Bookit (Room Booking Web App - Assignment 2 (Project Proposal))

[22/12/2023]

By: Noah Tambala (ntamb002@gold.ac.uk), Ben Craddock (bcrad001@gold.ac.uk), Khar Chew (kchew001@gold.ac.uk), Syed Sahaf (ssaha003@gold.ac.uk), Jake Brunnen (jbrun001@gold.ac.uk), Spike Elliot (selli006@gold.ac.uk), Shaquille Muhammad Uddin (suddi006@gold.ac.uk)

Group name: Kelvin's Kittens

Second Year Computer Science BSc: IS52018C/F/S - SOFTWARE PROJECTS/ COMPUTING PROJECT 2 (2023-24)

Goldsmiths University of London

Problem

Our first hand experience with room booking at Goldsmiths (two Society Leaders are in our group) and anecdotal accounts from lecturers, allowed us to recognise the genuine issues with the current email-based booking system and the desperate need for an updated system.

With this in mind, we propose to build a user-friendly web based application that aims to streamline the room booking procedure for lecturers, societies and students at Goldsmiths University of London.

We want to address these issues through automation, clarification, visualisation and ultimately, the simplification of the process.

Concept introduction

We aim to produce a real-time, dynamically updated, room-availability timetable, check-in system and a visual aid for booking options, event types and room specifications.

We believe that through automation of the booking process, we will enhance the University experience for attending students by allowing lecturers and societies to easily book appropriate spaces that meet their individual requirements [17, p. 1].

We also believe that this will allow lecturers and society leaders alike to realistically scale, plan and prioritise the conditions of their booking events further in advance. This in turn will enhance the experience of students/attendees and aid in the success of societal events and study periods at Goldsmiths.

According to [16] Online travellers were likely to search for additional information by comparing specific attributes'. With this in mind we need to ensure that potential users of our application are able to adjust the search specification to their needs.

For this we rely on the choice set model, a three-stage, consecutive and funnel-like procedure comprising an awareness set, consideration set and a final choice [20]. A lecturer or student/society rep will visit the booking page and be able to select from either a graphical representation of a particular building or from a drop-down menu to choose a room to book (awareness). The system will then provide the client a list of bookable rooms (consideration). The user will then select a room to book to confirm this booking (final choice) – at which point, depending on your level of clearance, a coordinator would be prompted to approve this.

The coordinator will interact with the system in the sense that they could confirm or deny bookings that have been presented to them by other users. The interface we aim to provide would offer a range of helpful features for such a process – such as presenting the user with a list of available facilities for rooms, as well as automatically templating a risk assessment for them should that be necessary to make their booking. This, in turn, will streamline the current process wherein society leaders are made to complete a full risk assessment each time they wish to book a room. Furthermore - this will also reduce the two-week wait period that is currently in place for confirmation of a booking by speeding up the very slow, non-automated confirmation process that is in place currently. Providing some automation in this way will make booking spaces for extra-curricular events easier and faster – and will encourage more events to be organised and further enrich the Goldsmiths experience.

The current room booking system displays further inadequacies when regarding the scheduling of spaces for educational events like lectures: allowing for bulk bookings over long time periods. This results in rooms being booked for events that may not end up taking place – wasting vital university facilities. Introducing a confirmation system for a previously made room booking on the day would remedy this - and therefore this is a primary consideration for the specification of our system.

Stakeholders/Ethics

For our project for the room booking system, we can acknowledge that the main stakeholders are for society leaders and professors for the university. In the paper *Understanding of online hotel booking process: A multiple method approach. Journal of Vacation Marketing* - [16] they used students and staff working for higher education institutions as the testing pool for hotel booking.From this research we gained insight into the booking behaviours of our primary stakeholders.

Although collecting data from students is a secondary concern (because students wouldn't need to book) we can still collect information from them as booking decisions directly impact their learning experience. This data collected from feedback is focused on how we can make the room booking system simple and easy to use based on the choice-set model [16]. The data we are collecting in our database will be the logins for societies and professor and the information for if a room is available or not. We need to ensure that our data for when the rooms are booked or not to be as accurate as possible as this would greatly upset and disappoint our stakeholders. With our research into databases, mysql will be our best option as it is easy to maintain and secure. We have considered helping individuals having issues with using the room booking system such as large text option [7], colouring for colour blindness [13] as well as universal emojis [11] for those who struggle with English as our prototype will be in English only. Bringing this idea of a project will allow

our stakeholders to be of ease when booking a room as the current method of being able to book a room is very slow, inconvenient and takes too long to both do a booking or wait for a response as you are currently required to submit a form 2 weeks in advance. Our module leader, Sean McGrath is also a stakeholder as he is a professor for the university who has to submit a form to book rooms for his lectures for the module. He as well will be guiding through the development of this application and will be the one grading it. Some of the members of our group are stakeholders as well due to being leaders or helpers of a society.

Market Research

Current System for Goldsmith's Room Booking

After having a brief conversation with Sean McGrath - our seminar leader - we've found that the current system for the university is not user-friendly. There is a system only exclusive to the room coordinators and there is a not-so-clear email system for booking. From this we can infer there isn't really a freedom of choices in what room lecturers need and they are just given a room automatically. Overall it seems like the app we're currently trying to outdo is too complex, does not give a freedom of choice and is automatically assigned to the university staff by the room co-ordinators.

Similarly, the current system for Society Leaders is also inconvenient and unintuitive. Society Leaders are required to navigate through multiple steps, such as filling out a Google Form to reserve a room for their events, only to then complete a separate Risk Assessment document. The current method lacks a streamlined process, and much like the room allocation system for lecturers, does not seem to be user-friendly.

University of Cambridge Room Booking System

Booker is a room and resource booking system used in the University of Cambridge. Before, Colleges each had their own Booking system, meaning there were over 30 different booking systems being used so Booker was developed by the University of Cambridge and in partnership with EventMAP to simplify the booking process [24].

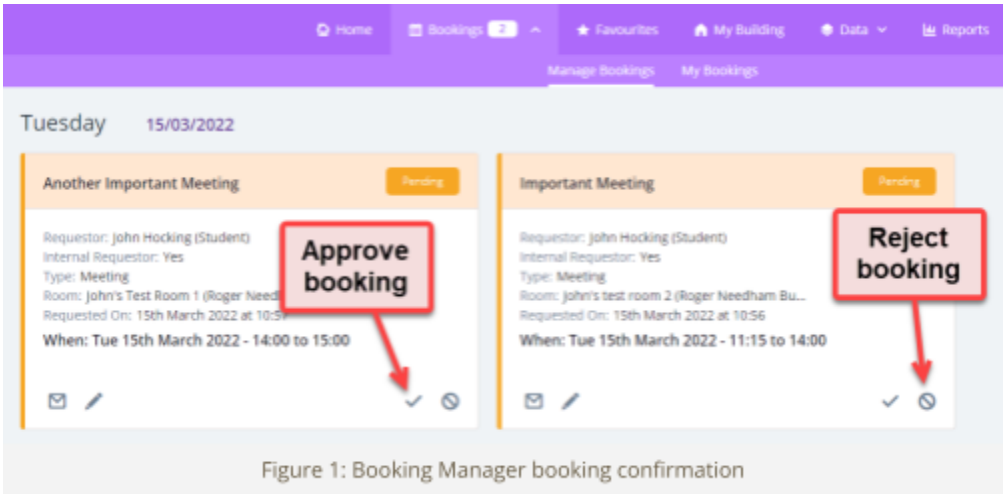


Figure 1: Booking Manager booking confirmation

Strengths:

- Specific user roles for the room booking process: The 'Room Requestor' is the most common role, allowing users to view available resources and make bookings. The 'Room Manager' oversees these bookings, with the authority to approve room bookings. The 'Department Manager' supervises the Room Requesters and Room Managers, overseeing all resources within their department. Lastly, 'Custodians' are responsible for the practical aspects of bookings, such as arranging room layout, catering, and any necessary facilities. Our implementation will be very similar; however, we will need to narrow down roles as Goldsmiths only allows specific roles such as Lecturers and Society Leaders to book roles while in Cambridge the “Room Requestor” is every Student and Lecturer. We will also need to implement the “Co-ordinator” role, similar to the “Room Manager”, who approves room bookings [24].

- They have a filter for rooms which is an invaluable feature to implement because of the implications it has on improving the user experience. They also present a final overview of individual bookings - similarly we plan to send notifications and emails to coordinators, users, organisers and invitees (potentially) [24].

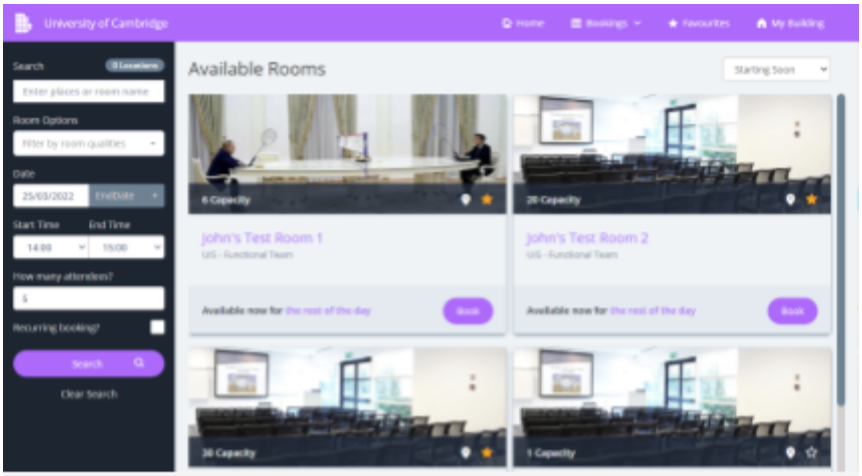


Figure 2: University of Cambridge Room Search

Weaknesses:

- Unlike the Booker system, we will need to include a Health and Safety team that assess Risk Assessments [24].
- Several complaints from staff indicate inconsistencies with the accuracy of information regarding room types and sizes [24].
- Overbooking, or instances where lecture rooms get overcrowded - **We should consider holding reservations so multiple concurrent bookings do not occur.** [24].

SchoolBooking.com

SchoolBooking is a room-booking platform used mostly by primary and secondary schools while our booking system will focus on a university setting, which has a larger scope as there are many more rooms and needs to account for many more students.

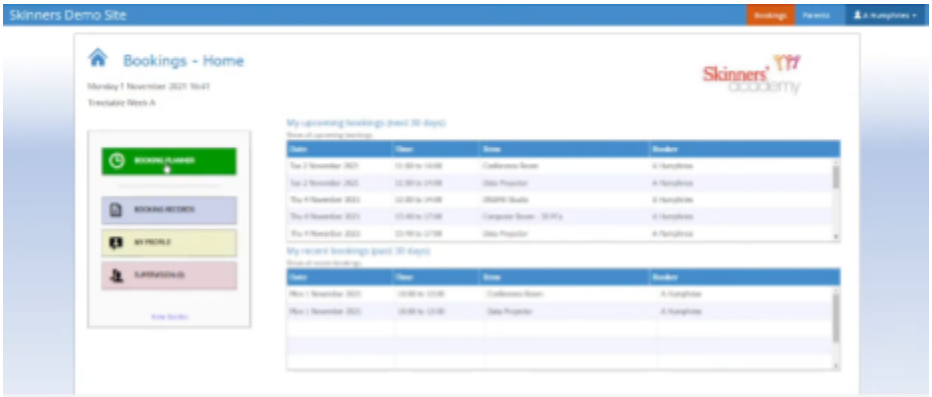


Figure 3: SchoolBooking Home Screen

As part of the booking system, SchoolBooking allows the use of single-sign on, allowing users to sign in using Office365 or Google. Past the sign in screen, the home page allows the user to book rooms, deciding if it's a single instance or recurring, edit or adjust bookings, cancel bookings and view the room booking schedule [26]. Other features included in SchoolBooking are syncing with calendar apps such as Google Calendar and Microsoft Outlook Calendar for real-time updates, offer role-based access controls to restrict or allow bookings by specific user groups, sending reminders and notifications regarding booking. Features such as integration with a calendar app are useful, however that might be too complex and not within our timescope to implement. Role-based control functionality will certainly be necessary to implement as different roles such as society leaders and lecturers will need varying levels of access. An email reminder or notification system will be useful to provide reminders to users [26].

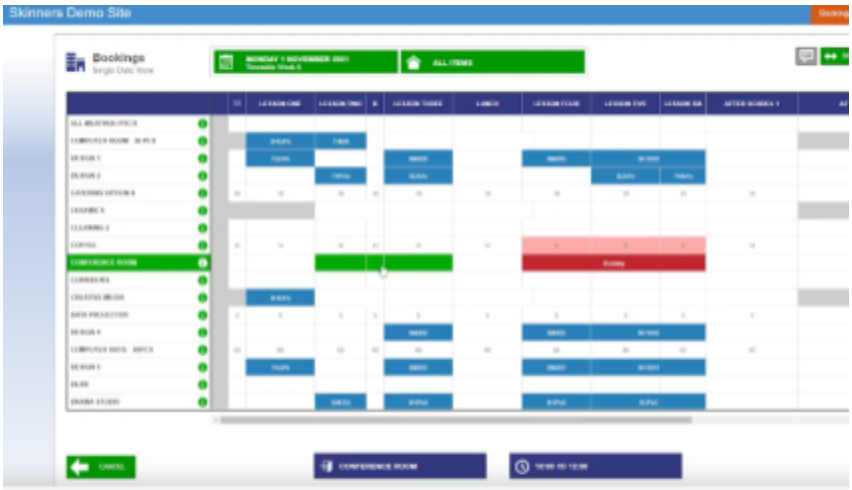


Figure 4: SchoolBooking 'Booking Planner' Screen

Aesthetically, we found the site to appear quite dated and rudimentary in design. Due to the interface's simplicity, however, it seems very easy to navigate - featuring an intuitive

layout and good usability. The user never finds themselves lost in a cascade of layered menus or drowned in a sea of over-designed and redundant features. The design's restraint in that regard is clearly one of its major strengths, and is a philosophy we should take note of, potentially implementing in our own proposed solution [26].

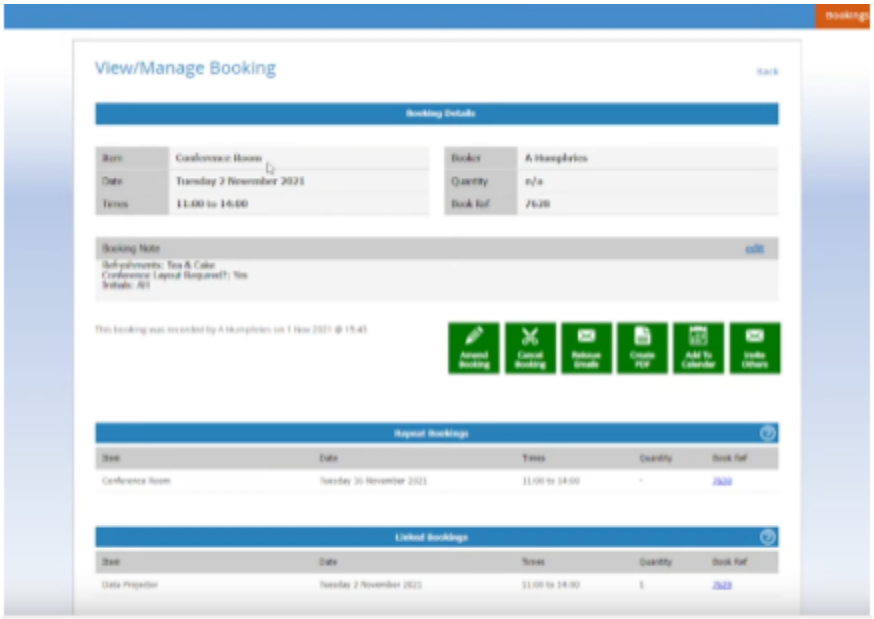


Figure 5: SchoolBooking 'Manage Booking' Screen

RoomBookingSystem.co.uk

This is a room booking system for schools which offers a variety of useful features to help staff within a school to effectively manage their rooms and resources when booking.

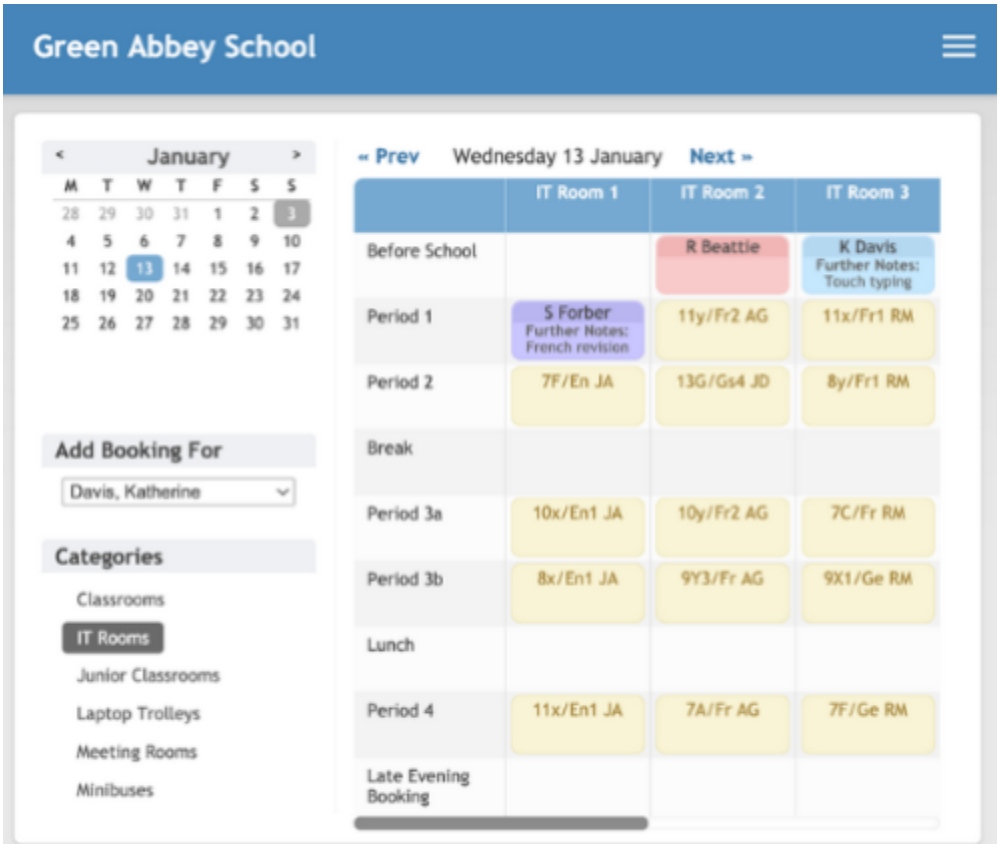


Figure 6: RoomBookingSystem 'Manage Booking' Screen

Strengths:

- They have flexible booking options: you can book by period or time. **Implementing this within our system would be beneficial since it will cater towards the timetables of Goldsmiths students and staff. **
- Integration with School Management System (MIS) (they describe it as automatic). Ensures only staff can book rooms. This shortens the booking process and avoids mistakes or unwanted entities booking rooms. - *permissions are important and we aim to have this implemented for our app - the highest praise this receives from the management signal that this is worth investing in.*
- Also manages sports facilities and specific facilities to book (meeting rooms and theatres etc). * Goldsmiths has a cinema for example, so booking specific facilities is worth implementing*
- Booking management is simplified: emails reminders for the upcoming booking . **useful for resource management - avoiding wasted bookings**
- powerful user" permissions feature; handles all booking settings (booking limits, edit/delete bookings, approving/declining bookings etc). **we will need a super user who can do this** [25].

Considerations:

Stakeholders are slightly different for school systems - only staff can book **we will need to consider for both staff and societies **

Scope

The MVP (minimum viable product) is the version of the final solution stripped back to the bare essentials. This, in turn, allows the aforementioned stakeholders to better visualise what the solution to the initial problem will look like. Thus - we eliminate the risk of developing a system which does not meet the requirements of our users [2] Revision milestones

As a group, we made the decision to use an iterative approach to refine the MVP - communicating frequently with primary stakeholders to adjust the MVP according to their relevant feedback. From this process, we produced the following MVP model as our initial revision of the MVP - which we will refer to as R1:

R1

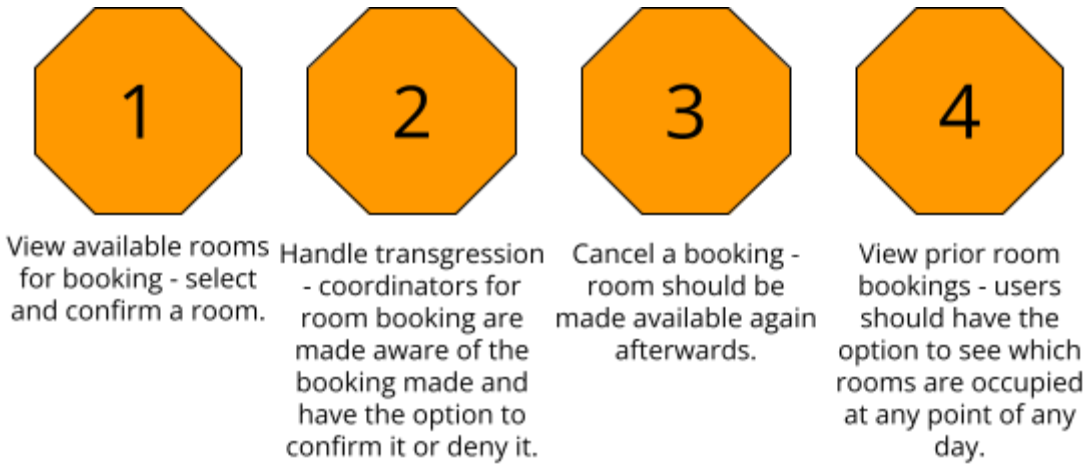


Figure 7: Revision 1 MVP Model

While the following features were not deemed necessary as part of the R1 - we concluded that they would be advantageous additions to our final solution:

- Confirmation system for lecturers to ensure they will use the room they have booked.

