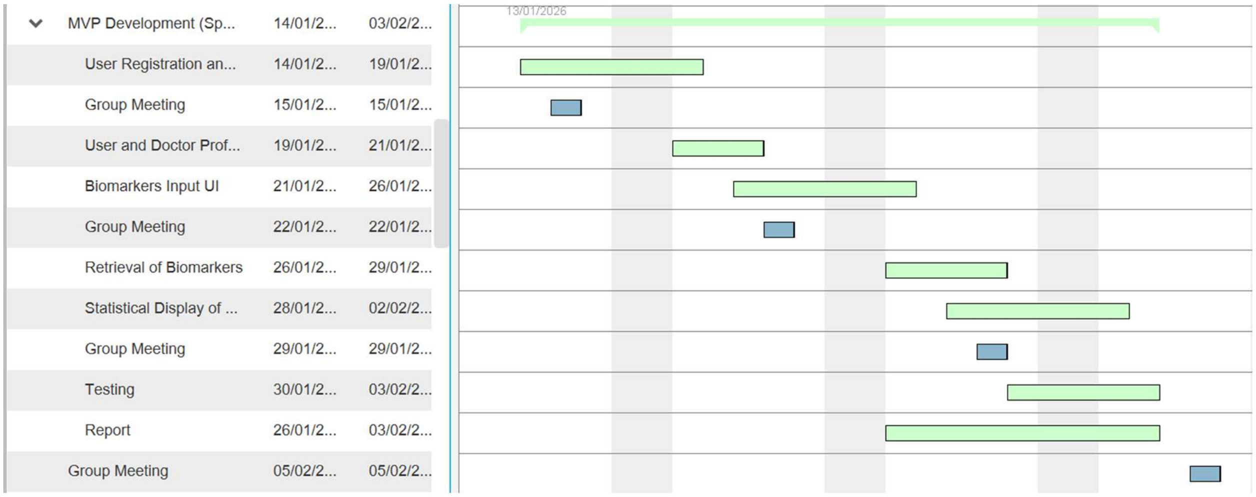
Name	Begin date	End date
Stage 2	18/12/2...	27/03/2...
Planning & Setup (Spri...	18/12/2...	23/12/2...
Holidays (Break)	24/12/2...	09/01/2...
Group Meeting	12/01/2...	12/01/2...
MVP Development (Sp...	14/01/2...	03/02/2...
Group Meeting	05/02/2...	05/02/2...
Extended Features (Sp...	09/02/2...	20/02/2...
Group Meeting	23/02/2...	23/02/2...
Security and Testing (S...	25/02/2...	10/03/2...
Group Meeting	13/03/2...	13/03/2...
Usability Test (Sprint 4)	13/03/2...	25/03/2...
EXPO	25/03/2...	25/03/2...
Finalise Report	20/03/2...	26/03/2...
Group Meeting	26/03/2...	26/03/2...
Deadline (Stage 2)	27/03/2...	27/03/2...

6.4.2 Sprint 0 – Planning and Setup



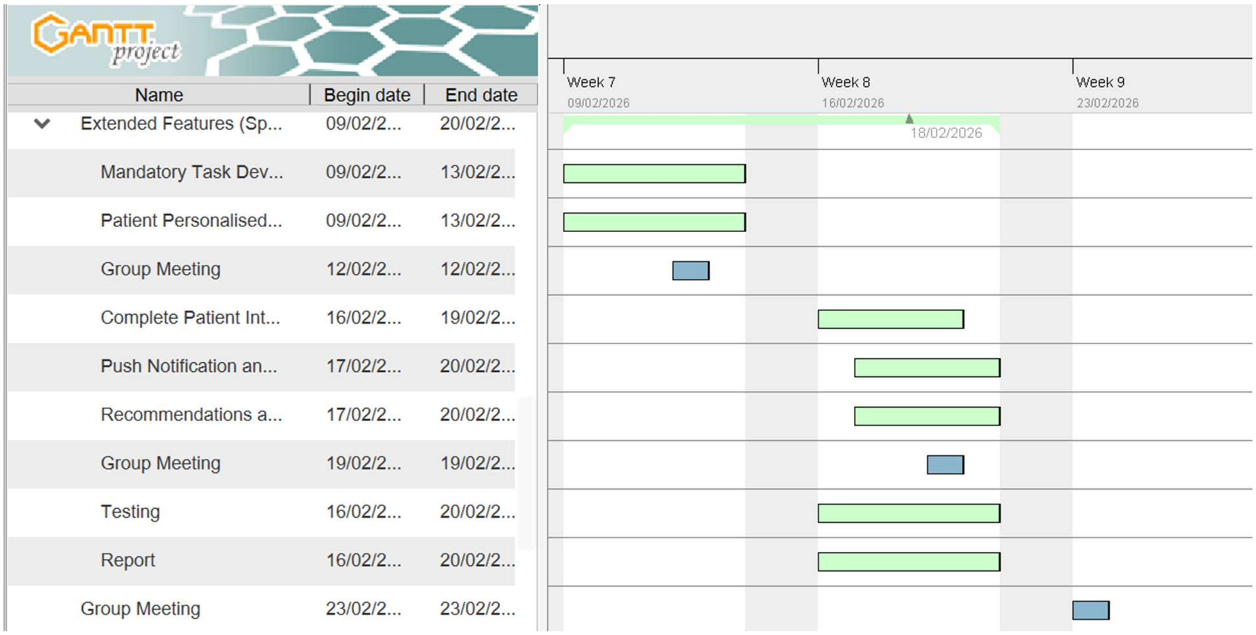
Name	Begin date	End date
Stage 2	18/12/2...	27/03/2...
Planning & Setup (Spri...	18/12/2...	23/12/2...
Group Meeting	18/12/2...	18/12/2...
Setup Environments	18/12/2...	18/12/2...
Database Schema	18/12/2...	18/12/2...
UI Review and Furth...	19/12/2...	22/12/2...
Intial Testing Plan	19/12/2...	22/12/2...
Plan Sprint 1	22/12/2...	23/12/2...
Holidays (Break)	24/12/2...	09/01/2...
Group Meeting	12/01/2...	12/01/2...