- Templating system for risk assessments.
- Facilities lookup - allow for choosing a room based on the facilities available.
    - The system should suggest a room to you with similar facilities should your original booking request be denied.
- Map display of the available rooms.

Shortly after the production of R1 - we decided that we should once again take an iterative approach to refine our MVP to better meet the needs of our stakeholders and users. For this - we agreed on the below timeframe and process for refining our MVP.

We decided that 3 further revisions to our MVP would put us in a good position for the production of a system that adequately solved our main problem, satisfied our goal detailed in our concept, and met the needs of our users and stakeholders. We decided to call these revisions R2, R3, and R4 respectively:

- R2: revision that would be produced after the initial requirements gathering process - the requirements gathering process would give us a basic insight into the main concerns of our stakeholders regarding the current system and give us a better idea of what to focus on when considering our priorities for development.
- R3: revision that would be produced after the final round of the requirements gathering process (once our card sorting activity data had been collected) - this is the point where we would have the greatest insight into which aspects of our system should become a priority with regards to our stakeholder's opinions. We also felt that between the production of R2 and R3, we should start with the initial prototypes and low-fidelity wireframes of our system and further refine R3 based on user and stakeholder feedback.
- R4: final revision of the MVP before main development commences. This would be produced after the ultimate refinement of our low-fidelity wireframes and prototypes and their finalisation into their high-fidelity counterparts based on user and stakeholder feedback. Our aim was to have this be the version of the MVP that best reflected the system we wished to create that would satisfy our stakeholders and effectively solve our problem.

We decided on the following time frame for refining our MVP along with the prototyping process and gathering user feedback:

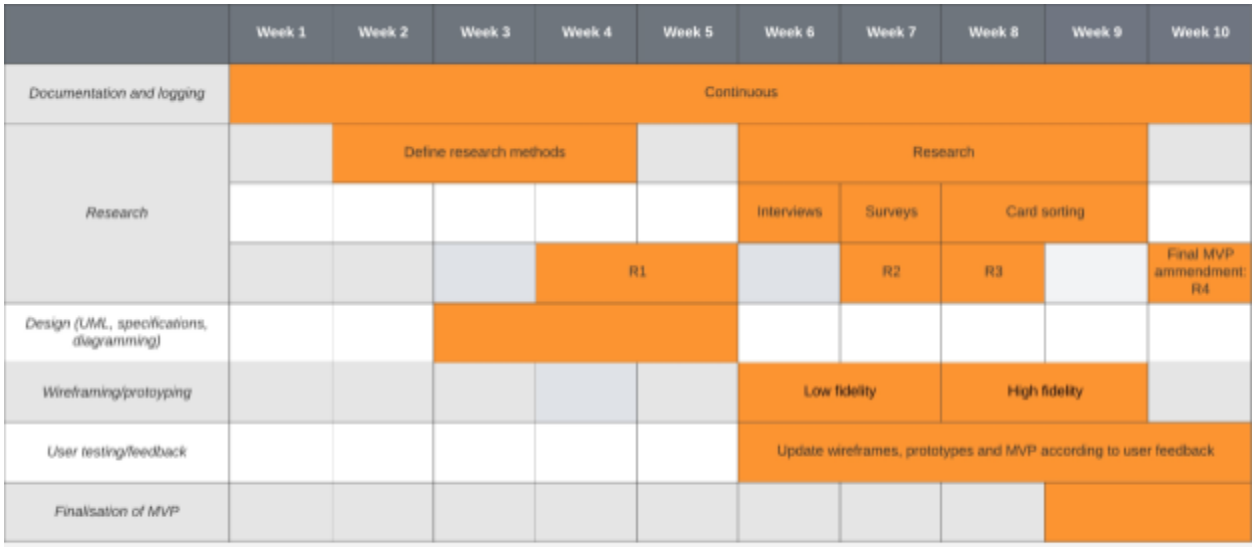


Figure 8: Planned MVP Timeline

R2

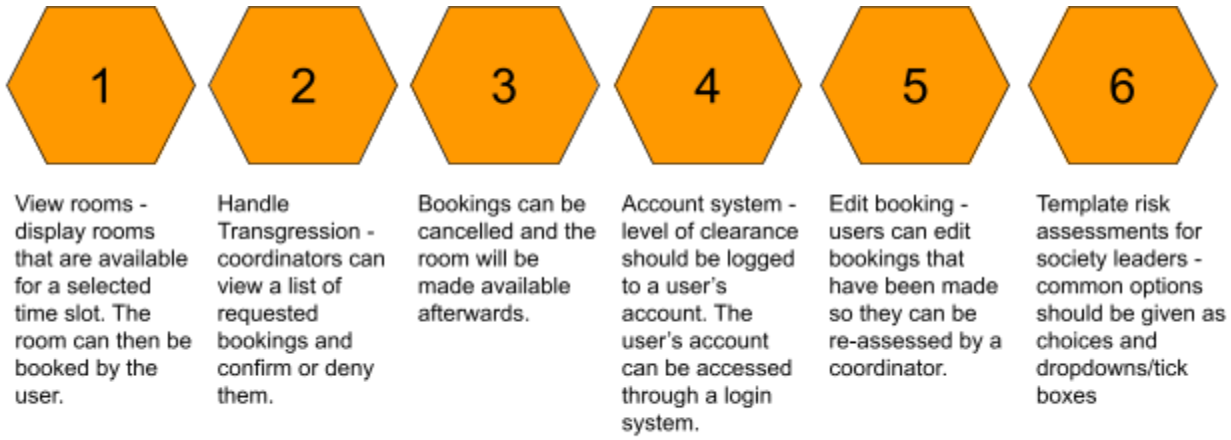


Figure 9: Revision 2 MVP Model

While the following features were not deemed necessary as part of the R2 - we concluded that these, along with the aforementioned additional features proposed for R1, would be advantageous additions to our final solution:

- Calendar display for selecting dates and times of desired booking
- Separate pages for viewing current bookings and making a new booking
- FAQ page

R3

- Changes will be made based on the feedback from R2
- See “Design - Prototyping - Prototype Set #3 (R3)”

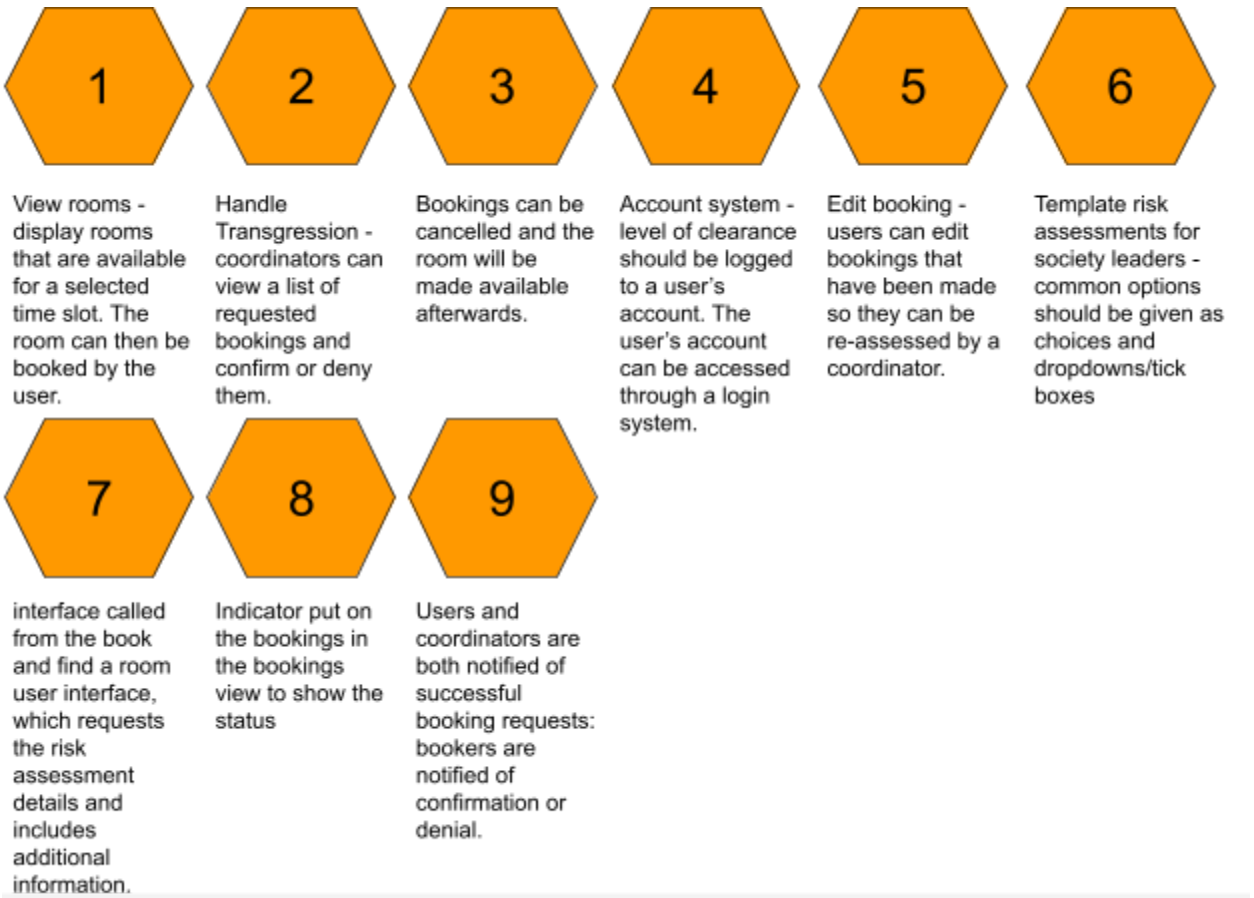


Figure 10: Revision 3 MVP Model

R4

- Changes will be made based on the feedback from R3 and also design considerations including colours, accessibility, and the flow of the application.
- See “Design - Prototyping - Prototype Set #4 (R4)”

Figure 11: Revision 4 MVP Model

Design

We proposed this as our original blueprint for how we would approach designing our application:

Figure 12: Original Blueprint

Use Cases

We decided to start our design process with a comprehensive set of use cases. This ensured that we kept our users in mind while coming up with our specifications [5].

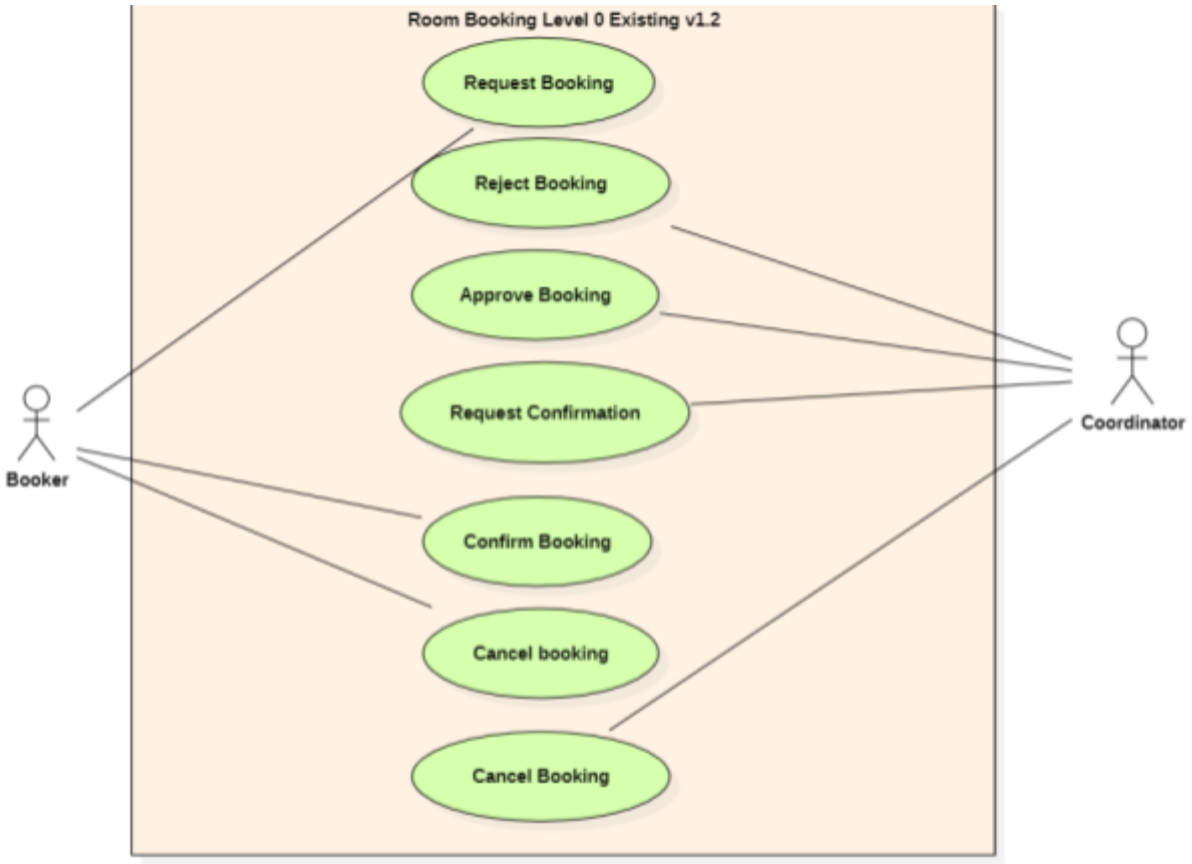


Figure 13: Room Booking Use Case Level 0

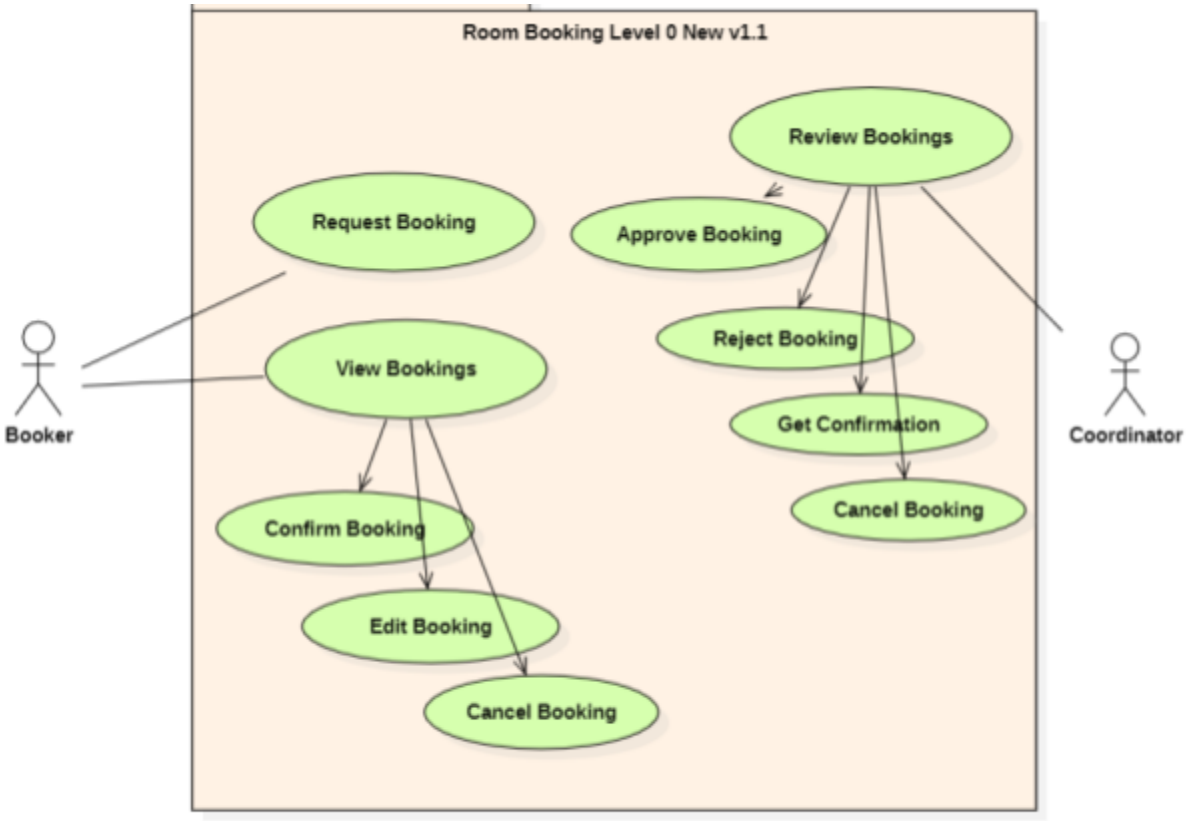


Figure 14: Room Booking Use Case Level 0

Level 1:

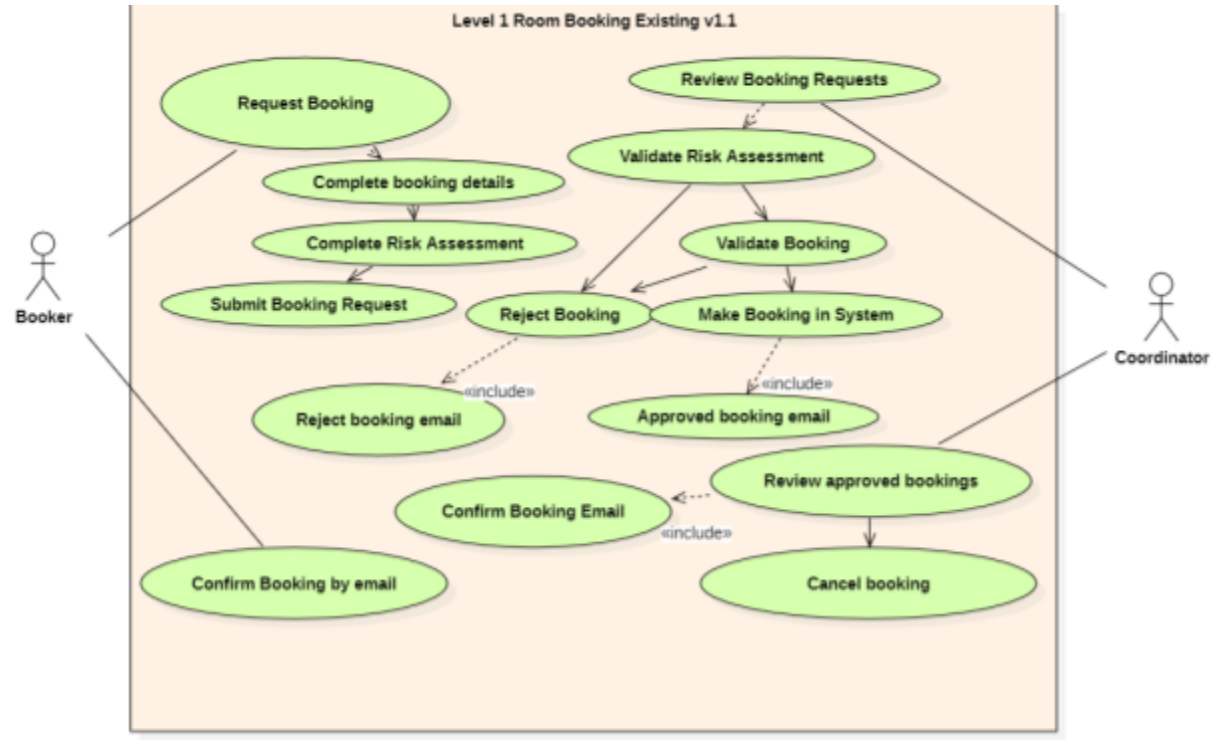


Figure 15: Room Booking Use Case Level 1

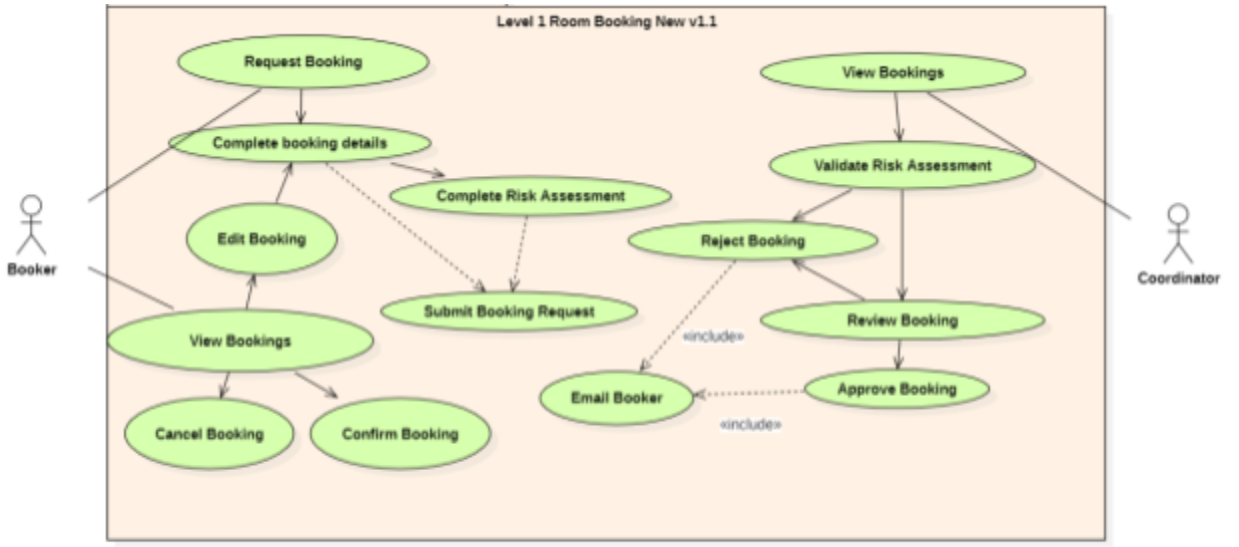


Figure 16: Room Booking Use Case Level 1

Activity diagram

We originally proposed the following flow structure for our application:

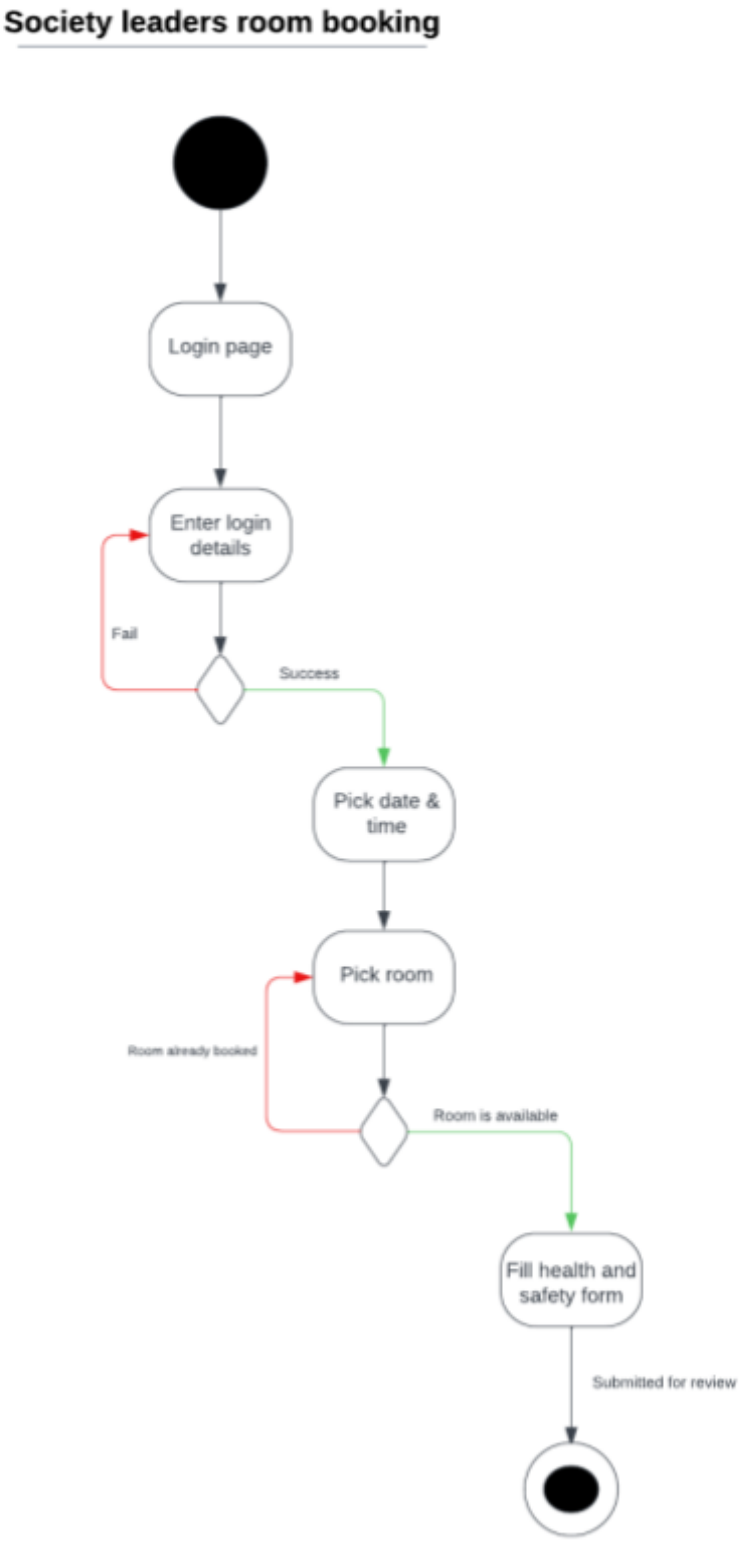


Figure 17: Activity Diagram Revision 1 for Society leaders room booking

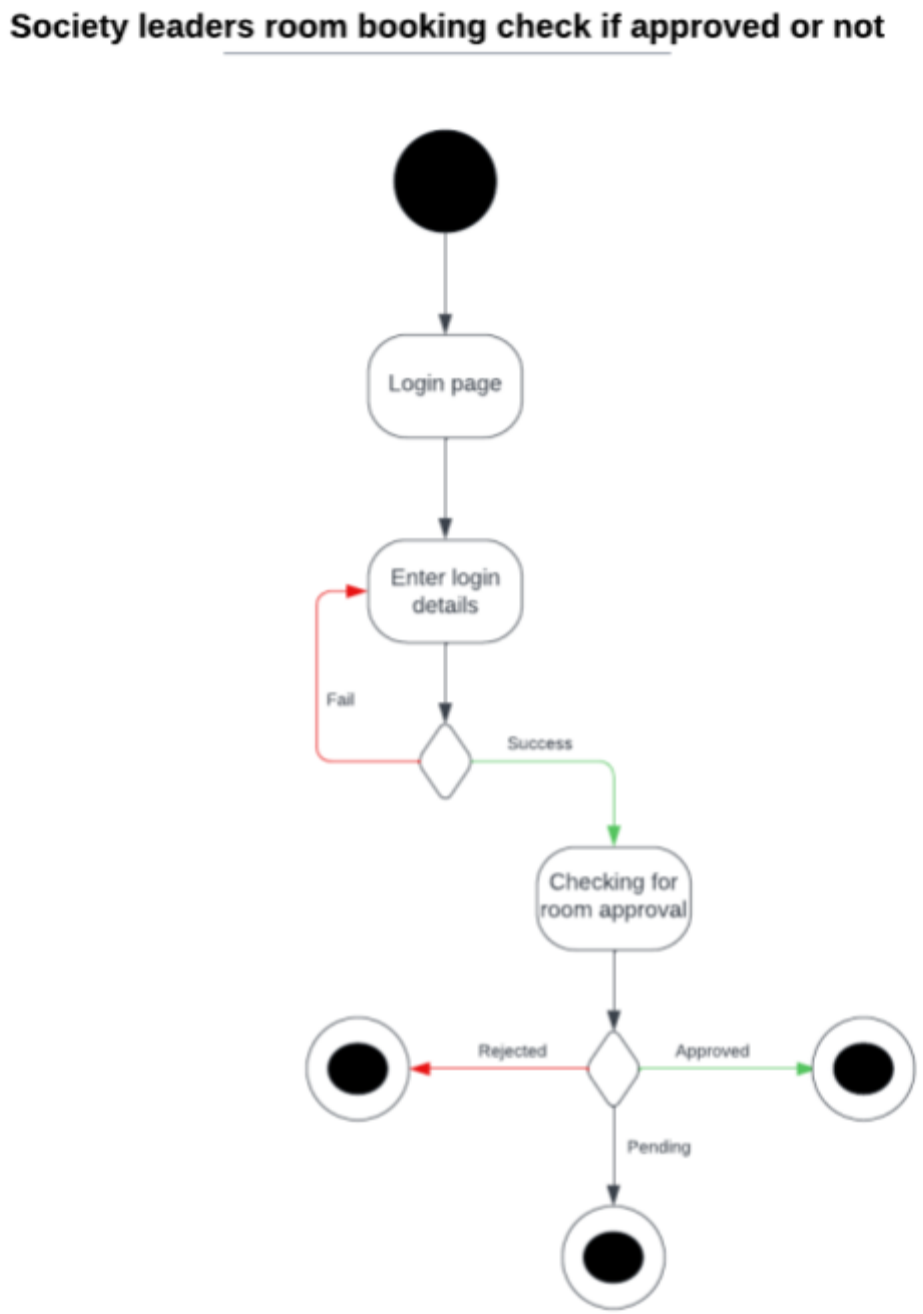


Figure 18: Activity Diagram Revision 1 for Society leaders room booking check if approved or not

However, this structure did not prove to be comprehensive enough for our purposes. As a result - we reconvened as a group and decided it would be more prudent to produce a single activity diagram to specify the flow of the whole system. While this looks very

complicated, having it all in one place ended up giving us a greater understanding of the processes that would be implemented into the final product [5, p. 2184]. See below:

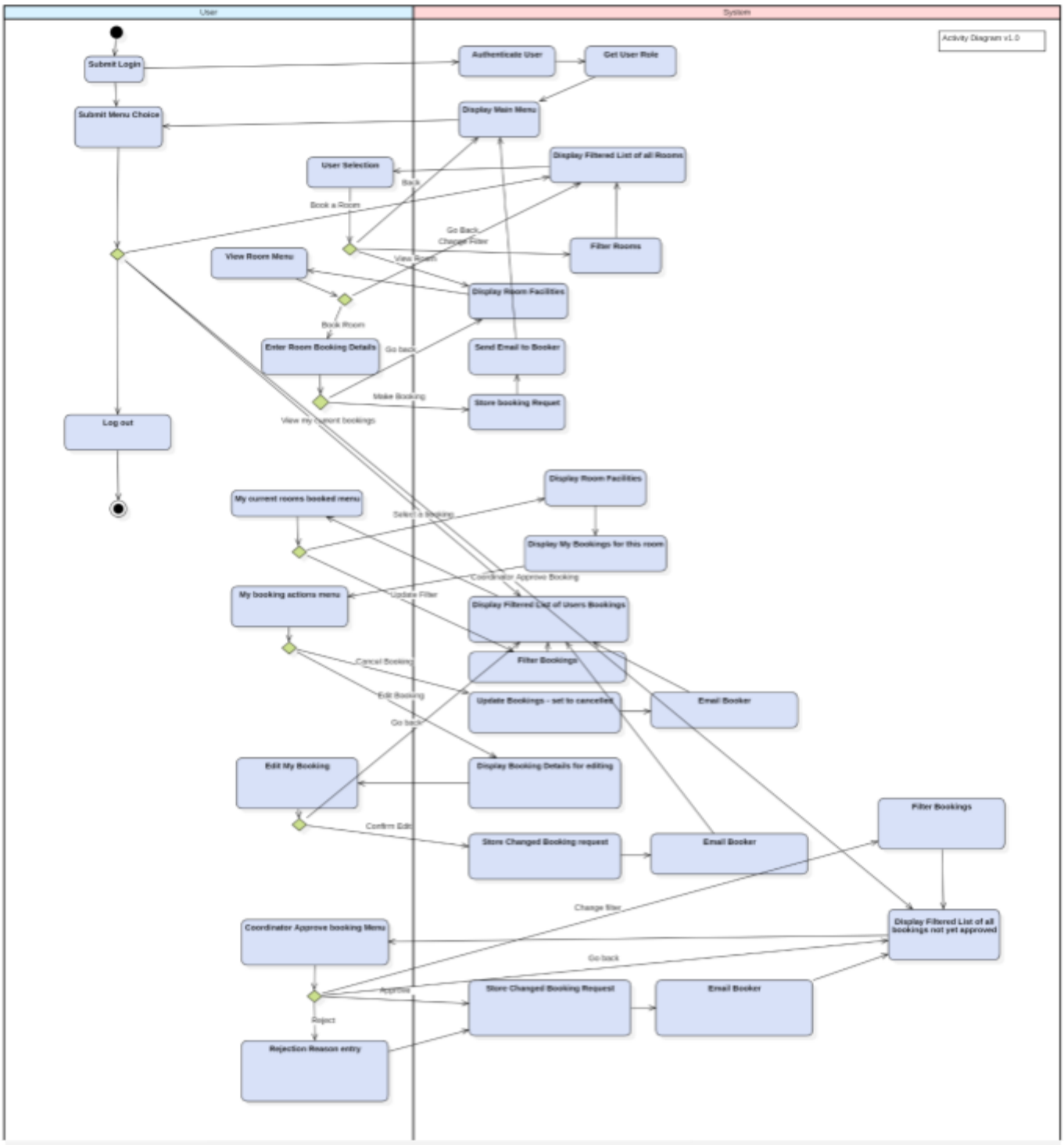


Figure 19: Activity Diagram Final Revision

Sequence diagrams

We began by creating the below sequence diagram - however we felt it did not accurately represent the processes between the proposed systems within our solution [1, p. 340]. As a result of this, we created two further amendments to this sequence diagram - eventually ending up on the third revision as seen below as the agreed upon version of the sequence diagram:

First sequence diagram revision

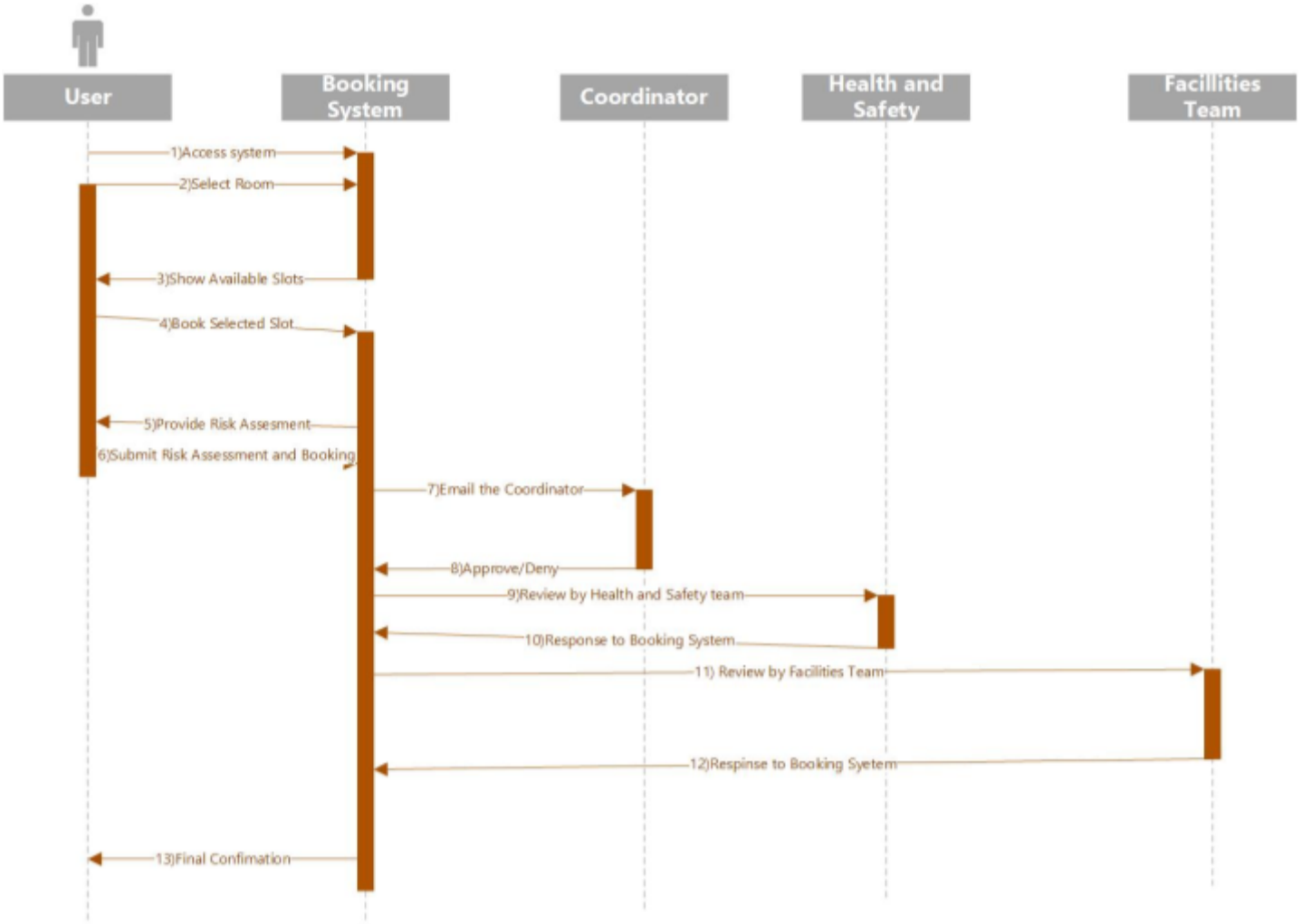


Figure 20: First sequence diagram revision

Second sequence diagram amendment

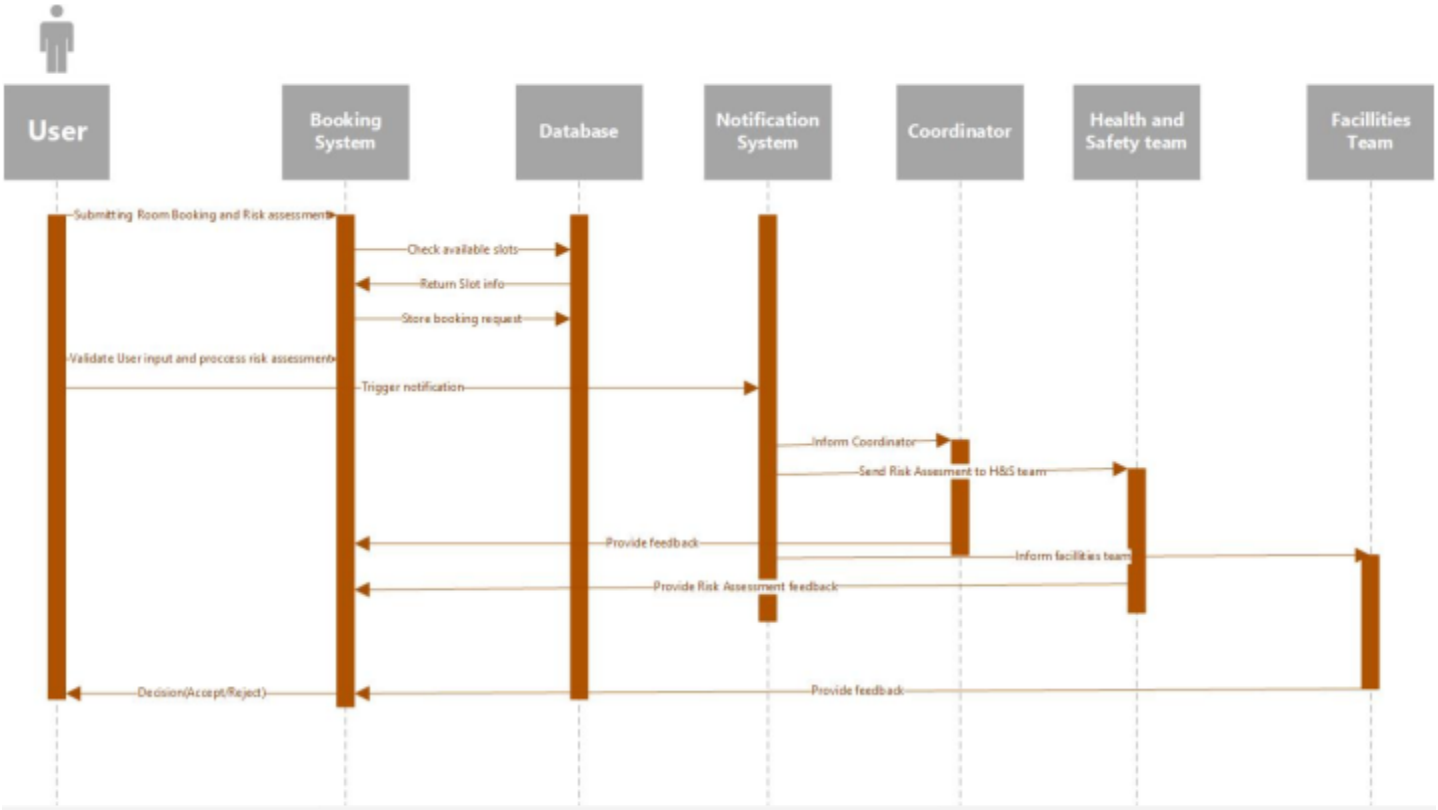


Figure 21: Second sequence diagram revision

Third sequence diagram amendment

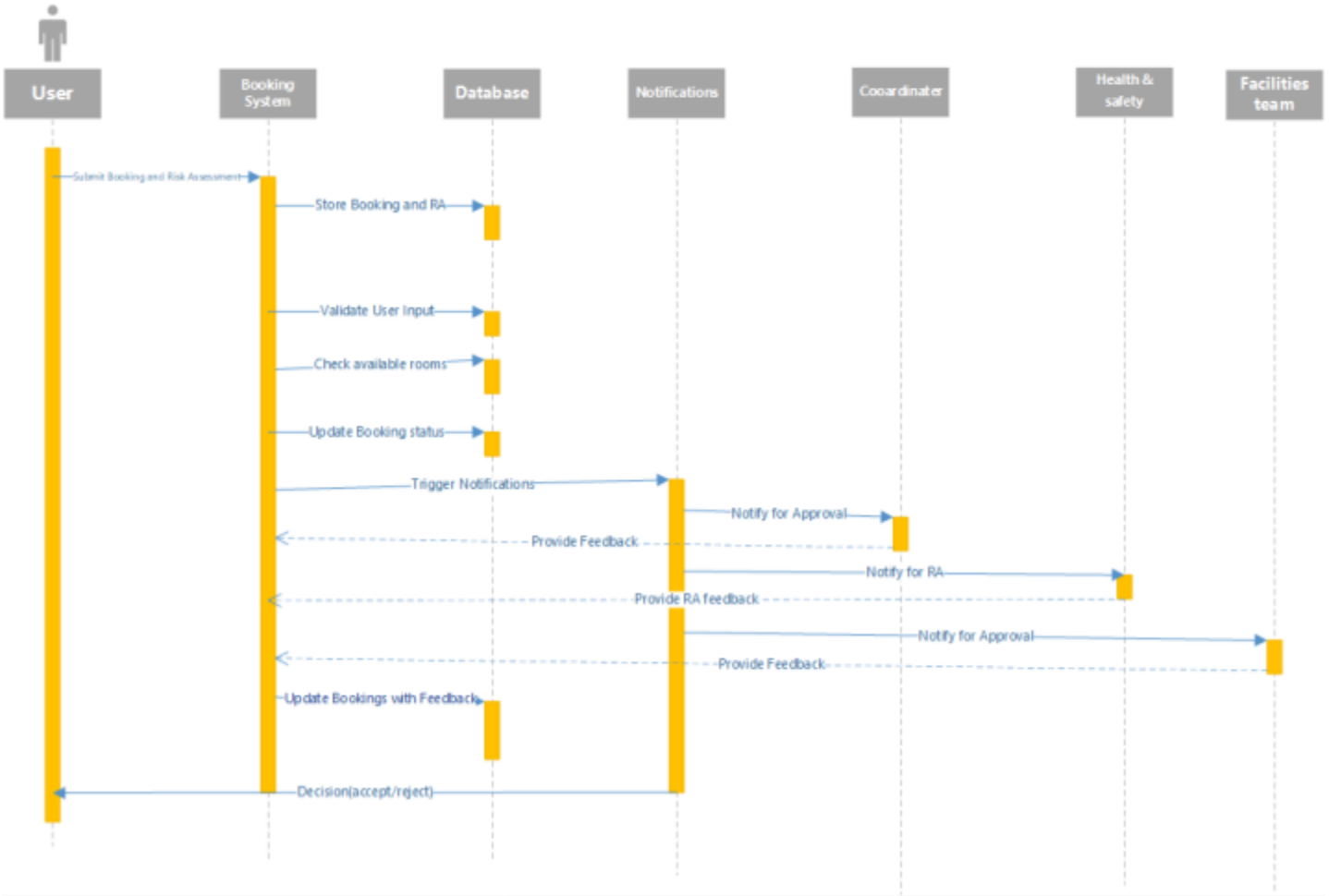


Figure 22: Final sequence diagram revision

Class diagram

The "User" class represents anyone who interacts with the room booking system. Each user is identified uniquely by a UserID and with attributes like their UserRole and Email. The different roles are represented as subclasses, all of which inherit attributes and methods from the User class. The Society Leader is a leader of a student society with the capability to submit bookings and submit risk assessments. The Lecturer can schedule lectures in rooms. The Coordinator has the responsibility to review, approve, or reject bookings. The Health & Safety Staff reviews risk assessments associated with bookings. The Facilities Staff review room bookings.