6.4.3 Sprint 1 – MVP Development



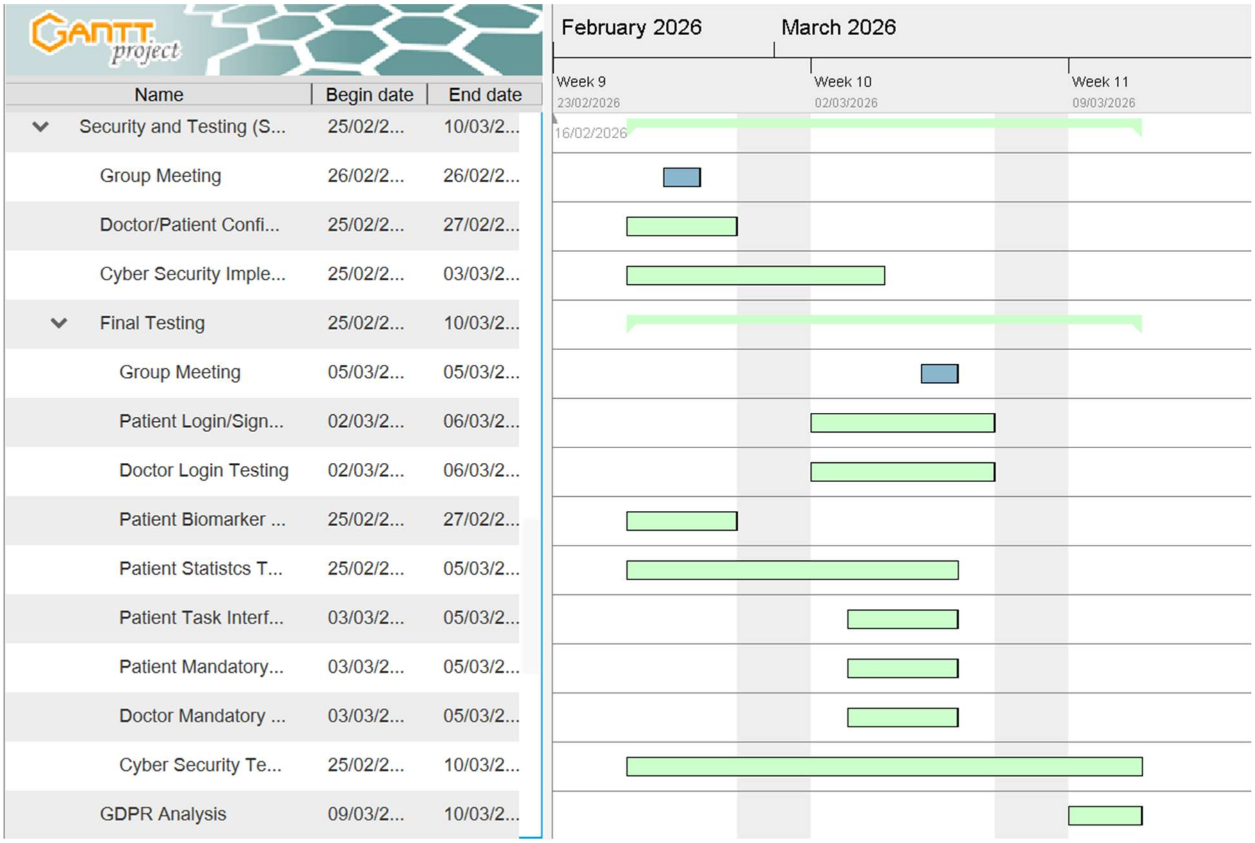
MVP Development (Sp...	14/01/2...	03/02/2...
User Registration an...	14/01/2...	19/01/2...
Group Meeting	15/01/2...	15/01/2...
User and Doctor Prof...	19/01/2...	21/01/2...
Biomarkers Input UI	21/01/2...	26/01/2...
Group Meeting	22/01/2...	22/01/2...
Retrieval of Biomarkers	26/01/2...	29/01/2...
Statistical Display of ...	28/01/2...	02/02/2...
Group Meeting	29/01/2...	29/01/2...
Testing	30/01/2...	03/02/2...
Report	26/01/2...	03/02/2...
Group Meeting	05/02/2...	05/02/2...

6.4.4 Sprint 2 – Extended Features



Name	Begin date	End date
Extended Features (Sp...	09/02/2...	20/02/2...
Mandatory Task Dev...	09/02/2...	13/02/2...
Patient Personalised...	09/02/2...	13/02/2...
Group Meeting	12/02/2...	12/02/2...
Complete Patient Int...	16/02/2...	19/02/2...
Push Notification an...	17/02/2...	20/02/2...
Recommendations a...	17/02/2...	20/02/2...
Group Meeting	19/02/2...	19/02/2...
Testing	16/02/2...	20/02/2...
Report	16/02/2...	20/02/2...
Group Meeting	23/02/2...	23/02/2...

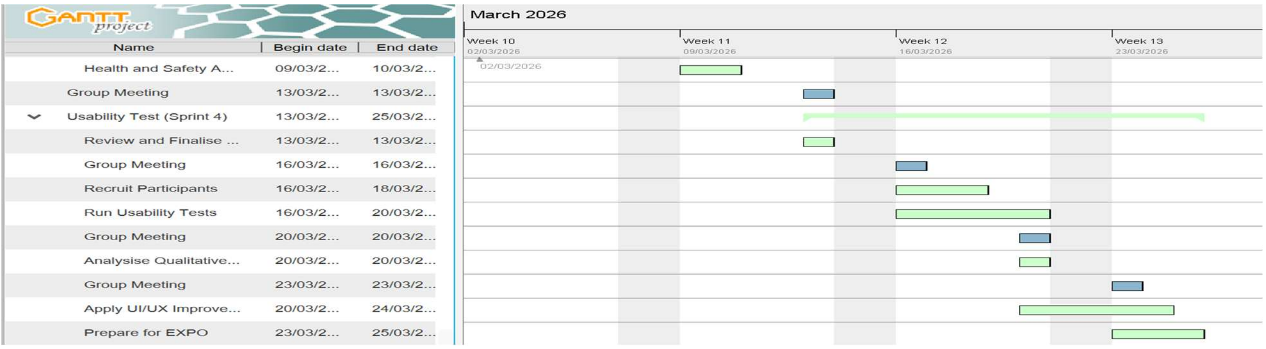
6.4.5 Sprint 3 – Security and Testing



Name	Begin date	End date
Security and Testing (S...	25/02/2...	10/03/2...
Group Meeting	26/02/2...	26/02/2...
Doctor/Patient Confi...	25/02/2...	27/02/2...
Cyber Security Imple...	25/02/2...	03/03/2...
Final Testing	25/02/2...	10/03/2...
Group Meeting	05/03/2...	05/03/2...
Patient Login/Sign...	02/03/2...	06/03/2...
Doctor Login Testing	02/03/2...	06/03/2...
Patient Biomarker ...	25/02/2...	27/02/2...
Patient Statistcs T...	25/02/2...	05/03/2...
Patient Task Interf...	03/03/2...	05/03/2...
Patient Mandatory...	03/03/2...	05/03/2...
Doctor Mandatory ...	03/03/2...	05/03/2...
Cyber Security Te...	25/02/2...	10/03/2...
GDPR Analysis	09/03/2...	10/03/2...

Name	Begin date	End date
Health and Safety A...	09/03/2...	10/03/2...
Group Meeting	13/03/2...	13/03/2...
Usability Test (Sprint 4)	13/03/2...	25/03/2...
Review and Finalise ...	13/03/2...	13/03/2...
Group Meeting	16/03/2...	16/03/2...
Recruit Participants	16/03/2...	18/03/2...
Run Usability Tests	16/03/2...	20/03/2...
Group Meeting	20/03/2...	20/03/2...
Analysise Qualitative...	20/03/2...	20/03/2...
Group Meeting	23/03/2...	23/03/2...
Apply UI/UX Improve...	20/03/2...	24/03/2...
Prepare for EXPO	23/03/2...	25/03/2...

6.4.6 Sprint 4 – Usability Test



Name	Begin date	End date
Health and Safety A...	09/03/2...	10/03/2...
Group Meeting	13/03/2...	13/03/2...
Usability Test (Sprint 4)	13/03/2...	25/03/2...
Review and Finalise ...	13/03/2...	13/03/2...
Group Meeting	16/03/2...	16/03/2...
Recruit Participants	16/03/2...	18/03/2...
Run Usability Tests	16/03/2...	20/03/2...
Group Meeting	20/03/2...	20/03/2...
Analysise Qualitative...	20/03/2...	20/03/2...
Group Meeting	23/03/2...	23/03/2...
Apply UI/UX Improve...	20/03/2...	24/03/2...
Prepare for EXPO	23/03/2...	25/03/2...

6.5 Gantt Chart Breakdown Tables

6.5.1 Stage 1

Task ID	**Task Name**	**Begin Date**	**End Date**	**Team Member**
1	Group Allocation	11/09/2025	11/09/2025	All
2	Communication Set Up	12/09/2025	12/09/2025	All
3	Group Familiarisation	15/09/2025	19/09/2025	All
4	**The Bid**	**19/09/2025**	**17/12/2025**	**All**
5	Idea Creation	19/09/2025	09/10/2025	All
6	Mentor Meeting	06/10/2025	06/10/2025	SC, ZN, AF, SG
7	Group Meeting	09/10/2025	09/10/2025	SC, AF, HN
8	Role Assignment	09/10/2025	16/10/2025	All
9	Draft Requirements	17/10/2025	22/10/2025	IS, SG
10	Purpose	17/10/2025	23/10/2025	IS
11	Scope	17/10/2025	23/10/2025	IS
12	Aims and Objectives	17/10/2025	23/10/2025	IS
13	Functional Requirements	23/10/2025	29/10/2025	IS
14	Non-Functional Requirements	23/10/2025	29/10/2025	IS
15	UML Diagrams	23/10/2025	29/10/2025	SG
16	Risk Analysis	23/10/2025	27/10/2025	SC
17	Group Meeting	23/10/2025	23/10/2025	All
18	Usability Evaluation	29/10/2025	13/11/2025	AF
19	Mentor Meeting	30/10/2025	30/10/2025	All
20	Project Planning	04/11/2025	06/11/2025	SC
21	Gantt Chart	04/11/2025	06/11/2025	SC
22	Resources	04/11/2025	06/11/2025	SC
23	Document Revision	07/11/2025	11/11/2025	All
23.1	Requirement Revision	07/11/2025	07/11/2025	All
23.2	Risk Analysis Revision	07/11/2025	07/11/2025	All

23.3	Project Planning Revision	09/11/2025	11/11/2025	All
24	Group Meeting	06/11/2025	06/11/2025	All
25	Group Meeting	10/11/2025	10/11/2025	All
26	Mentor Meeting	13/11/2025	13/11/2025	All
27	Project Costing	07/11/2025	18/11/2025	ZN
28	Group Meeting	17/11/2025	17/11/2025	All
29	Group Meeting	20/11/2025	20/11/2025	All
30	Document Finalisation	12/11/2025	20/11/2025	All
30.1	Requirement Finalisation	12/11/2025	13/11/2025	All
30.2	Risk Analysis Finalisation	13/11/2025	13/11/2025	All
30.3	Project Decisions Finalisation	14/11/2025	19/11/2025	All
30.4	Project Costing Finalisation	14/11/2025	19/11/2025	All
30.5	Usability Finalisation	17/11/2025	19/11/2025	All
30.6	Document Layout Finalisation	20/11/2025	20/11/2025	All
31	**Deadline (The Bid)**	**21/11/2025**	**21/11/2025**	**All**
32	Exam Season (Break)	24/11/2025	17/12/2025	All