The Booking System makes booking requests, stores confirmed bookings, and sends notifications to users. Bookings are recorded by the unique bookingId and maintained in

the list of CurrentBookings The Booking System is responsible for validating booking requests, storing confirmed bookings, and sending notifications to users.

The Booking System is used to book a Room, which is its own class and represents the physical room that can be reserved. Every room has a unique roomID, a roomNumber, a specified capacity, and a list of Facilities it offers.

Every Society Leader will need to fill out a Risk Assessment, which is represented by a class, which is used to evaluate potential hazards for room bookings. Each assessment has a unique assessment, the different risks, the approval status, and a record of who approved it.

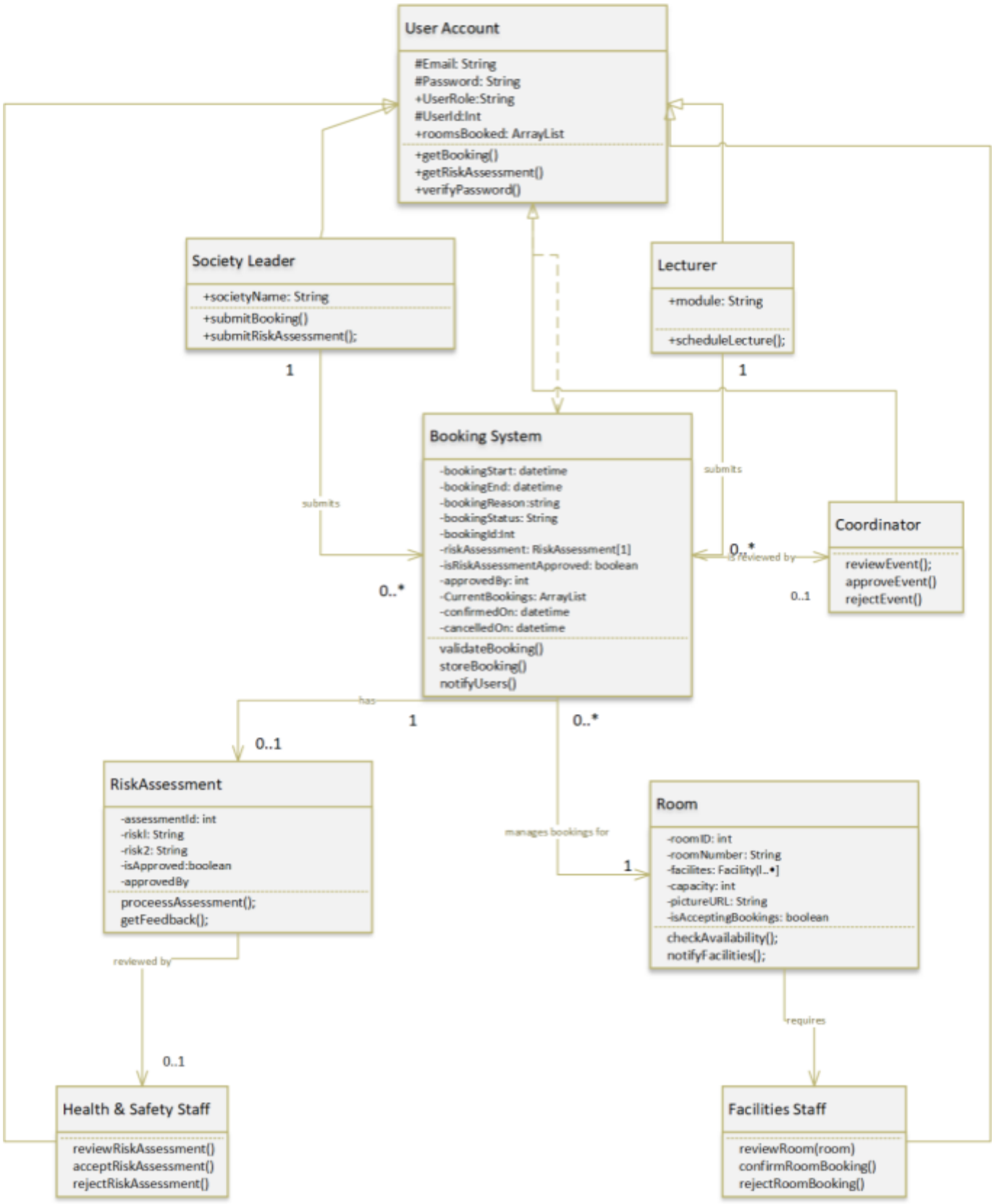


Figure 23: Class Diagram

Design - Prototyping

Paper Prototype, Set #1

Summary

Before starting our interviews and questionnaires we have created paper-prototype sketches with the plan of receiving user-feedback and then implementing these improvements into a second set of paper prototypes. Four main pages were created to start off with, inspired by our previous blueprints and diagrams as inspiration for what we may look for. Those pages are: Home page for users, home page for co-ordinators, booking page and view current bookings page. This process allows us to collect straightforward pros and cons as feedback, allowing the implementation of necessary changes in our second set of paper prototypes.

Prototype Drawings

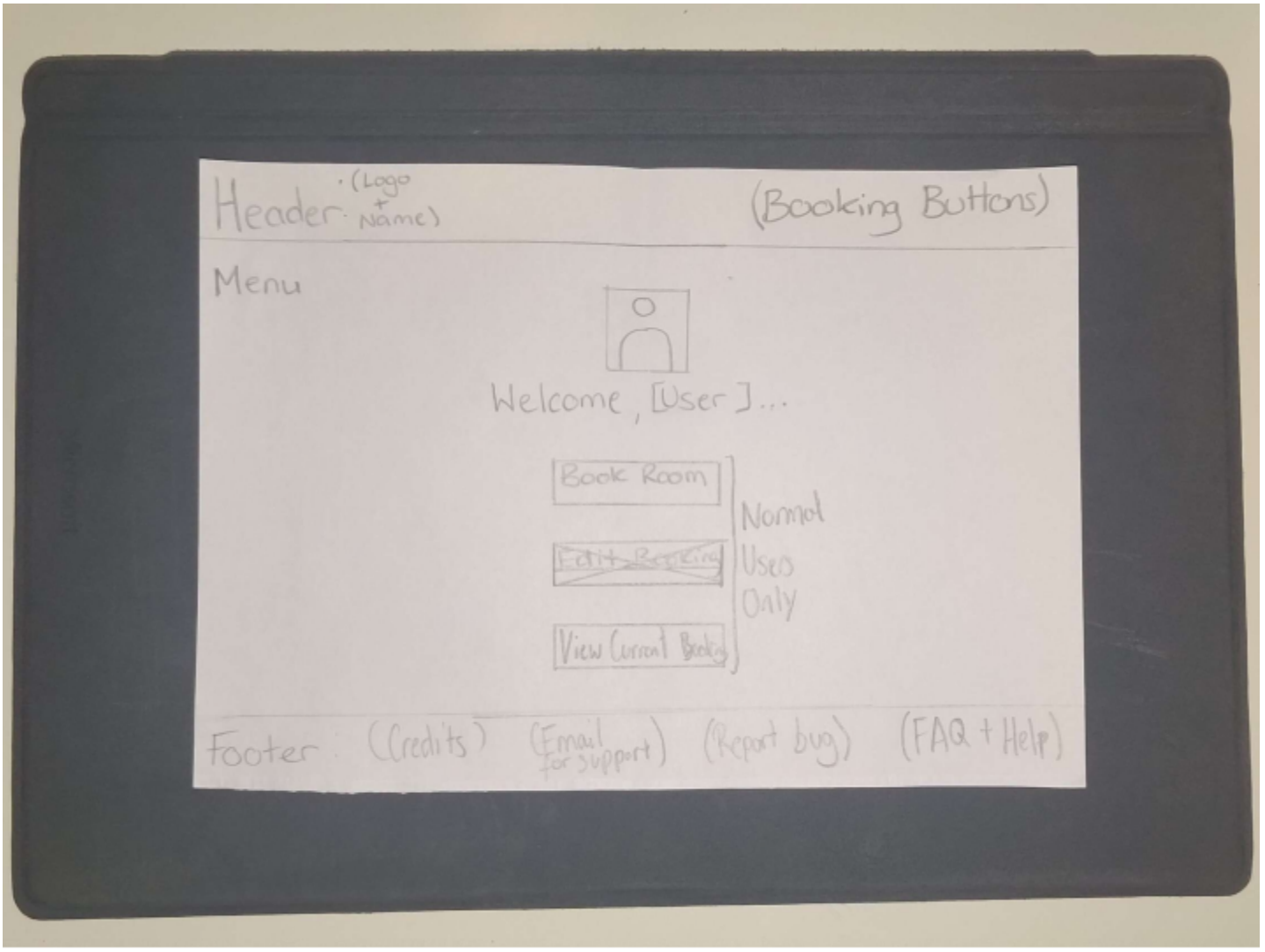


Figure 24: Paper Prototype 1 for User Login

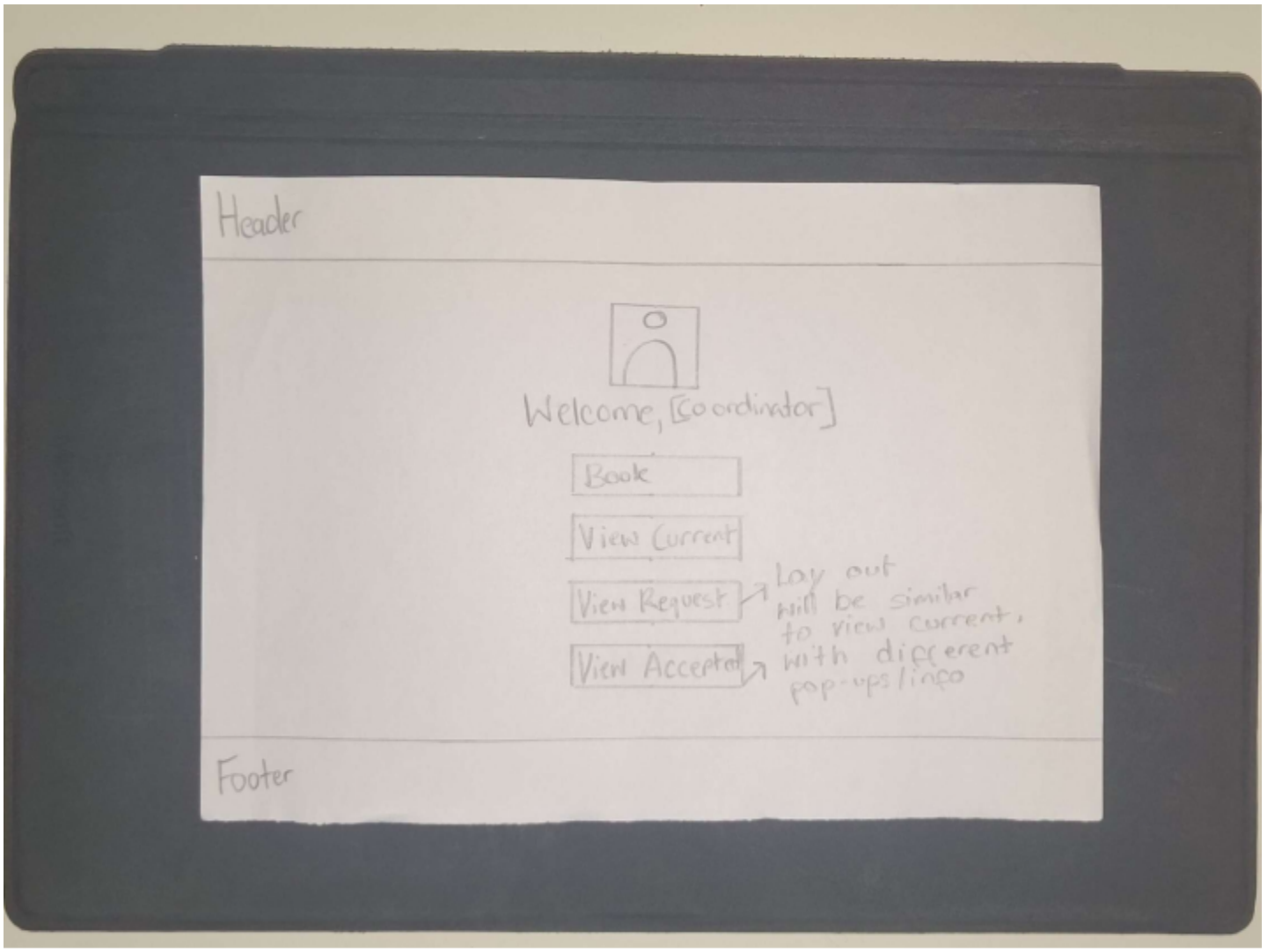


Figure 25: Paper Prototype 1 for Coordinator Login

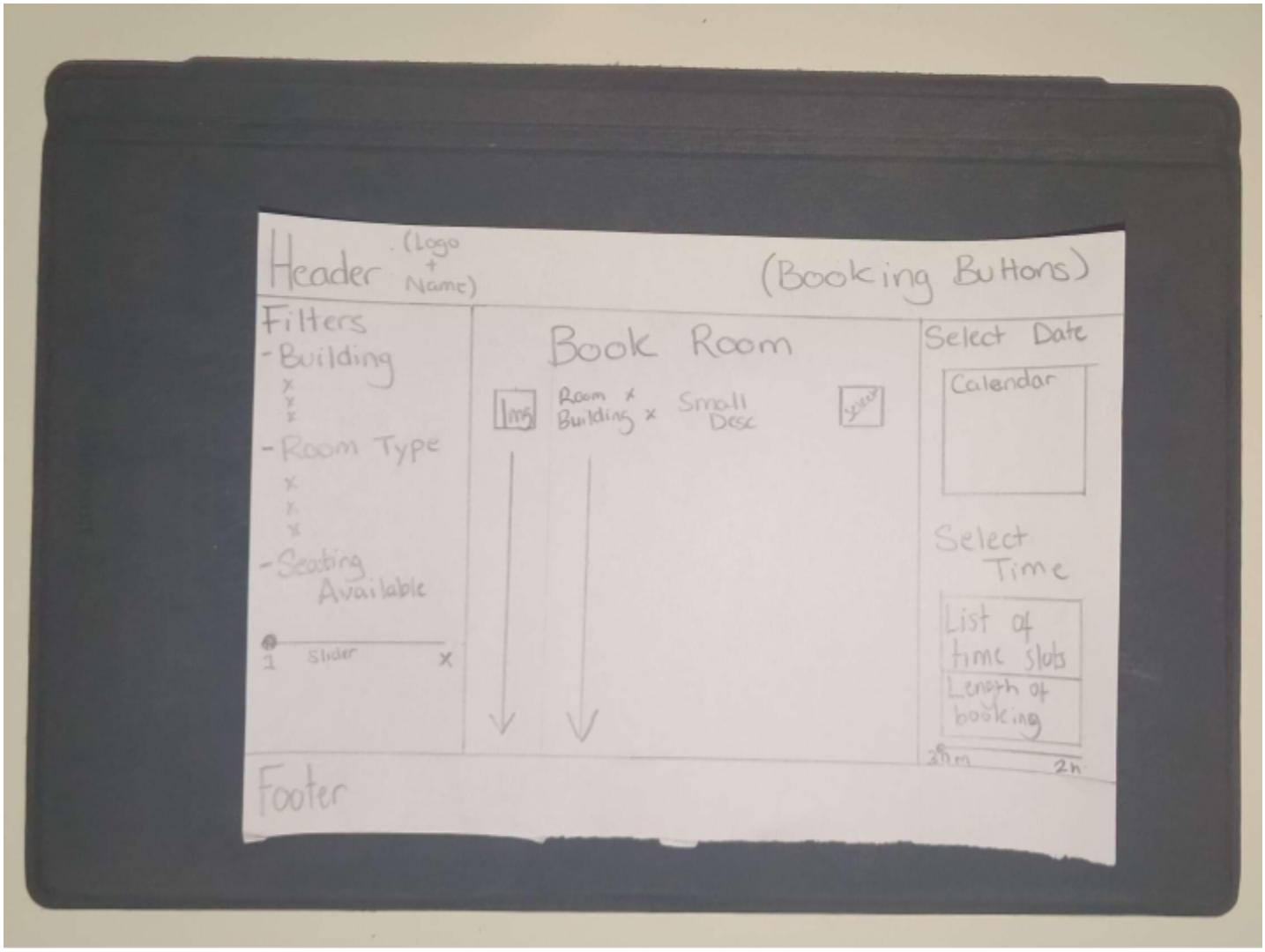


Figure 26: Paper Prototype 1 for booking rooms

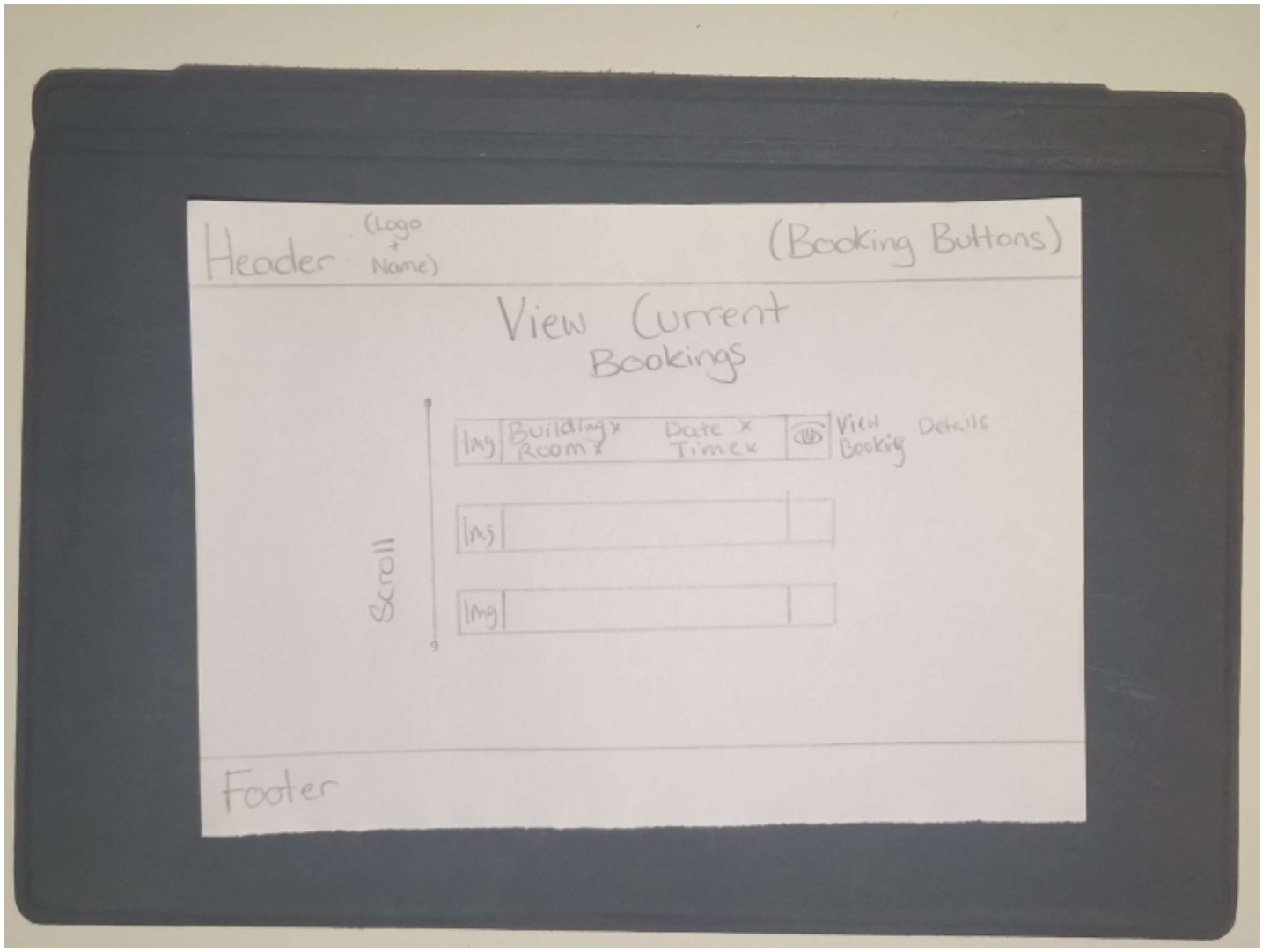


Figure 27: Paper Prototype 1 for current bookings

Pros

1. Good layout for user and coordinator screens, good incorporation of calendar.
2. Separate screen for viewing current bookings is good
3. Calendar display and large section for booking options are nice.
4. Differences in welcome pages are nice.
5. Filter options to sort rooms are nice

Cons

1. Include risk assessments in wireframes
2. Log-in page needed

3. Incorporate a calendar display
4. Create page for current bookings page
5. Confusion with display options on both sides
6. Edit booking screen needed in prototype
7. Layout of room booking screen doesn't give enough space for available time slots - create a feature to give this space

Cons - Solutions

1. More prototyping (new page for risk-assessment?)
2. More prototyping
3. Change booking prototyping
4. More prototyping
5. Change booking prototyping
6. More prototyping
7. More prototyping (pop-up?)

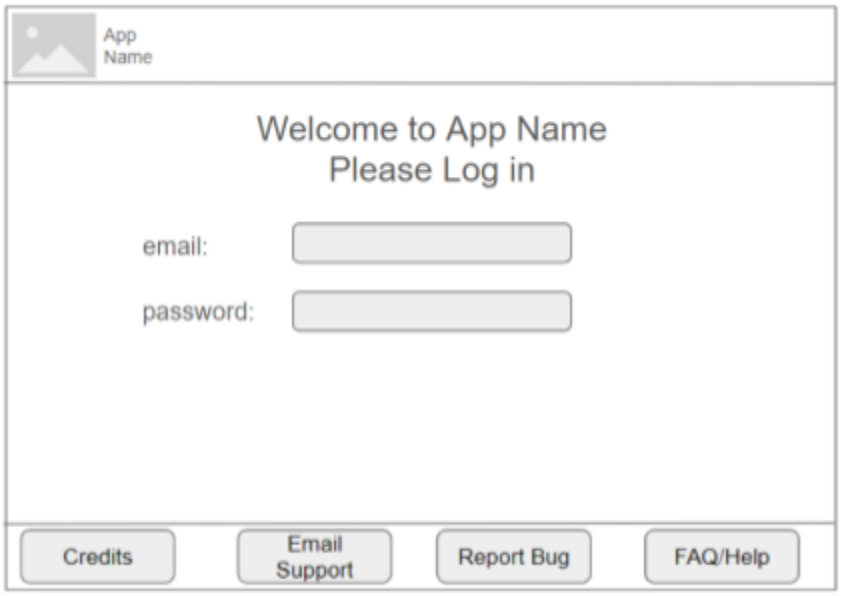
Prototype, Set #2 (R2)

Summary

Following the R1 feedback, and reviewing the requirements for R2 we produced a set of prototype drawings, for review.

Prototype Drawings

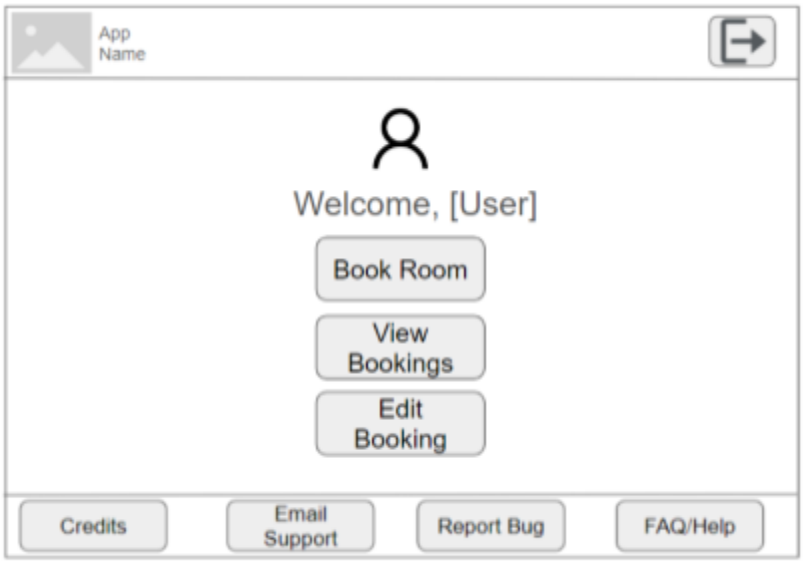
UI: Login



Notes: Account system. Login page

Figure 28: Digital Prototype 1 login screen

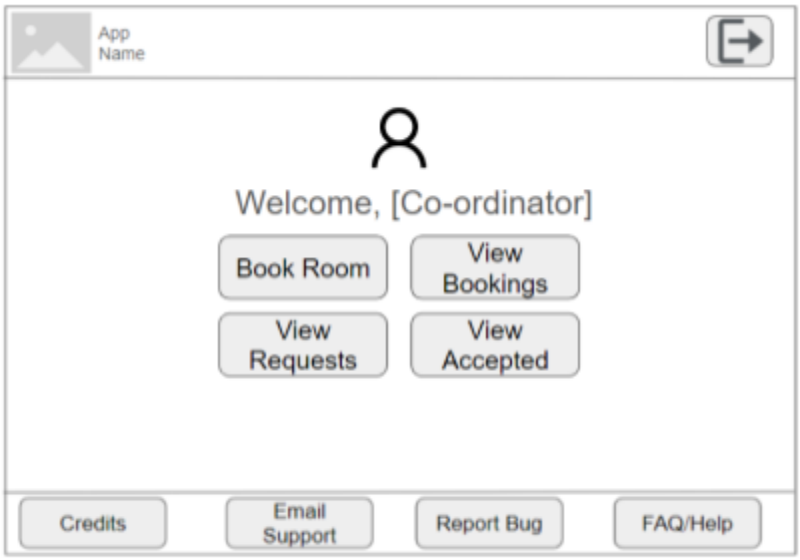
UI: Main menu for lecturer or society leader



Notes: Non coordinators get a custom menu. The logout button, top left, is part of the Account System.

Figure 29: Digital Prototype 1 main menu screen

UI: Main menu for Coordinator

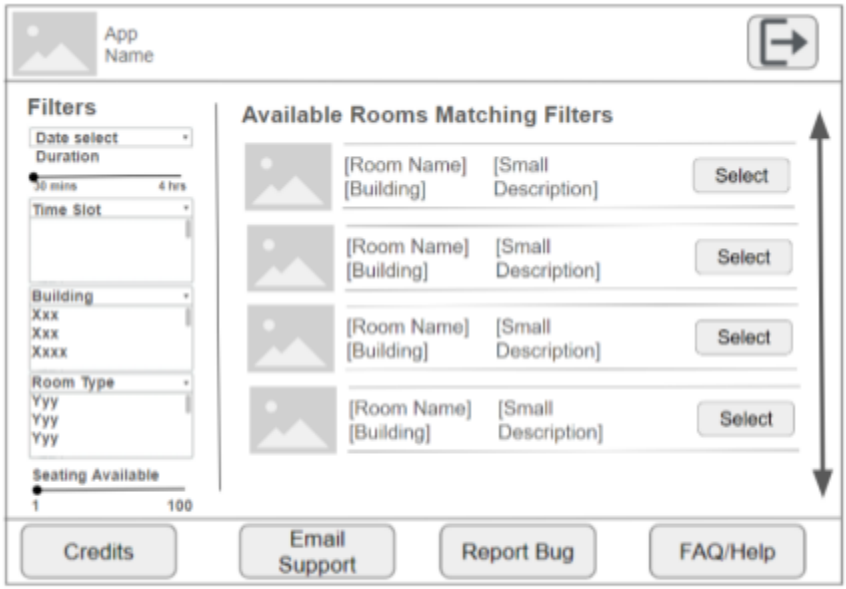


Notes:

Coordinators get different menu options.

Figure 30: Digital Prototype 1 main menu screen for coordinators

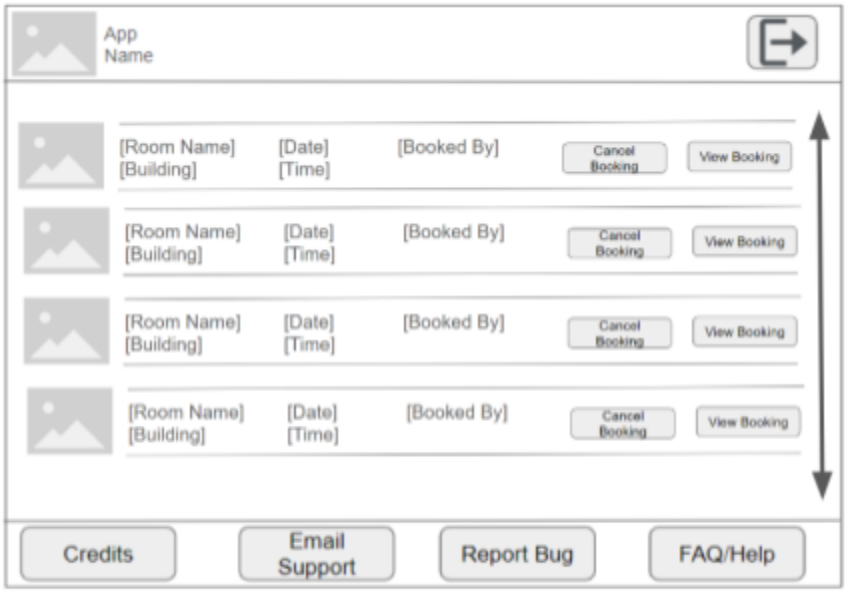
UI: Book Room - Find Room and time. Room Bookings



Notes:

Figure 31: Digital Prototype 1 book room screen

UI: View Current Bookings - called from User Main Menu

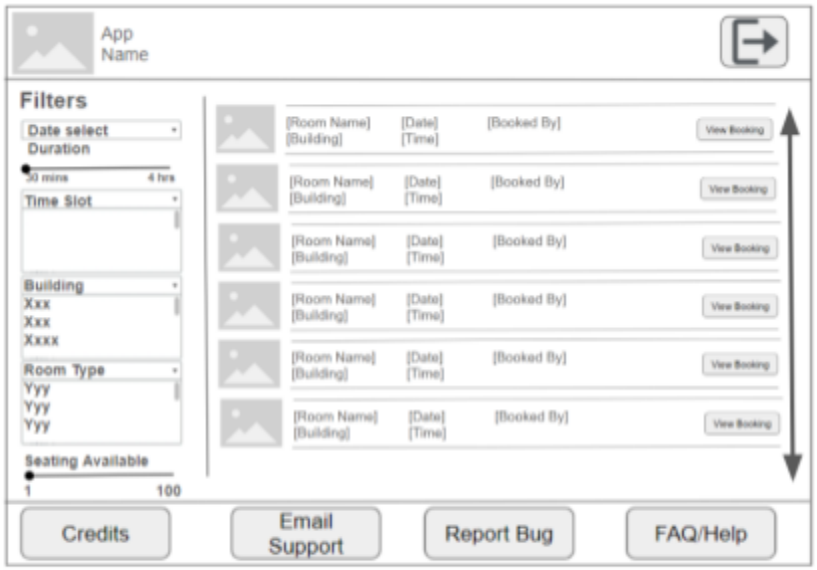


Notes:

View booking displays view booking UI. Filters based on currently logged in user.

Figure 32: Digital Prototype 1 current bookings screen

UI: View Current Bookings - from Co-ordinator Menu

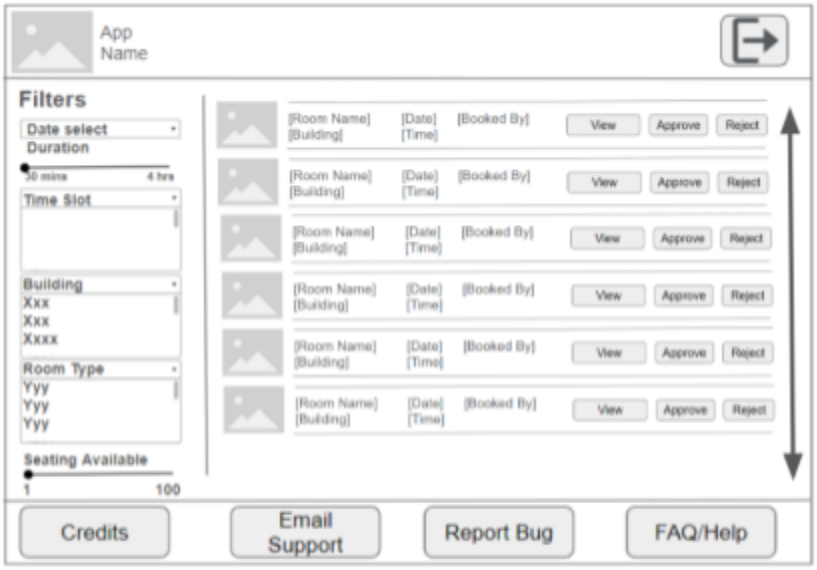


Notes:

View booking displays view booking UI. Coordinator will be managing lots of bookings so a filter is provided displays only confirmed bookings AND filters

Figure 33: Digital Prototype 1 current bookings screen for coordinators

UI: View Requests - from Co-ordinator Menu

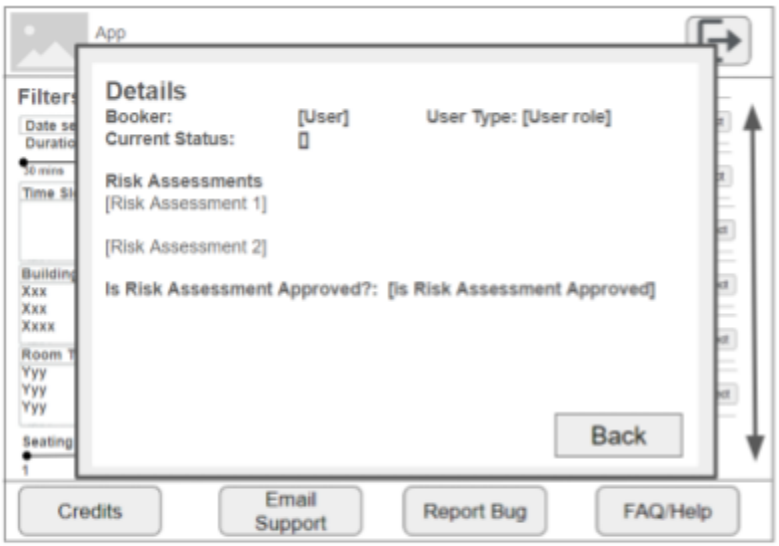


Notes:

Shows all booking requests that have not been approved or rejected. View button shows booking details UI

Figure 34: Digital Prototype 1 view requests screen

UI: View Booking Detail / Requests Detail



Notes:

This is a pop up UI which appears over the current UI. It is used from multiple other UIs

Figure 35: Digital Prototype 1 view requests screen (details)

Pros

1. The design is intuitive and simple. All the information is provided and it makes sense in the context of the application.
2. The information hierarchy is present and makes clear sense.
3. Navigation is clear. I can tell how the program will flow.

4. The filters being on one side rather than two makes sense and is easier to read.

Cons

1. There is no place to enter the risk assessment from the find booking user interface for the user. There is just a select.
2. There is no edit booking user interface
3. There is no view accepted user interface for the coordinator
4. We know the application name.
5. How will a user know if their booking has been accepted or rejected in the system
6. How will the user be notified about a booking being accepted or rejected

Cons - Solutions

1. New interface called from the book and find a room user interface, which requests the risk assessment details and includes additional information.
2. Create an edit booking interface.
3. Make a new interface based on view requests but with a filter that only shows accepted requests.
4. Change the application name to Bookit.
5. An indicator could be put on the bookings in the bookings view to show the status and maybe another filter for status, and options to order the list by status, room or date.
6. The user can get notified by email with a link to the booking (they will have to log in first).

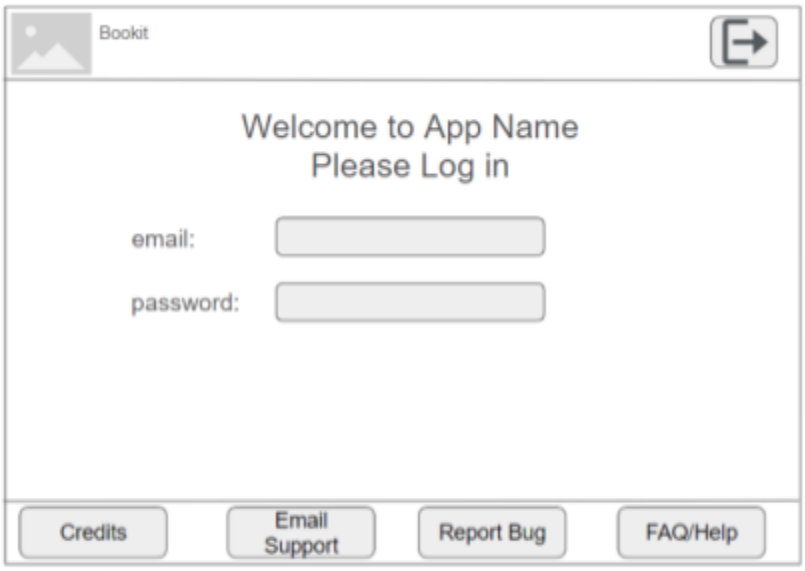
Prototype, Set #3 (R3)

Summary

Following the R1 & R2 feedback, and we produced revised set of prototype drawings, for review.

Prototype Drawings

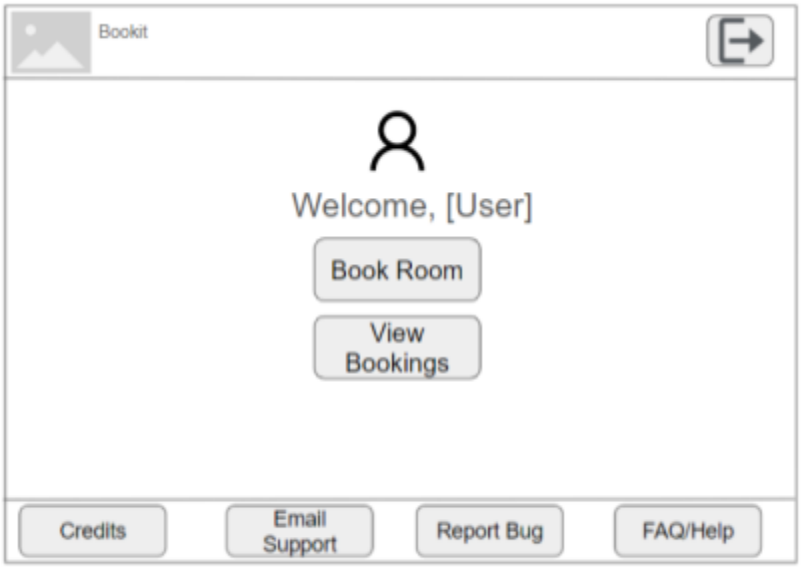
UI: Login



Notes: Account system. Login page

Figure 36: Digital Prototype 2 login

UI: Main menu for lecturer or society leader



Notes:

Non coordinators get a custom menu. The logout button, top left, is part of the Account System.

R3. edit a booking button was removed. This is now a button on the view bookings interface.