6.5.2 Stage 2

Task ID	Task Name	Begin Date	End Date	Team Members
1	Planning & Setup (Sprint 0)	18/12/2025	23/12/2025	All
1.1	Group Meeting	18/12/2025	18/12/2025	All
1.2	Setup Environment	18/12/2025	18/12/2025	HN
1.3	Database Schema	18/12/2025	18/12/2025	All

1.4	UI Review and Further Mockups	19/12/2025	22/12/2025	All
1.5	Initial Test Plan	19/12/2025	22/12/2025	All
1.6	Plan Sprint 1	22/12/2025	23/12/2025	All
2	Holidays (Break)	24/12/2025	09/01/2026	All
3	Group Meeting	12/01/2025	12/01/2025	All
4	**MVP Development (Sprint 1)**	**14/01/2026**	**31/01/2026**	**All**
4.1	User Registration/Log in	14/01/2026	19/01/2026	SC, SG
4.2	Group Meeting	15/01/2025	15/01/2025	All
4.3	User and Doctor Profiles	19/01/2026	21/01/2026	ZN
4.4	Biomarker Input UI	21/01/2026	26/01/2026	IS, ZN
4.5	Group Meeting	22/01/2025	22/01/2025	All
4.6	Retrieval of Biomarkers	26/01/2026	29/01/2026	HN
4.7	Statistical Displays of Biomarkers	28/01/2026	02/02/2026	IS, HN
4.8	Group Meeting	29/01/2026	29/01/2026	All
4.9	Testing	30/01/2026	03/02/2026	AF, SG
4.10	Report	26/01/2026	03/02/2026	SC
5	Group Meeting	05/02/2026	05/02/2026	All

6	**Extended Features (Sprint 2)**	**09/02/2026**	**20/02/2026**	All
6.1	Mandatory Task Development	09/02/2026	13/02/2026	HN
6.2	Patient Personalised Tasks and Interface	09/02/2026	13/02/2026	ZN
6.3	Group Meeting	12/02/2026	12/02/2026	All
6.4	Complete Patient Interface with Comparison Statistics	16/02/2026	19/02/2026	HN, SC
6.5	Push Notifications and Reminders	17/02/2026	20/02/2026	IS
6.6	Recommendations and Warnings	17/02/2026	20/02/2026	IS
6.7	Group Meeting	19/02/2026	19/02/2026	All
6.8	Testing	16/02/2026	20/02/2026	SG, AF
6.9	Report	16/02/2026	20/02/2026	SC
7	Group Meeting	23/02/2026	23/02/2026	All
8	**Security and Testing (Sprint 3)**	**25/02/2026**	**17/03/2026**	All
8.1	Group Meeting	26/02/2026	26/02/2026	All
8.2	Doctor/Patient Confidentiality Analysis and Implementation	25/02/2026	27/02/2026	ZN

8.3	Cyber Security Implementation	25/02/2026	03/03/2026	AF
8.4	Final Testing	25/02/2026	10/03/2026	All
8.5	Group Meeting	05/03/2026	05/03/2026	All
8.6	Patient Login/Sign-up Testing	02/03/2026	06/03/2026	SC
8.7	Doctor Login Testing	02/03/2026	06/03/2026	IS
8.8	Patient Biomarker Input Testing	25/02/2026	27/02/2026	IS, ZN
8.9	Patient Statistics Testing	25/02/2026	05/03/2026	ZN
8.10	Patient Task Interface Testing	03/03/2026	05/03/2026	SG
8.11	Patient Mandatory Task Testing	03/03/2026	05/03/2026	SG, HN
8.12	Doctor Mandatory Task Testing	10/03/2026	12/03/2026	SG, HN
8.13	Cyber Security Testing	25/02/2026	10/03/2026	AF
8.14	GDPR Analysis	09/03/2026	09/03/2026	AF
8.15	Health and Safety Analysis	09/03/2026	09/03/2026	SC
9	Group Meeting	13/03/2026	13/03/2026	All
10	**Usability Test (Sprint 4)**	**13/03/2026**	**25/03/2026**	All

10.1	Review and Finalise Usability Test	13/03/2026	13/03/2026	All
10.2	Group Meeting	16/06/2026	16/06/2026	All
10.3	Recruit Participants	16/03/2026	18/03/2026	All
10.4	Run Usability Tests	16/03/2026	20/03/2026	SG
10.5	Group Meeting	20/03/2026	20/03/2026	All
10.6	Analyse Quantitative and Qualitative Results	20/03/2026	20/03/2026	SC
10.7	Group Meeting	23/03/2025	23/03/2026	All
10.8	Apply UI/UX	20/03/2026	24/03/2026	IS, ZN, HN
10.9	Prepare for EXPO	20/03/2026	24/03/2026	All
11	**EXPO**	**25/03/2026**	**25/03/2026**	All
12	Finalise Report	20/03/2025	26/03/2025	All
13	Group Meeting	26/03/2026	26/03/2026	All
14	**Deadline (Stage 2)**	**27/03/2026**	**27/03/2026**	All

6.6 Resources Chart

Team Member	Individual Tasks Assigned		Weekly Average	
	Stage 1	**Stage 2**	**Stage 1**	**Stage 2**
Samara Cunningham	6, 7, 20, 21, 22	4.1, 4.10, 6.4, 6.9, 8.6, 10.6	7	10

Alex Forsyth	6, 7, 18	4.9, 6.8, 8.3, 8.13, 8.14	6.5	10.5
Suprita	6, 9, 15	4.1, 4.9, 6.8, 8.10, 8.11, 8.12, 10.4	6	10.5
Zain	6, 27	4.3, 4.4., 4.7, 8.2, 8.8, 8.9, 10.8	6	10.5
Henry Ni	7	1.2, 3.6, 4.7, 6.1, 6.4, 8.11, 8.12, 10.8	4.5	11.5
Ismail Soleiman	9, 10, 11, 12, 13, 14	4.4, 4.7, 6.5, 6.6, 8.7, 8.8, 10.8	7	9.5

6.7 Total Cost Tables

6.7.1 Cost Calculation Overview

No	Items	Expected Cost (£)
1	People	79,500
2	Equipment	3,600
3	Training	3,300
4	Testing	2,600
5	Support & Maintenance	8,200
6	Reserves (15%)	14,580
	Total Expected Cost	111,780

6.7.2 Cost Calculation – People

No	Items	Hours	Units	Cost per unit hour (£)	Expected Cost (£)
1	People				
(a)	Technical Manager (Henry)	300	1	45	13,500
(b)	Organisational Manager (Ismail)	300	1	42	12,600
(c)	Full-stack Developers (Ismail & Zain)	300	2	38	22,800

(d)	Front-end Developers (Samara & Suprita)	300	2	35	21,000
(e)	Tester (Alex)	300	1	32	9,600
	Total People				79,500

6.7.3 Cost Calculation – Equipment

No	Items	Units	Type	Cost per unit (£)	Expected Cost (£)
2	Equipment				
(a)	Hosting Server / VM	1	Cloud-hosted	1,200	1,200
(b)	Development Hardware	6	Developer PCs	400	2,400
(c)	Development Tools	–	Open Source	0	0
	Total Equipment				3,600

6.7.4 Cost Calculation – Training

No	Items	Units	Cost per unit (£)	Expected Cost (£)
3	Training			
(a)	Security / Data Protection Training	6	150	900
(b)	Development Tool Training	6	400	2,400
	Total Training			3,300

6.7.5 Cost Calculation – Testing

No	Items	Units	Cost per unit (£)	Expected Cost (£)
4	Testing			

(a)	Software Functional Testing	1	600	600
(b)	Security Testing	1	1,200	1,200
(c)	Usability / Integration Testing	1	800	800
	Total Testing			2,600

6.7.6 Cost Calculation – Support & Maintenance

No	Items	Duration / Units	Cost per service (£)	Expected Cost (£)
5	Support & Maintenance			
(a)	Software Updates	Once per 3 months (1 year, 4 updates)	800	3,200
(b)	Software Enhancements	Enhancement budget	5,000	5,000
	Total Support & Maintenance			8,200

6.8 Cost Description Table

Member	Time Spent on Main Role	Time Spent on Sub Role
Henry	100% as Technical Manager	50% as Back-end Developer
Ismail	100% as Organisational Manager	50% as Full-stack Developer
Zain	100% as Full-stack Developer	50% as Financial Coordinator / Reporting
Samara	100% as Front-end Developer	50% as Reporter / Documentation
Suprita	100% as Front-end Developer	50% as Liaison with stakeholders
Alex	100% as Tester	50% as Back-end Developer

6.8 Sample Blank Consent Form

F29SO Consent Form for Usability Experiments

DIABETES TRACKER/GROUP 11

Heriot-Watt University, Computer Science Department

Consent to Act as a Participant in an Experimental Study

Principal Investigators: Alex Forysth (af2065@hw.ac.uk), Henry Ni, Ismail Soleiman, Samara Cunningham, Suprita Gajulapalle and Zain Nawazani.

Description: The purpose of this project is to study the usability of a diabetes tracker website application.

There are minimal risks for you to participate in this study. All personal information will be kept confidential in a secure filling cabinet or in password-protected computer directories. Your participation will not affect how well you do in your courses (if you are a student) of affect your relationship with the university in any way.

You are free to decline to participate in this study. Should you decide to participate, you are free to end your participation at any time. Such a decision by you will not adversely affect or alter your status with the university in any way.

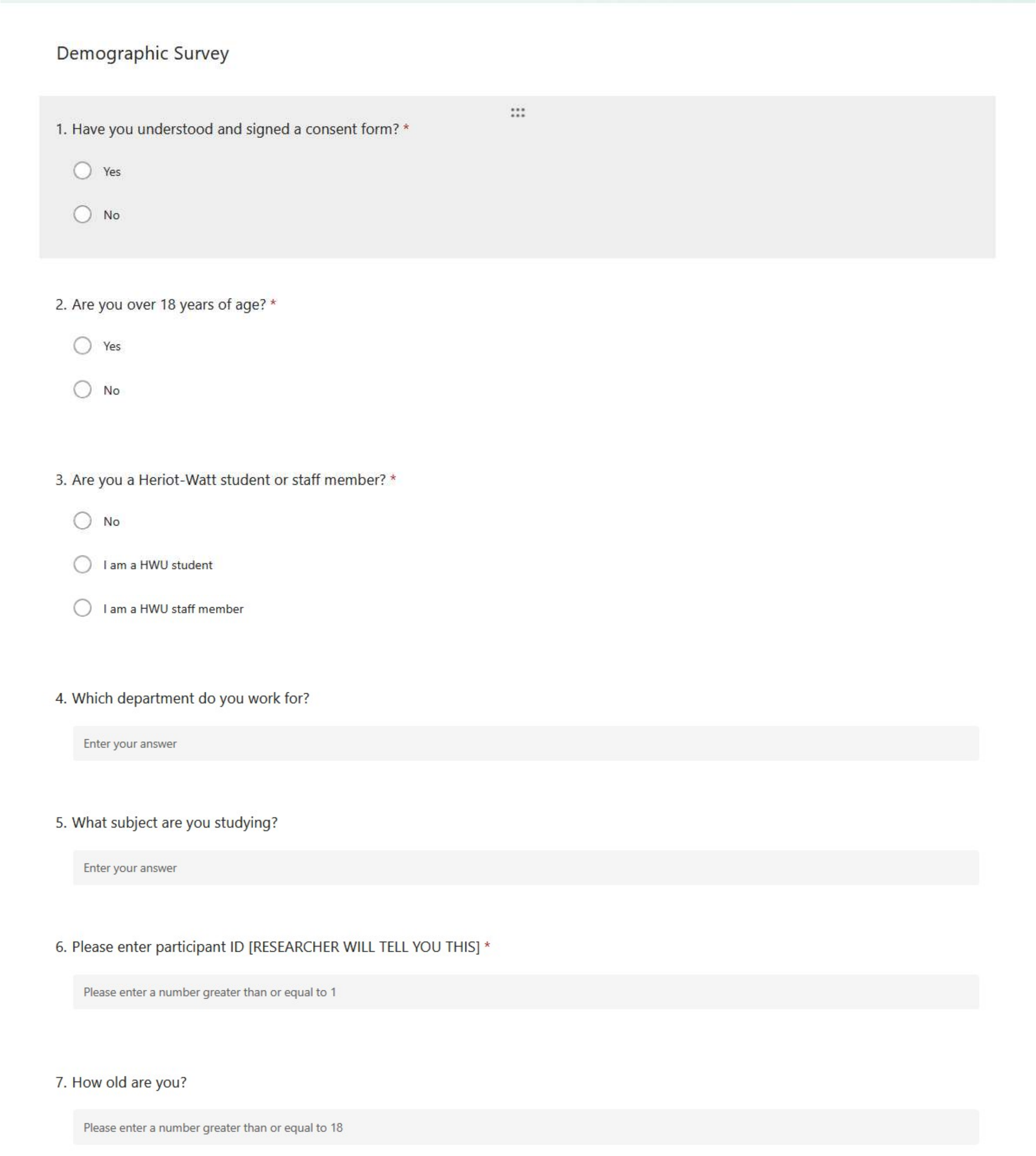
Voluntary consent: I confirm that I have read the preceding and that I understand its contents. I confirm that I have no known impediment that might affect my ability to provide consent and participate in the study. Any questions I have pertaining to the research have been and will be answered by the team. My signature below means that I have freely agreed to participate in this study, and that I agree to the publication of the results for scientific purposes and to the distribution of the recordings and transcripts of the sessions for research purposes so long as my identity is not revealed.