Figure 37: Digital Prototype 2 main menu for society leader or lecturer

UI: Main menu for Coordinator

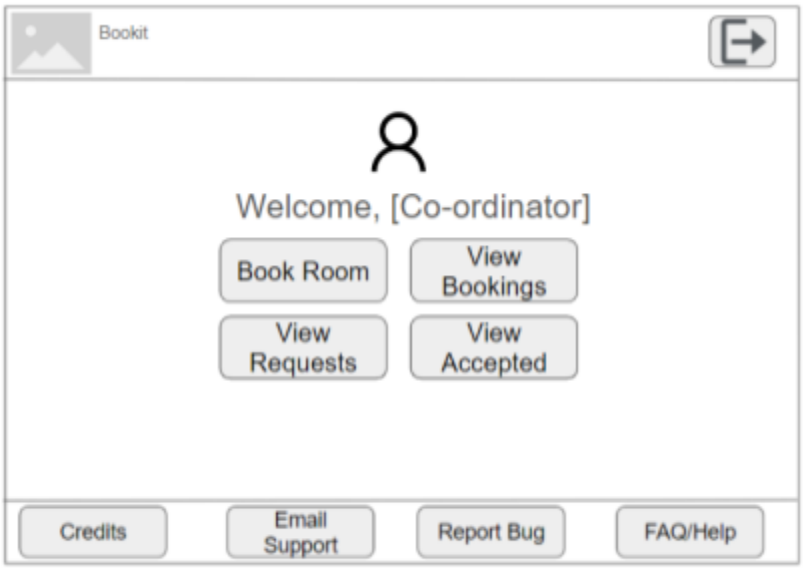


Figure 38: Digital Prototype 2 main menu for coordinator

UI: Book Room - Find Room and time. Room Bookings

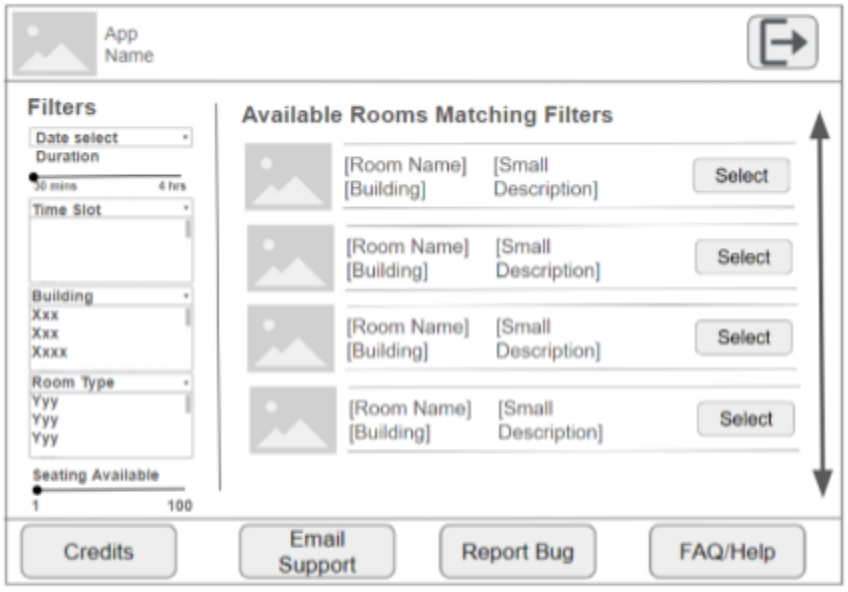
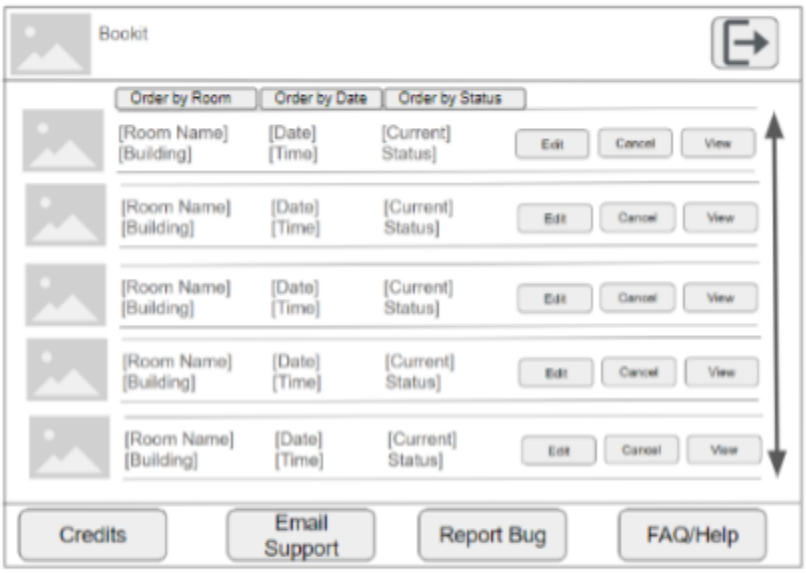


Figure 39: Digital Prototype 2 book room screen

UI: View Current Bookings - called from User Main Menu



Notes:

Filters data based on currently logged in user.

View displays view booking UI.

Cancel. Cancels the booking.

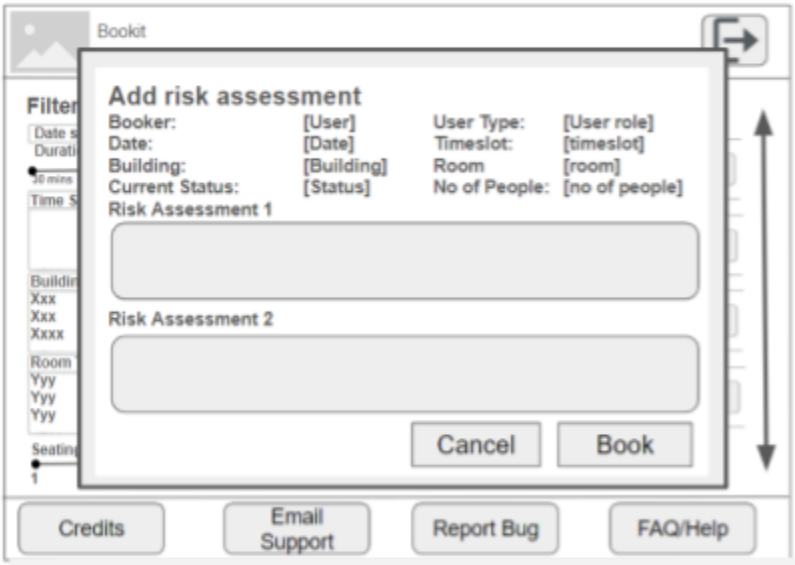
R3: Edit button which displays the Edit booking interface

R3: Current status is show so user can tell which bookings have been approved, rejected or are waiting.

R3: added list ordering buttons to help find data if there are a lot of bookings listed

Figure 40: Digital Prototype 2 current bookings

UI: Complete Booking (enter risk assessment)



Notes:

R3: Added risk assessment Entry user interface. Allows user to enter the risk assessment for their booking.

R3: Cancel ignores changes and returns to find a room.

R3: Book saves the risk assessment details and changes the status of the booking so the coordinator will see it in their view requests list

Figure 41: Digital Prototype 2 risk assessment screen

UI: View Current Bookings - from Co-ordinator Menu

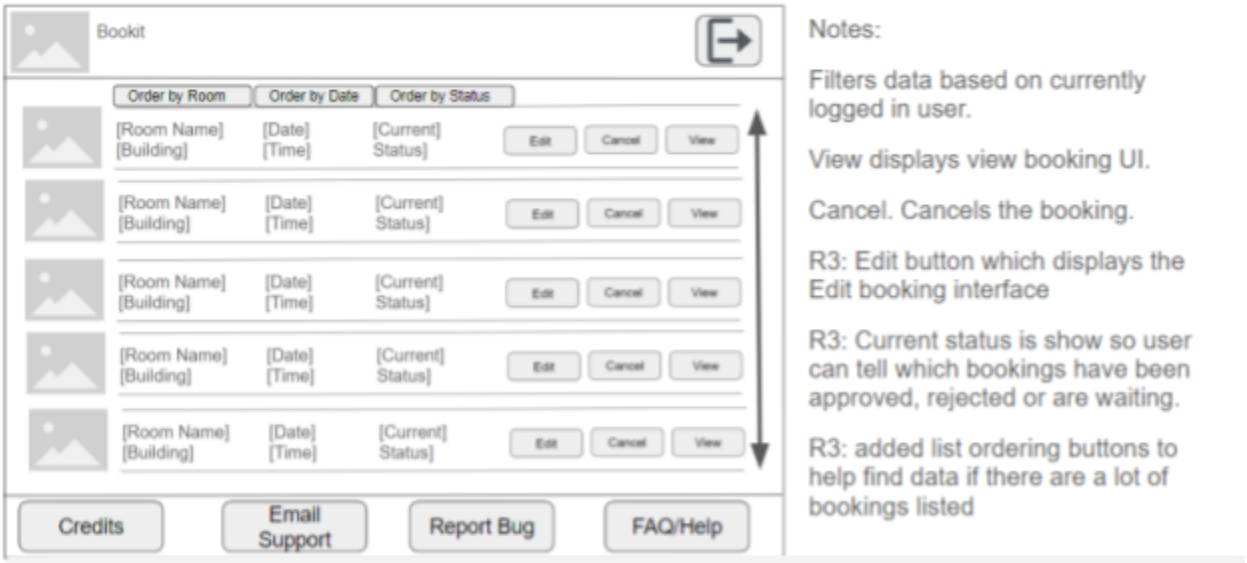


Figure 42: Digital Prototype 2 current bookings coordinator

UI: View Requests - from Co-ordinator Menu

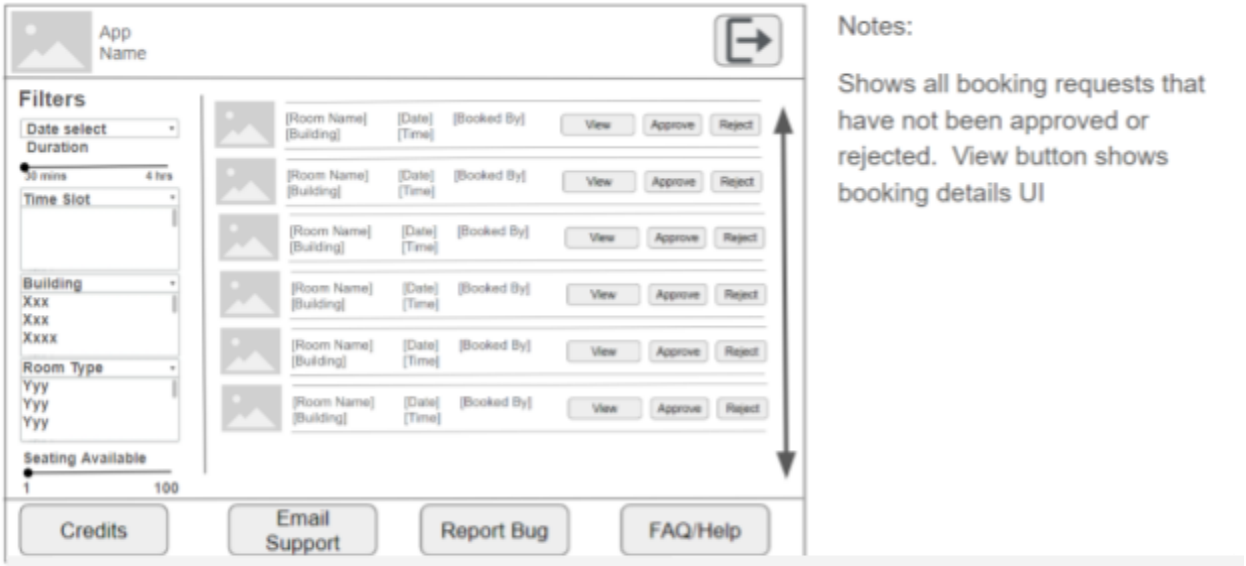
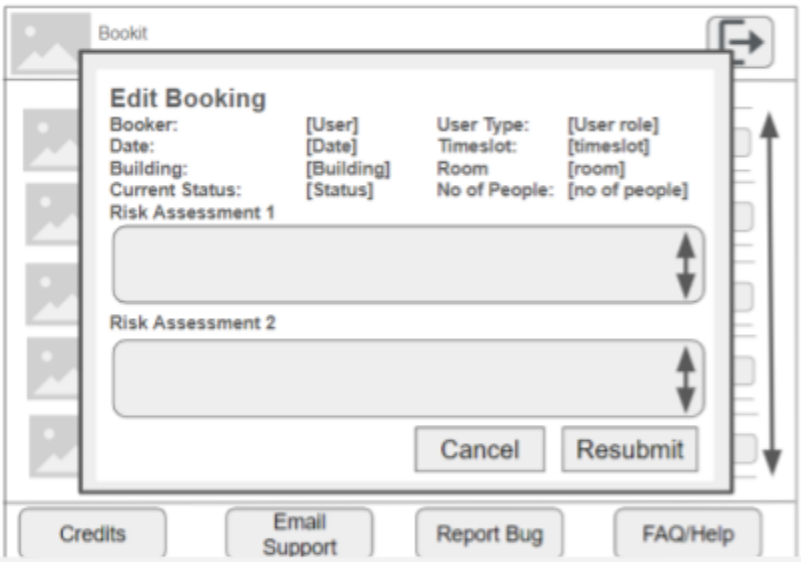


Figure 43: Digital Prototype 2 current bookings requests for coordinator

UI: Edit Booking



Notes:

R3: Allows editing of booking details, for instance if a request was rejected then the details could be changed and be re-submitted.

R3: Cancel ignores all changes and returns to the view bookings page.

R3: Resubmit saves the details and changes the status so the coordinator can reassess

Figure 44: Digital Prototype 2 edit booking

UI: View Accepted Requests (From coordinator menu)

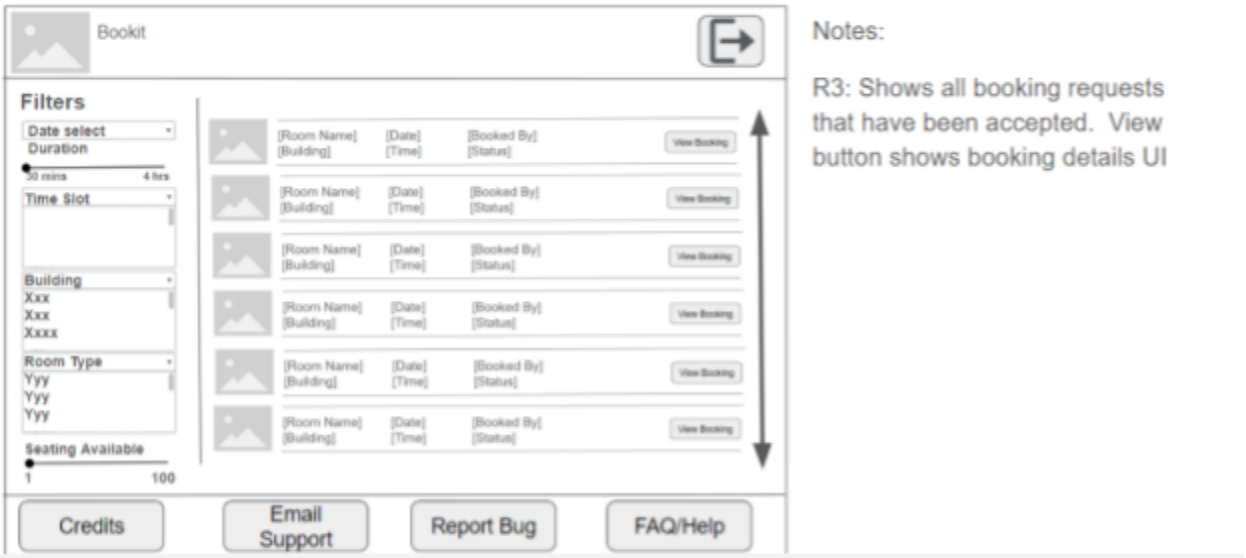


Figure 45: Digital Prototype 2 accepted bookings

UI: View Booking Detail / Requests Details from co-ordinator lists

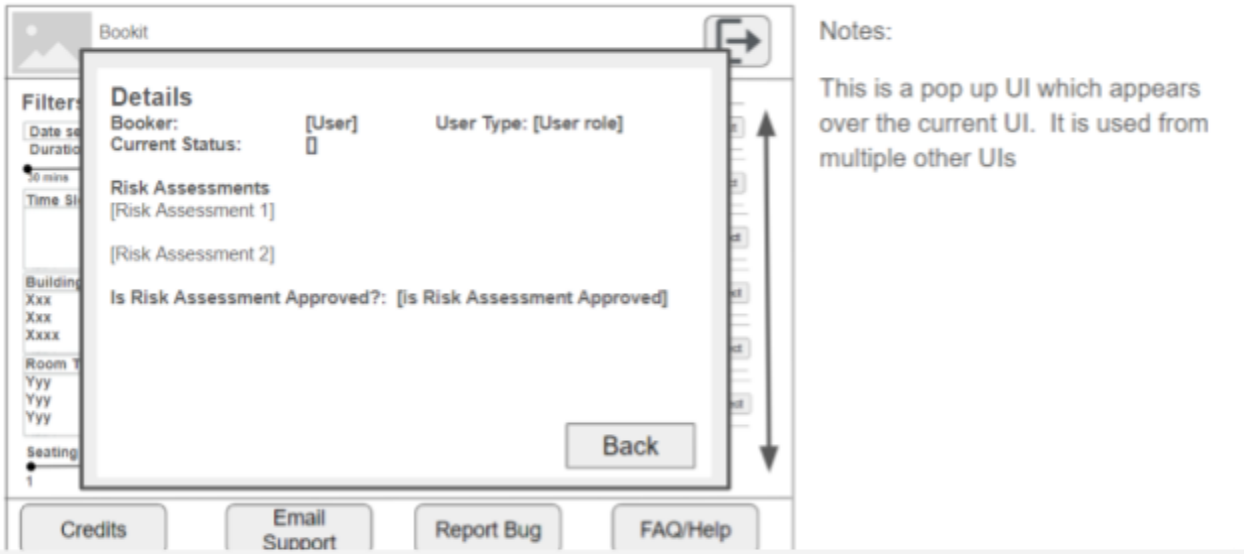


Figure 46: Digital Prototype 2 booking details

Pros

1. Missing elements are now there

Cons

1. Needs styling
2. Needs fonts that are appropriate for users with Dyslexia

3. Needs accessibility considerations - high contrast for example
4. Can the calendar not be a select, can it be a scrollable calendar?

Cons - Solutions

1. Consider in R4
2. Consider in R4
3. Consider in R4 - there may be space when the high fidelity versions are done

Prototype, Set #4 (R4, application flow, hi fidelity wireframes and clickable prototype)

Summary

Based on the feedback from R3 we constructed the high fidelity wire-frames. A lot of this feedback was about styling. We also created an interface flow based on the user interface. Additionally we made an interactive prototype. We did this because it's easier for users to click on a prototype and understand the flow than it is for them to see it on paper and fully understand it. The clickable high fidelity prototype is here:
https://app.uizard.io/p/739c5641

Note only the top row of each list has buttons that are clickable, all other buttons, sliders and select boxes are clickable. If you click anywhere on the prototype, the buttons that are clickable highlight briefly.

It is anticipated that in the next phase we will be able to "re-use" some of the interfaces by passing in different parameters for example for the data set, and the role. By doing this we will be able to dynamically the target interface for some of the action buttons. By doing this we can avoid situations where we have two edit booking interfaces (one for a coordinator and one for a Lecturer or Society Leader). We will consider this in more detail in the next phase of the project.

Design and Style Considerations

Font

We have decided to use Trebuchet MS as our main font. This is because it is a member of the Sans Serif family. Trebuchet MS is recommended by The British Dyslexia Association as one of the most readable fonts [7].

We will also use font settings that align with guidelines set out by the British Dyslexia association to ensure Dyslexic users have the best possible experience:

">19px, with at least 3.5x the inter-word spacing to the inter-letter spacing. We will also aim for 1.5x line spacing and aim to make our headings 20% bigger than main body text" [7].

Figure 47: High fidelity fonts element

Colours

We selected the following colour palette for the application, and checked them for high contrast for accessibility (AAA compliance).

We used blue for action buttons, green for help buttons.

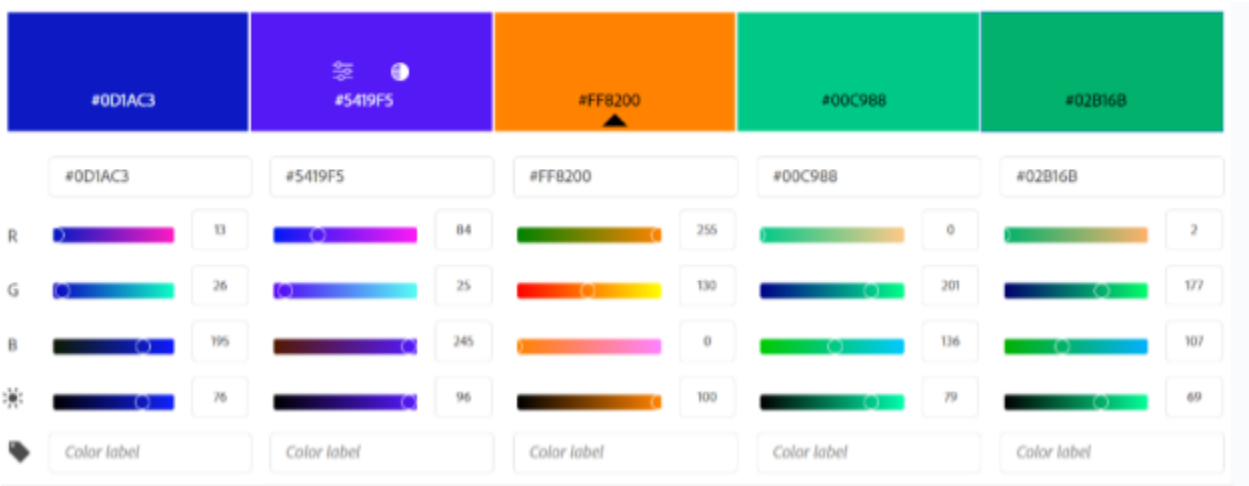


Figure 48: High fidelity colour palette

AAA Compliance Check

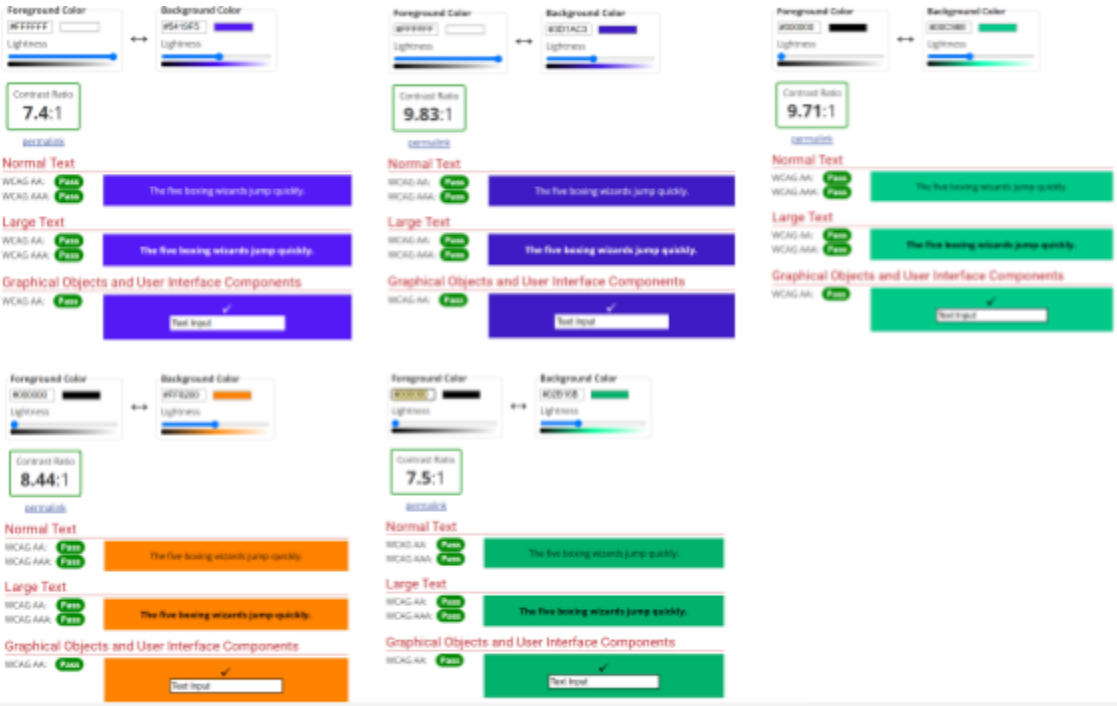


Figure 49: High fidelity compliance check

Interface Flow - https://app.uizard.io/p/739c5641 is the link for the clickable prototype