DD/MM/YYYY	FULL NAME	INITIALS
__________	________________________	____________
Date	Participant Signature	Indiv. Initials

Investigators certification: I certify that I have explained to the above individual the nature and purpose, the potential benefits, and possible risks associated with participation in this research study, have answered any questions that have been raised, and have witnessed the above signature.

DD/MM/YYYY	FULL NAME	INITIALS
__________	________________________	____________
Date	Investigator Signature	Indiv. Initials

6.9 Blank Questionnaire Survey



Demographic Survey

1. Have you understood and signed a consent form? *

○ Yes

○ No

2. Are you over 18 years of age? *

○ Yes

○ No

3. Are you a Heriot-Watt student or staff member? *

○ No

○ I am a HWU student

○ I am a HWU staff member

4. Which department do you work for?

Enter your answer

5. What subject are you studying?

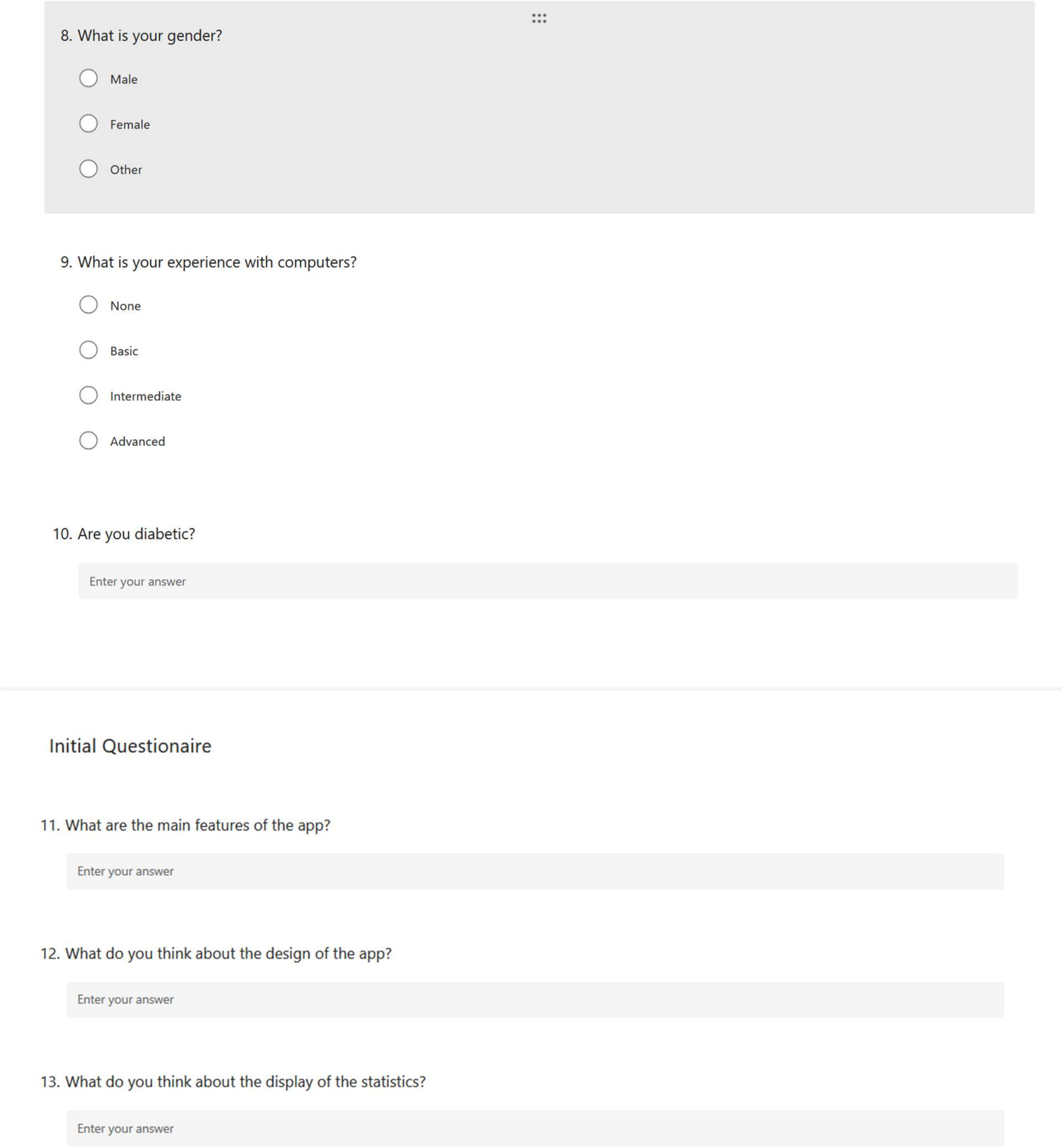
Enter your answer

6. Please enter participant ID [RESEARCHER WILL TELL YOU THIS] *

Please enter a number greater than or equal to 1

7. How old are you?

Please enter a number greater than or equal to 18



8. What is your gender?

○ Male

○ Female

○ Other

9. What is your experience with computers?