Figure 50: High fidelity interface flow

High Fidelity Wireframes

UI: Login

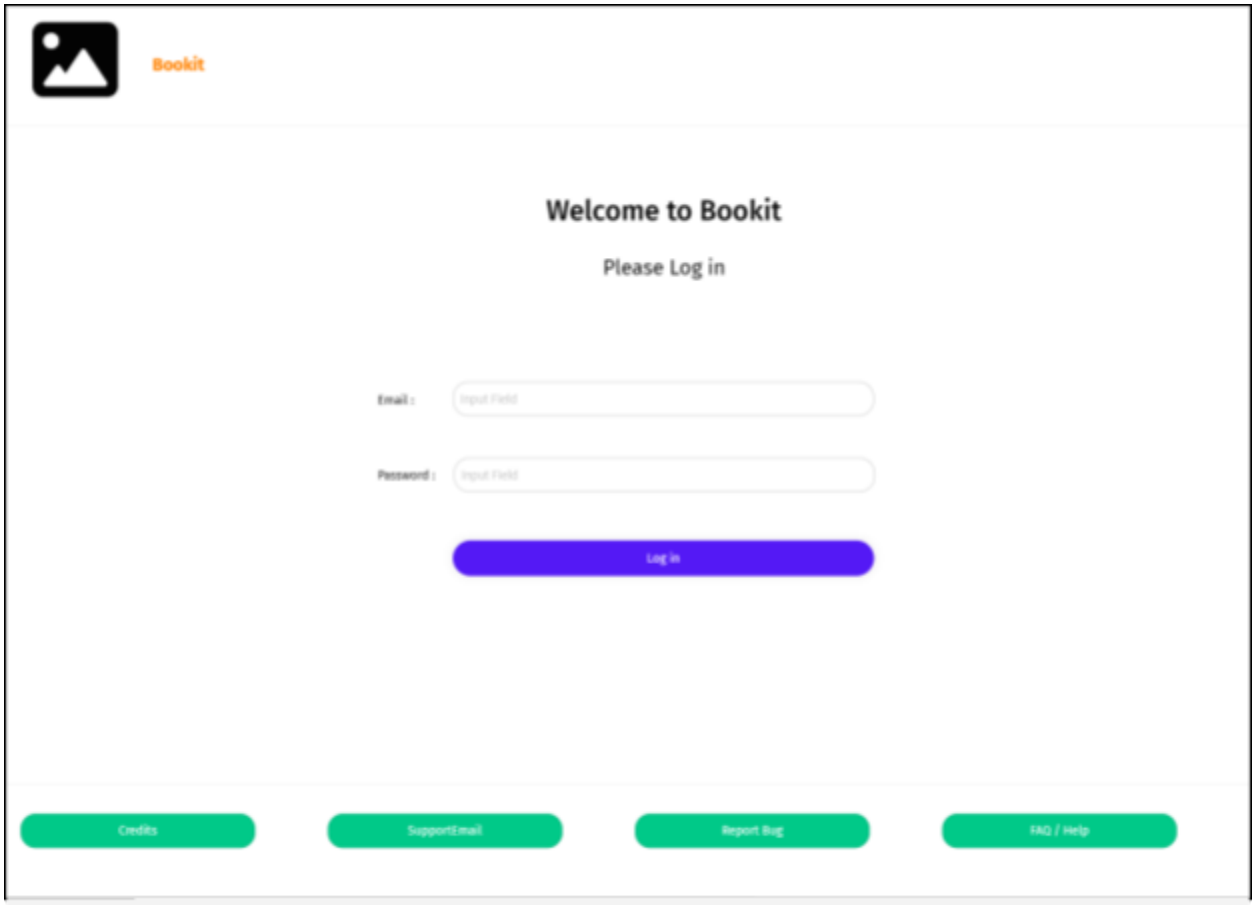


Figure 51: High fidelity UI login

UI: Main menu for lecturer or society leader

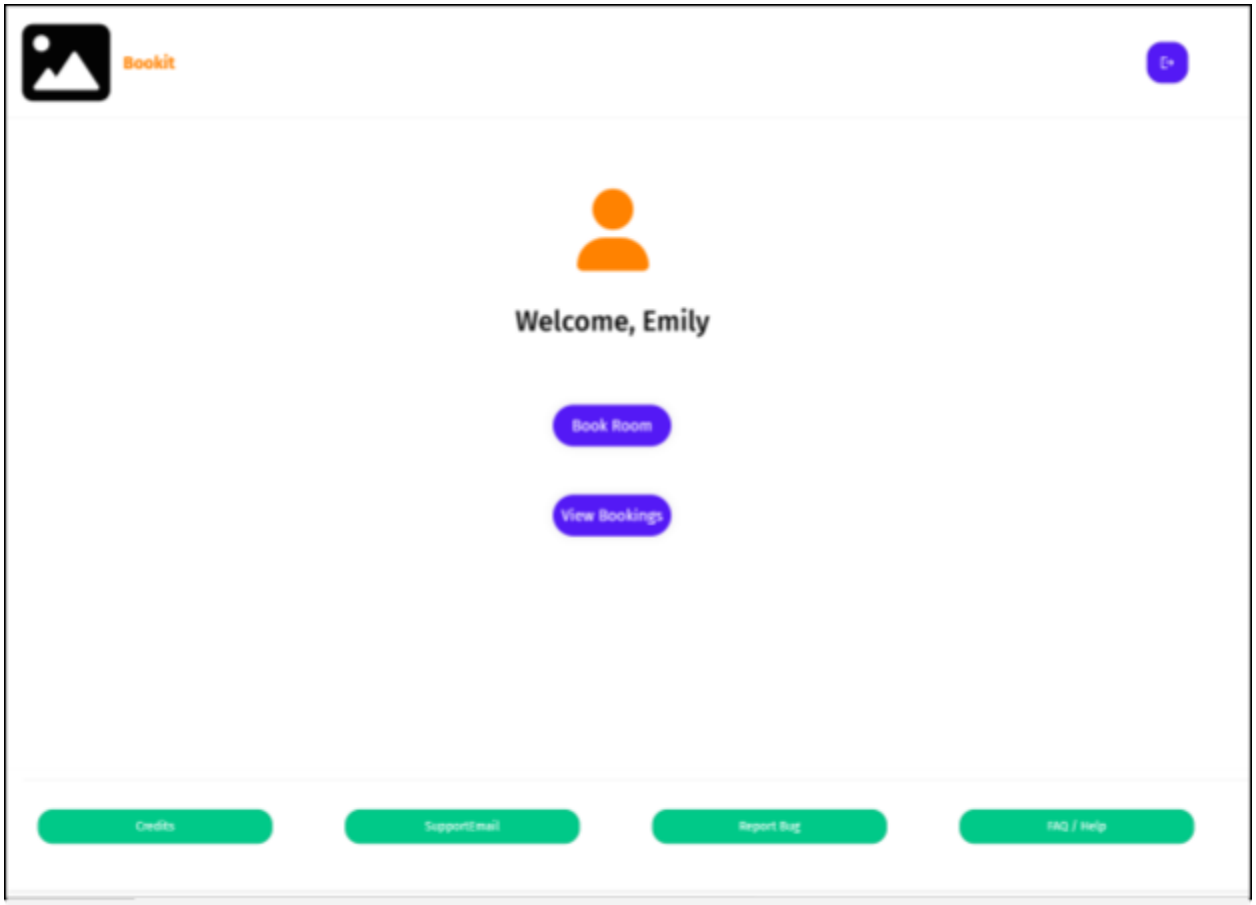


Figure 52: High fidelity UI main menu for Society Leaders and Lecturers

UI: Main menu for Coordinator

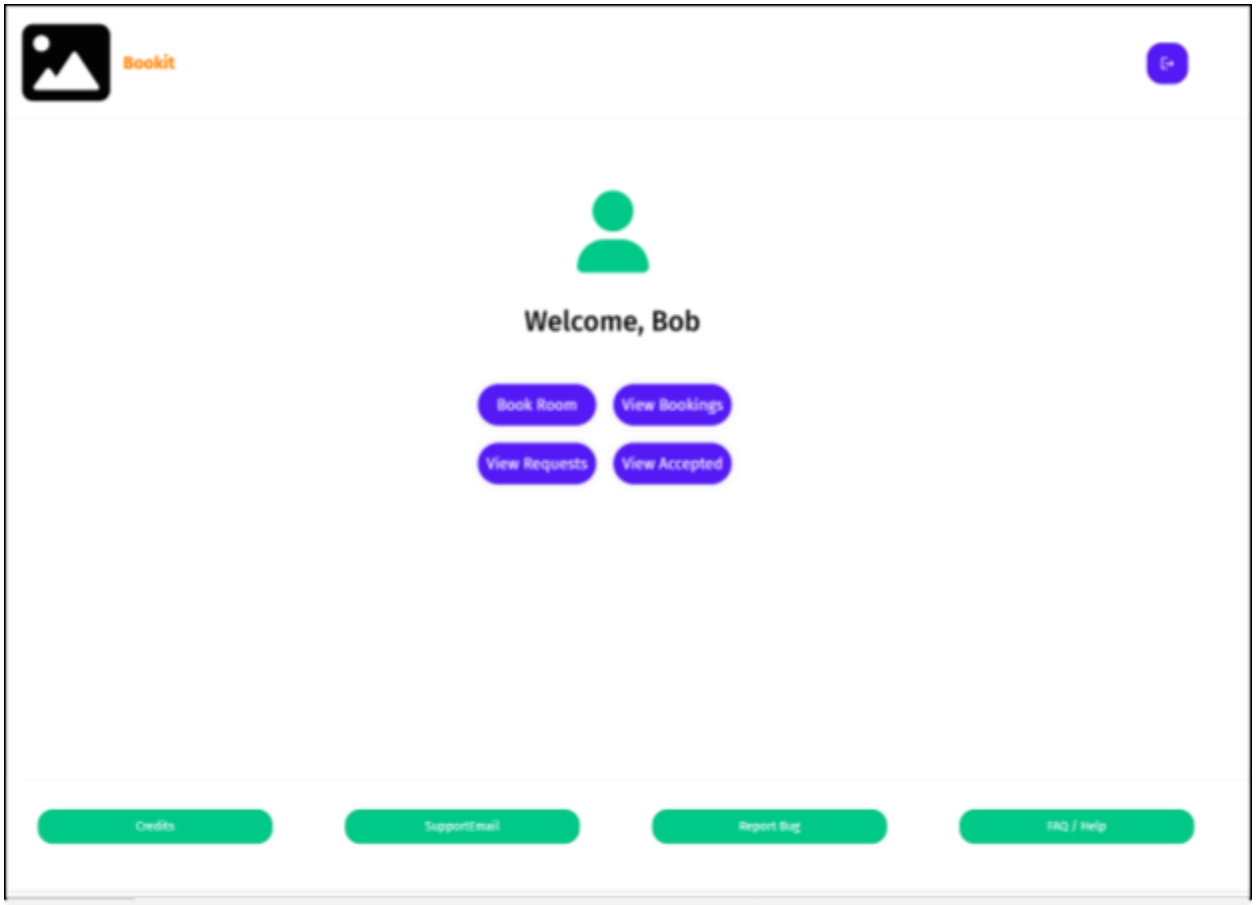


Figure 53: High fidelity UI main menu for coordinators

UI: Book Room - Find Room and time. Room Bookings

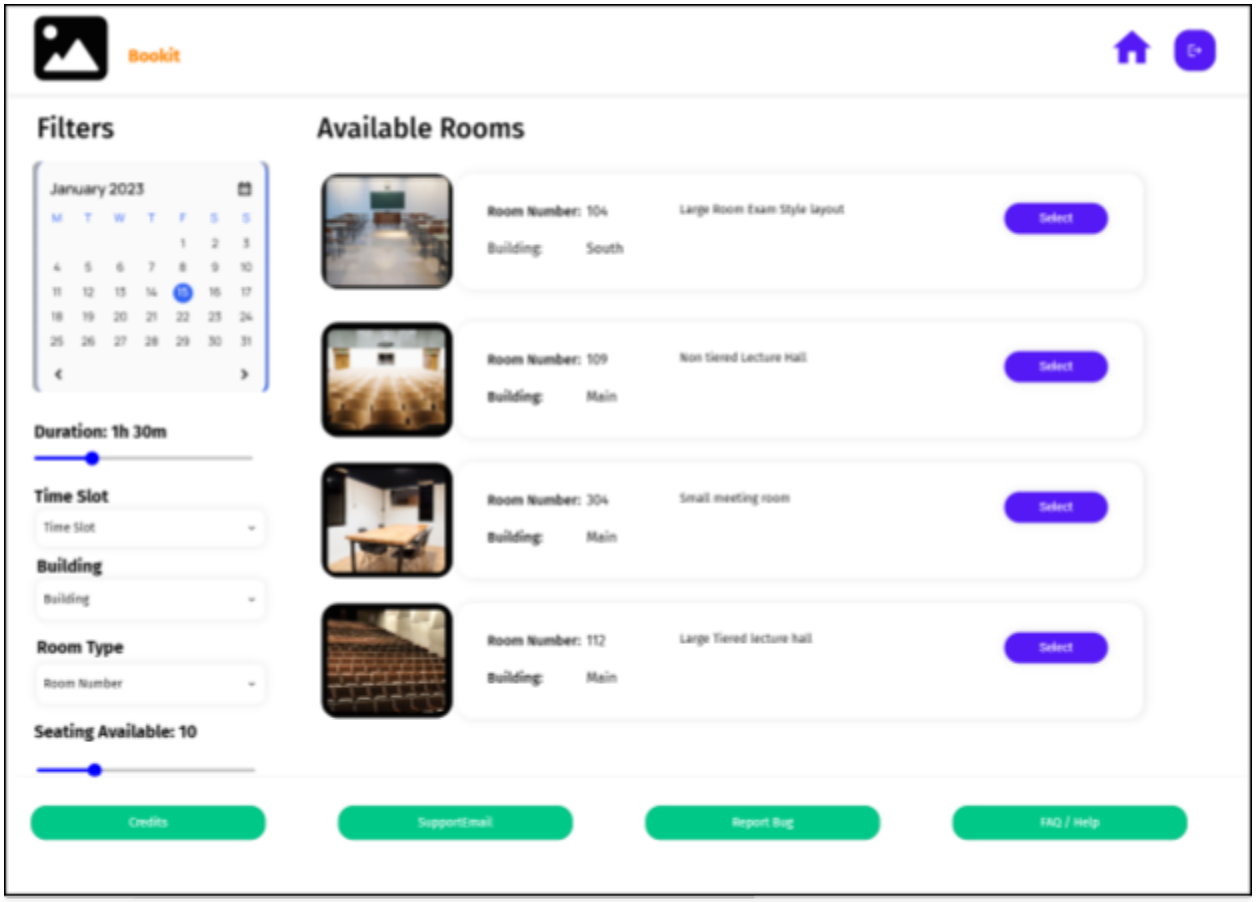


Figure 54: High fidelity UI for booking rooms

UI: Enter Risk Assessment and Complete Booking

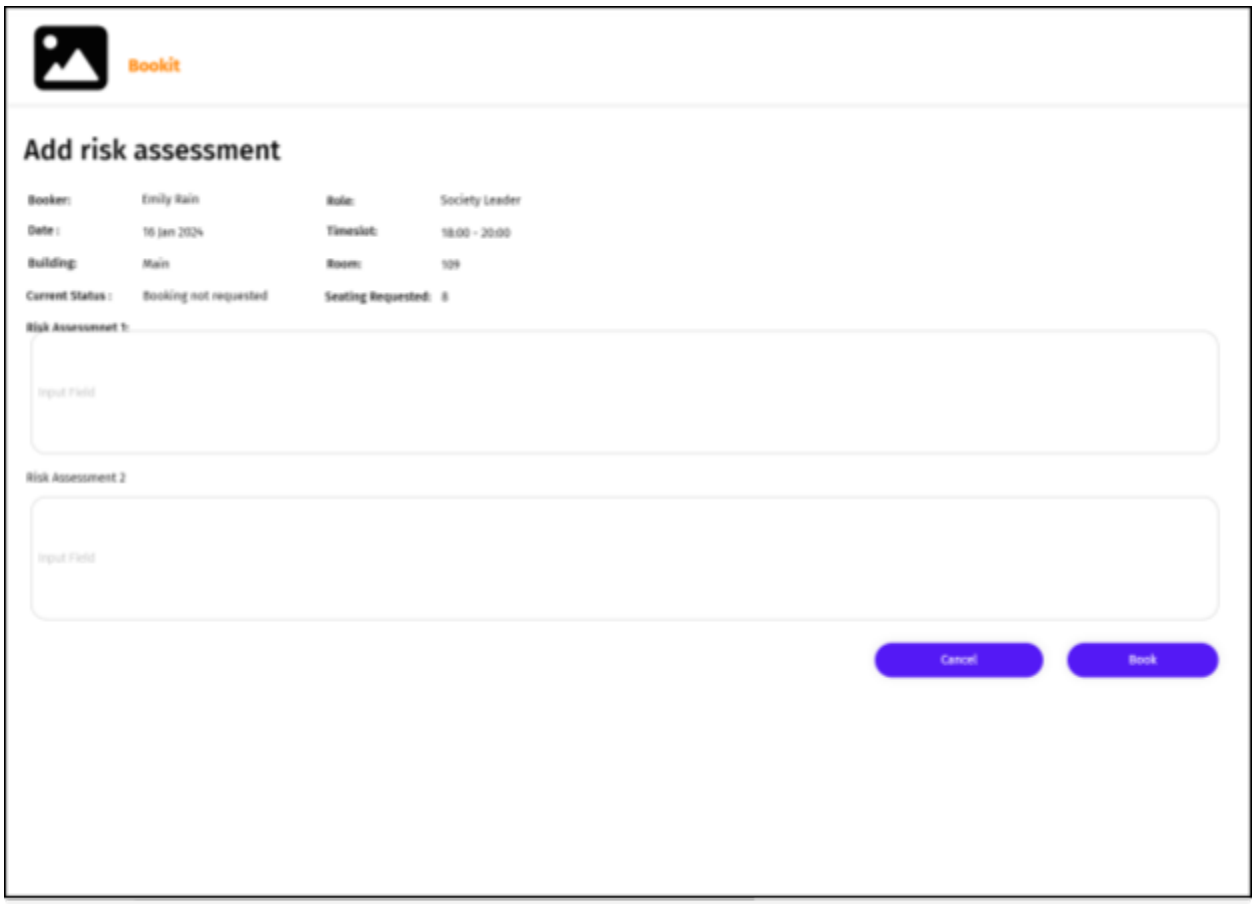


Figure 55: High fidelity UI for risk assessment and booking completion

UI: View Current Bookings - Society Leader / Lecturer

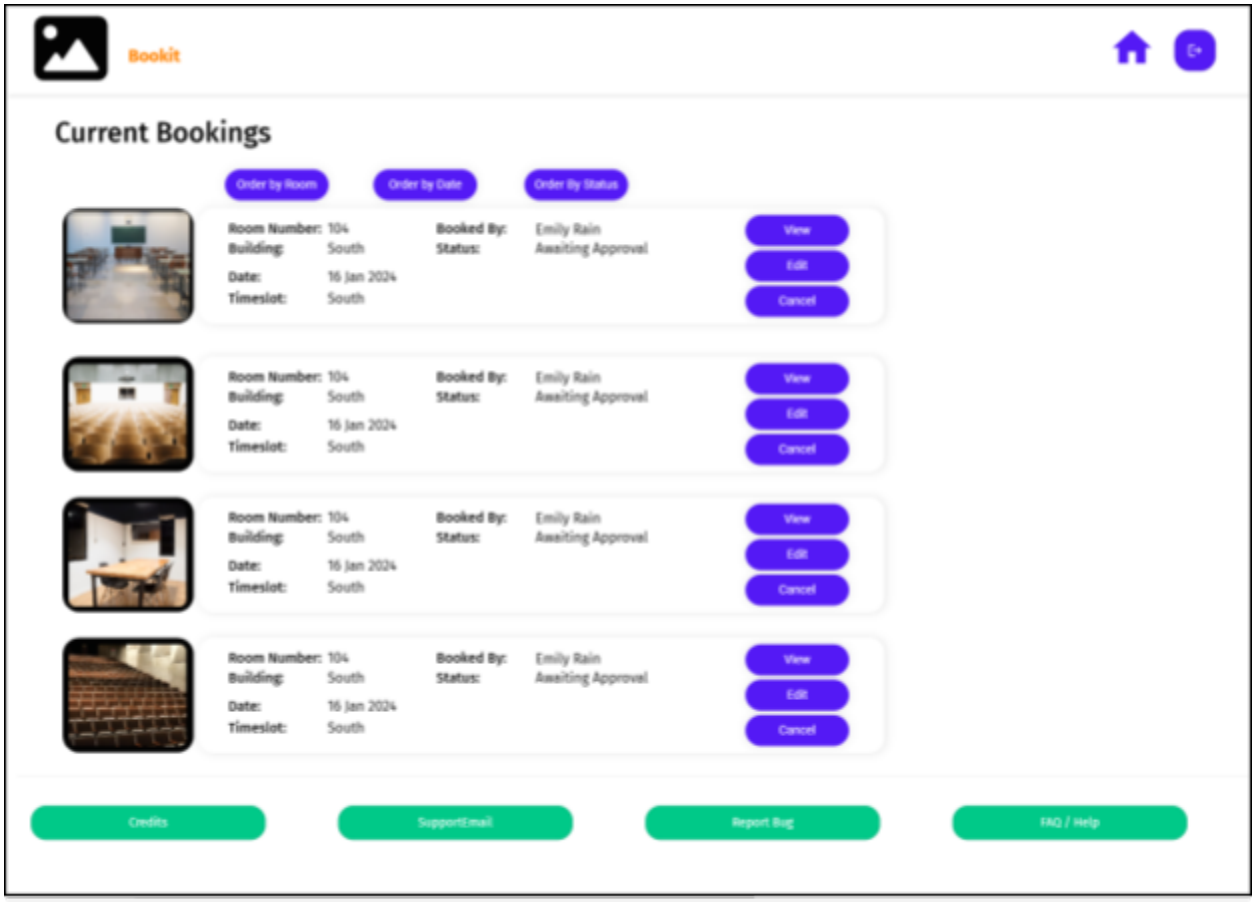


Figure 56: High fidelity UI for Society Leaders and Lecturer's current bookings

UI: Edit Booking

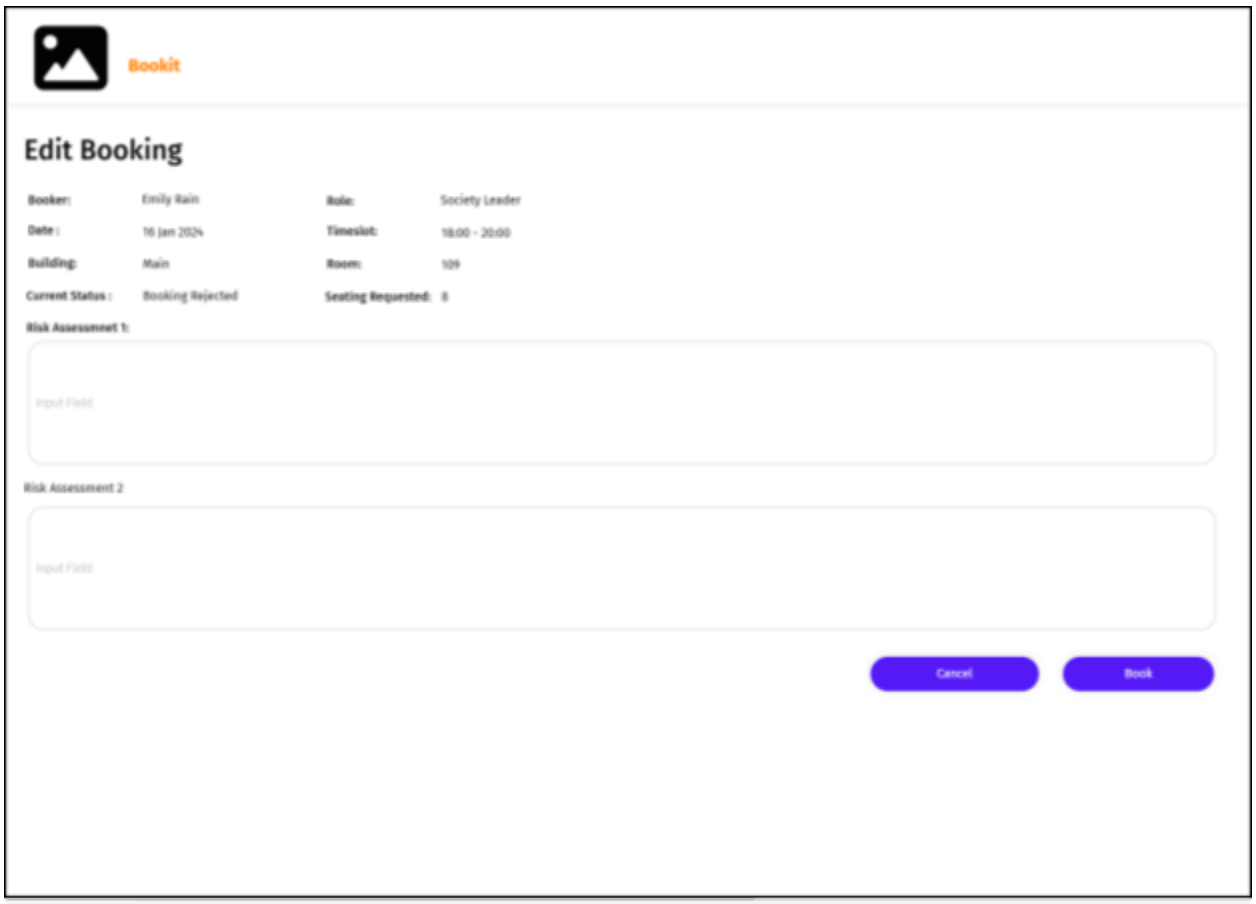


Figure 57: High fidelity UI for editing bookings

UI: View Current Bookings (From coordinator menu)

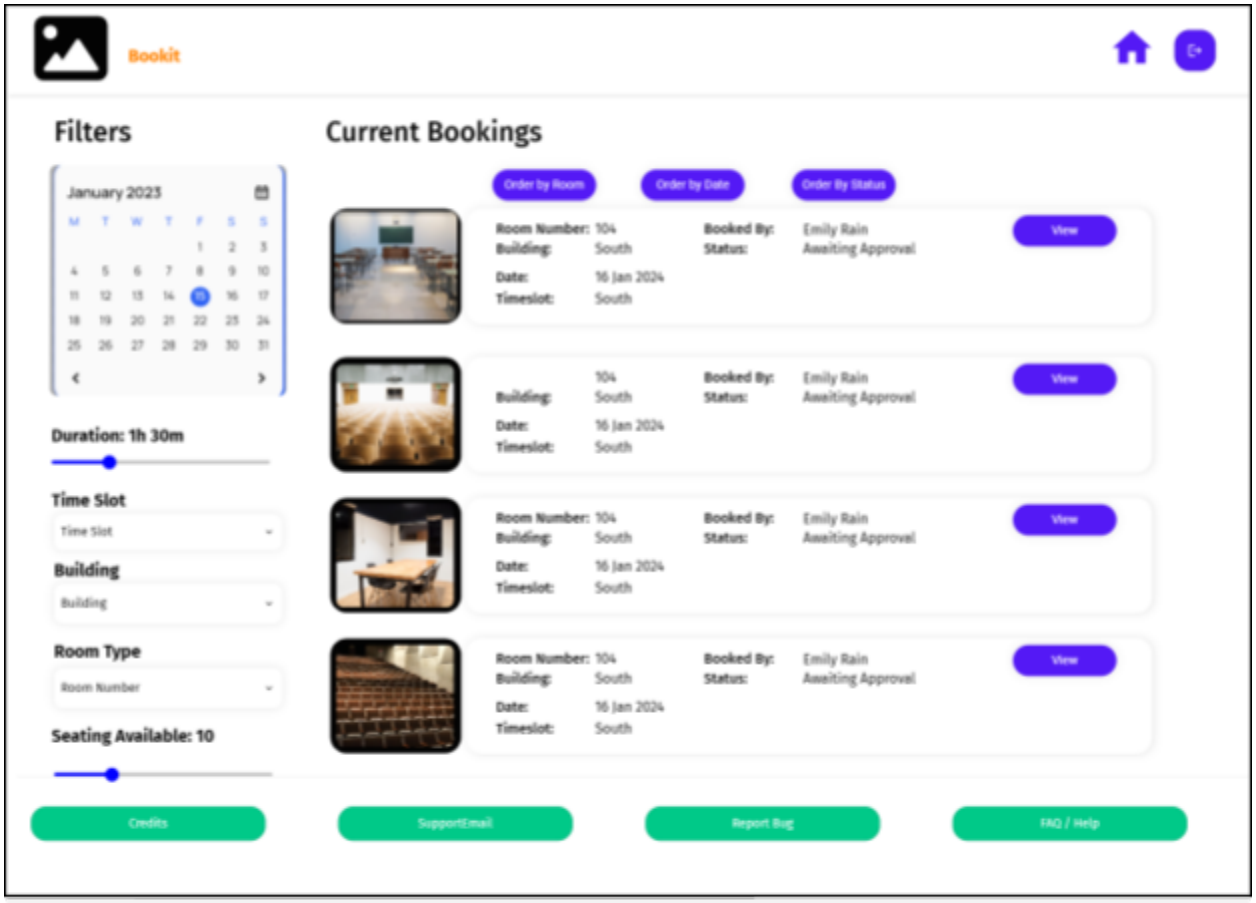


Figure 58: High fidelity UI for coordinators view on current bookings

UI: View Booking Detail / Requests Detail

Figure 59: High fidelity UI for viewing booking details

UI: View Requests - from Co-ordinator Menu

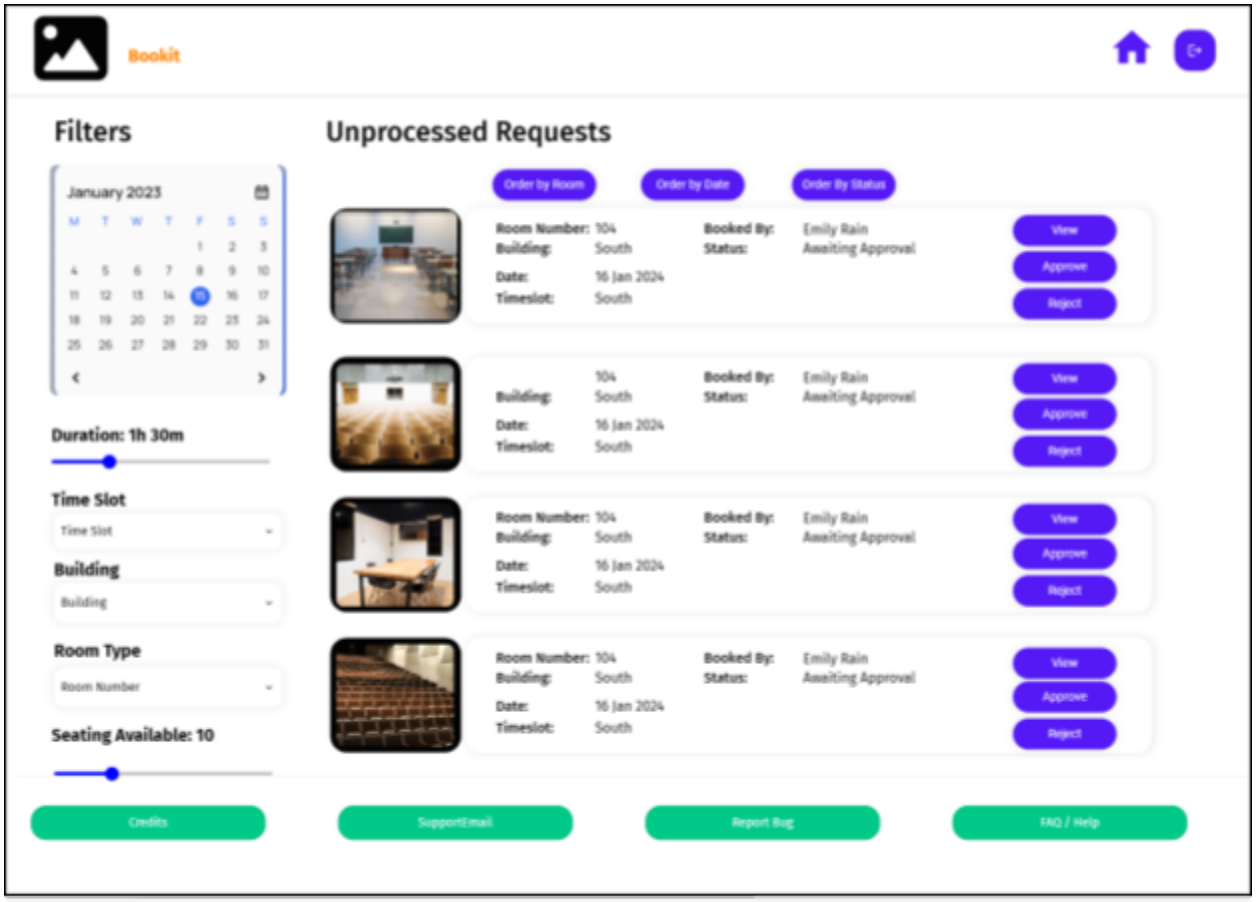


Figure 60: High fidelity UI for coordinator viewing booking requests

UI: View Accepted Requests - from Co-ordinator Menu

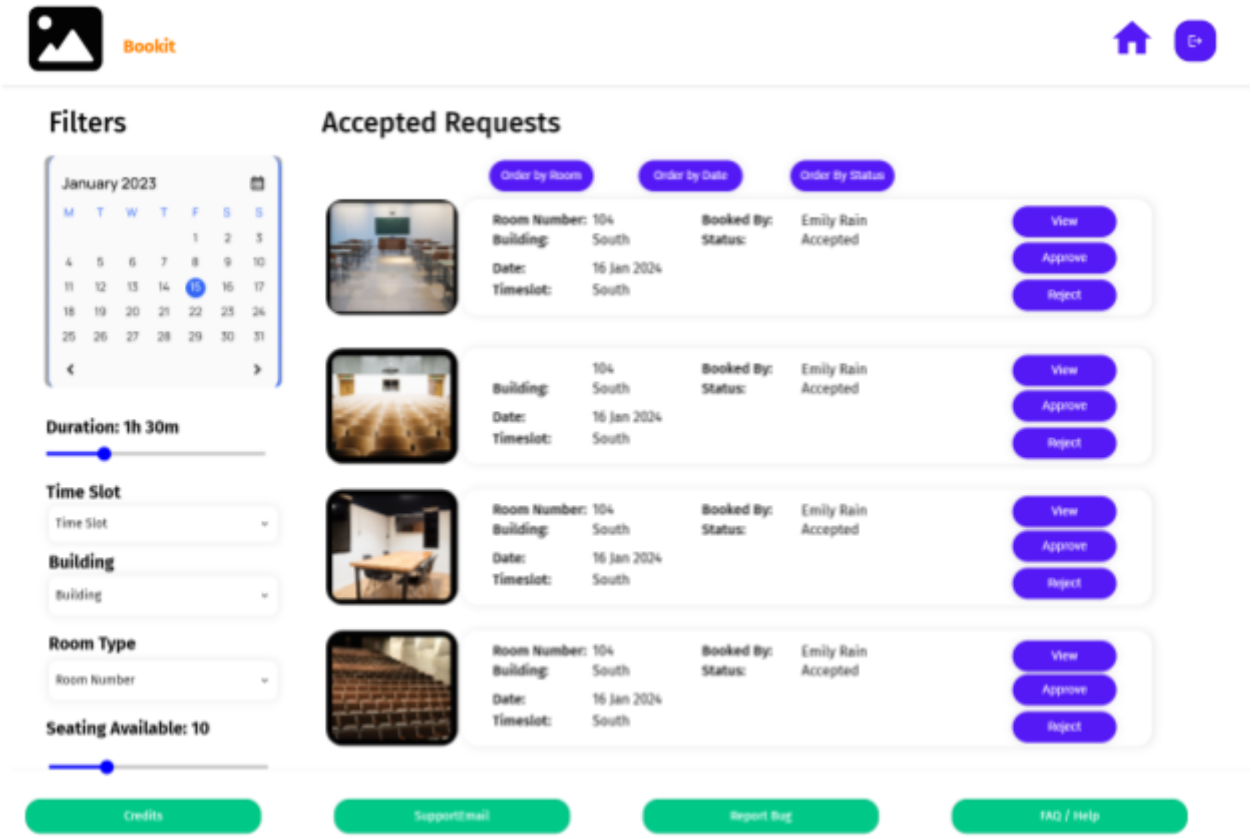


Figure 60: High fidelity UI for coordinator viewing accepted requests

Prototype Entity Relationship Diagram

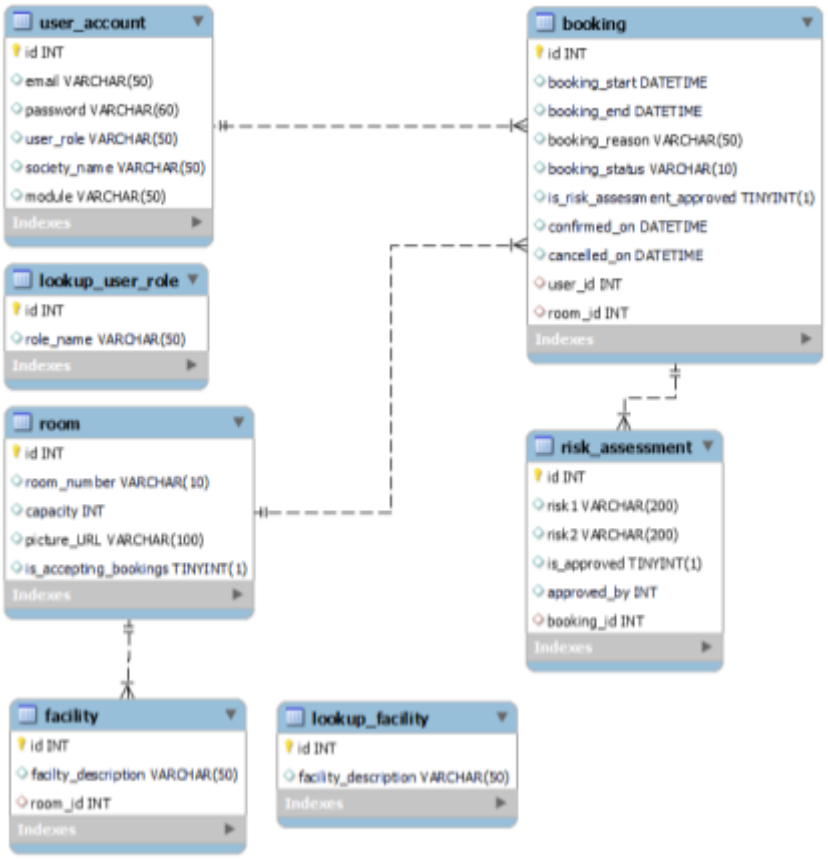


Figure 61: Prototype entity relationship diagram

These tables were based on the class diagram. The “Booking system” class was created as the booking table. The Society Leader class and the Lecturer class are subclasses of user_account, their details have been included in the superclass user_account, in the database design. These could have been created as subtables but it is simpler to have these in the super class table.

Foreign keys and relationships were added. Some changes to field names were made to be consistent with database naming standards. id was used for the primary key and {table}_id for foreign keys. A table was added for facilities allowing for a one to many relationship from room to facility. Added lookup_user_role and lookup_facility so drop down lists could be pulled from the database and wouldn’t need to be hard coded in the application. Additionally made sure that the password field was long enough to hold a bcrypt password hash which will be 60 characters.