○ None

○ Basic

○ Intermediate

○ Advanced

10. Are you diabetic?

Enter your answer

Initial Questionaire

11. What are the main features of the app?

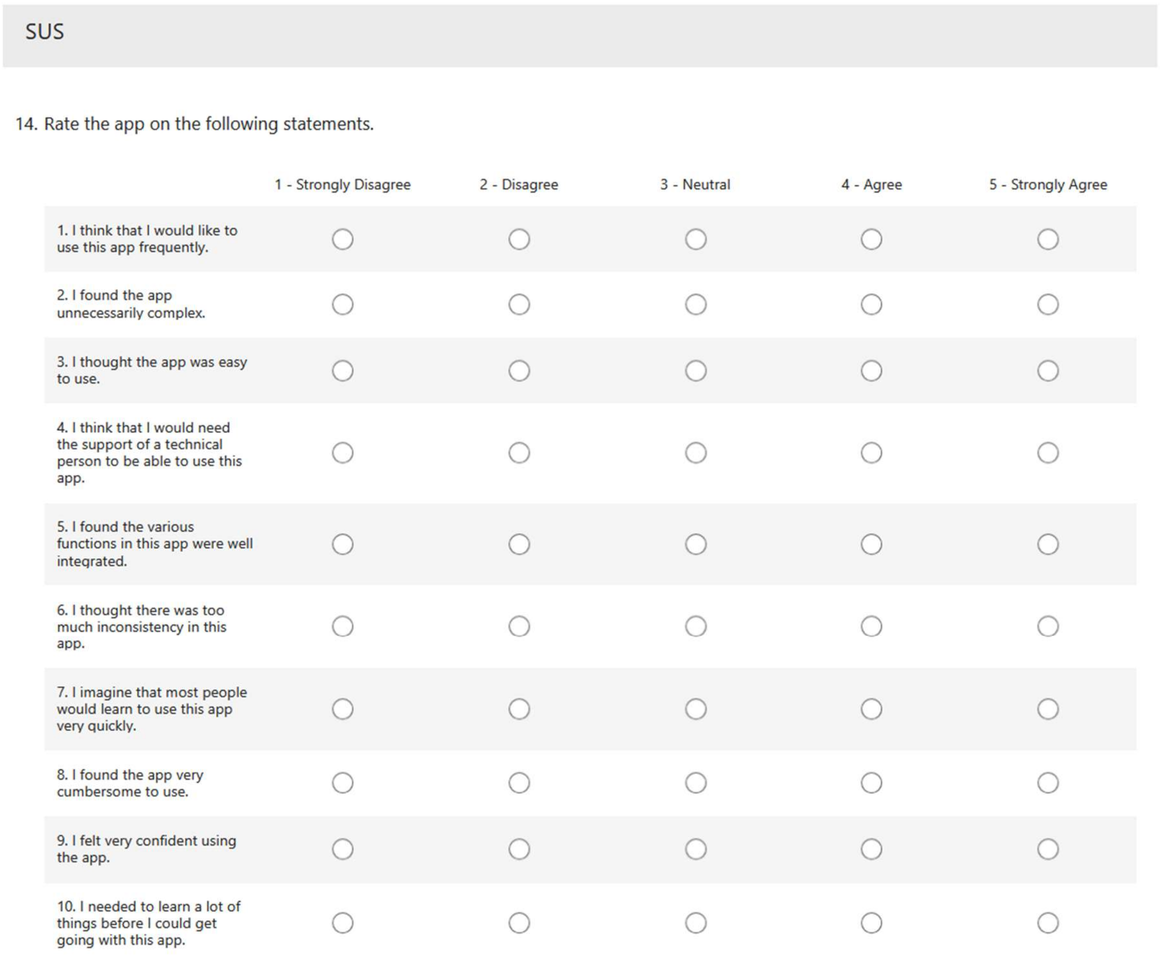
Enter your answer

12. What do you think about the design of the app?

Enter your answer

13. What do you think about the display of the statistics?

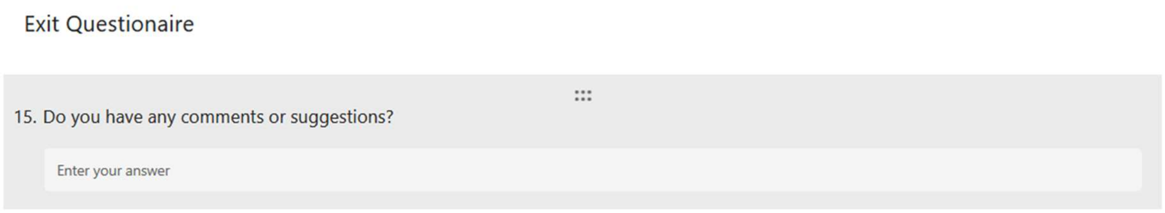
Enter your answer



SUS

14. Rate the app on the following statements.

	1 - Strongly Disagree	2 - Disagree	3 - Neutral	4 - Agree	5 - Strongly Agree
1. I think that I would like to use this app frequently.	○	○	○	○	○
2. I found the app unnecessarily complex.	○	○	○	○	○
3. I thought the app was easy to use.	○	○	○	○	○
4. I think that I would need the support of a technical person to be able to use this app.	○	○	○	○	○
5. I found the various functions in this app were well integrated.	○	○	○	○	○
6. I thought there was too much inconsistency in this app.	○	○	○	○	○
7. I imagine that most people would learn to use this app very quickly.	○	○	○	○	○
8. I found the app very cumbersome to use.	○	○	○	○	○
9. I felt very confident using the app.	○	○	○	○	○
10. I needed to learn a lot of things before I could get going with this app.	○	○	○	○	○



Exit Questionaire

15. Do you have any comments or suggestions?

Enter your answer

6.10 Testing Protocol

Risk Assessment

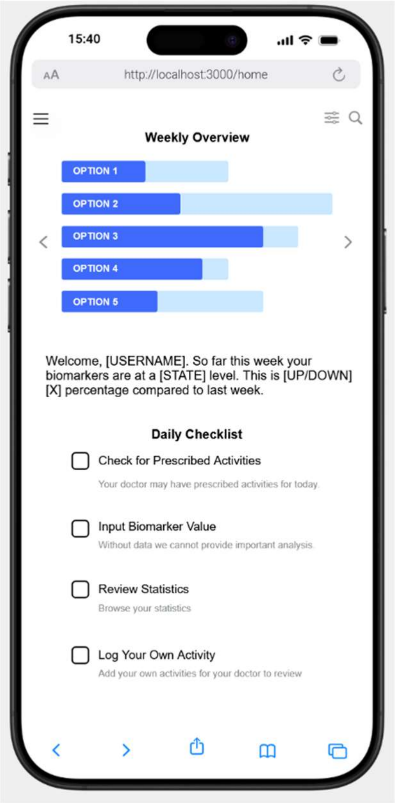
Risk Type	**Risk Level**	**Risk Mitigations**
Data Protection and Privacy	Medium	All personal data is being held in accordance with GDPR regulations.
Informed Consent	Medium	Inform participants of their right to withdraw from the study at any time and they can skip questions.
Psychological or Emotional Risks	Low	Inform participants that their answers and feedback is valued and there is no right or wrong answers.
Physical and Environmental Risks	Low	Eye strain from laptop screen and discomfort from sitting down are minimised by keeping duration of interview short.
Safeguarding	Low	Ensure participant is not a vulnerable user and they meet the ethics guidelines.

Testing Protocol

1. Inform participant of potential risk involved in the study. Consent form covers a lot of this.
2. Obtain informed consent from the participant. Ensure you have both signed consent form and it is saved.
3. Ask the participant to not talk to anyone outside of the study because we only want their own answers and opinions. Let them know there are no right or wrong answers and all feedback is valuable.
4. Assign a unique participant ID.
5. Add participant ID to consent form link to encrypted master sheet. Ensure this is saved.
6. Get participant to complete the demographic survey. Input their participant ID for them. Remind them they can leave any questions blank if they wish and the app is being tested, not them.
7. Read the following to the participant: “I am going to give you a prototype of a diabetes tracker application. I want you to explore the prototype for the next 3 minutes. You can navigate to several pages using the prototype. You will then be asked some questions regarding usability. Please answer them to the best of your abilities.”
8. After 3 minutes take away the prototype.
9. Get the participant to complete the initial questionnaire, the SUS and the exit questionnaire.
10. Thank the participant for their time and cooperation. Ensure the form is submitted.

6. 11 Prototypes

6.11.1 Home Page



Welcome, [USERNAME]. So far this week your biomarkers are at a [STATE] level. This is [UP/DOWN] [X] percentage compared to last week.

Daily Checklist

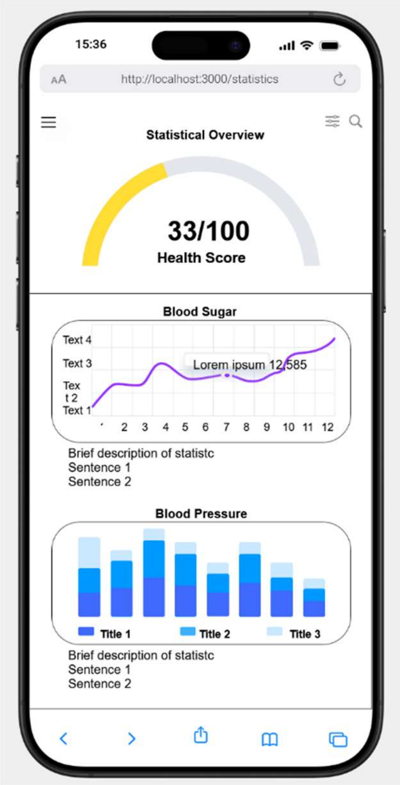
□ Check for Prescribed Activities
Your doctor may have prescribed activities for today.

□ Input Biomarker Value
Without data we cannot provide important analysis.

□ Review Statistics
Browse your statistics

□ Log Your Own Activity
Add your own activities for your doctor to review

6.11.2 Statistics Page

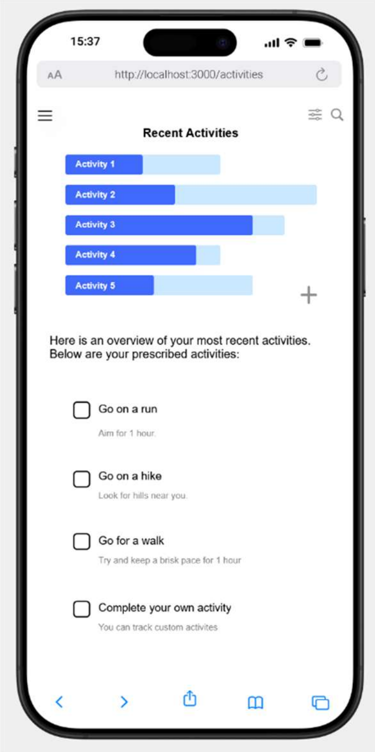


Brief description of statistc
Sentence 1
Sentence 2

Brief description of statistc
Sentence 1
Sentence 2

x

6.11.3 Activities



Recent Activities

Here is an overview of your most recent activities.
Below are your prescribed activities:

□ Go on a run
Aim for 1 hour.

□ Go on a hike
Look for hills near you.

□ Go for a walk
Try and keep a brisk pace for 1 hour

□ Complete your own activity
You can track custom activites