Known limitations: A user account can only have one role, so a Coordinator and a Lecturer must have two separate accounts. This is not expected to be an issue in the application context.

Prototype Database table scripts

Prototype mySQL database scripts were generated based on the design work completed.

```
CREATE DATABASE roomBooking;
USE roomBooking;
CREATE TABLE user_account (
    id INT AUTO_INCREMENT,
    PRIMARY KEY (id),
    email VARCHAR(50),
    password VARCHAR(60),
    # types of role are Society Leader, Lecturer, Coordinator, Facilities Staff, Health and Safety Staff
    # created a lookup table lookup_user_role to hold valid values for this
    user_role VARCHAR(50),
    # from subclass SocietyLeader
    society_name VARCHAR(50),
    # from subclass Lecturer
    module VARCHAR(50)
);
CREATE TABLE room (
    id INT AUTO_INCREMENT,
    PRIMARY KEY(id),
    room_number VARCHAR(10),
    capacity INT,
    picture_URL VARCHAR(100),
    is_accepting_bookings BOOLEAN
);
CREATE TABLE booking (
    id INT AUTO_INCREMENT,
    PRIMARY KEY(id),
    booking_start DATETIME,
    booking_end DATETIME,
    booking_reason VARCHAR(50),
    booking_status VARCHAR(10),
    is_risk_assessment_approved BOOLEAN,
    confirmed_on DATETIME,
    cancelled_on DATETIME,
    user_id INT,
    # a booking must be related to one user account - added foreign key
    FOREIGN KEY (user_id)
        REFERENCES user_account(id),
    # each booking is related to a room
    room_id INT,
    FOREIGN KEY (room_id)
        REFERENCES room(id)
);
CREATE TABLE risk_assessment (
    id INT AUTO_INCREMENT,
    PRIMARY KEY(id),
    risk1 VARCHAR(200),
    risk2 VARCHAR(200),
    is_approved BOOLEAN,
    # this will be an id from the user_account table from a user that has role health and safety
    approved_by INT,
    # a risk_assessment must be related to one booking
    booking_id INT,
    FOREIGN KEY (booking_id)
        REFERENCES booking(id)
);
```

```
# created new table for facility which was not in the draft class diagram
CREATE TABLE facility (
    id INT AUTO_INCREMENT,
    PRIMARY KEY(id),
    facilty_description VARCHAR(50),
    # each facility is related to a room
    room_id INT,
    FOREIGN KEY (room_id)
        REFERENCES room(id)
);
# facilities that can be in a room
CREATE TABLE lookup_facility (
    id INT AUTO_INCREMENT,
    PRIMARY KEY(id),
    facility_description VARCHAR(50)
);
# types of user role
CREATE TABLE lookup_user_role (
    id INT AUTO_INCREMENT,
    PRIMARY KEY(id),
    role_name VARCHAR(50)
);
# Create the room booking user to be used by the code and give it access to the database
CREATE USER 'roombookinguser'@'localhost' IDENTIFIED WITH mysql_native_password BY 'roombooking2027';
GRANT ALL PRIVILEGES ON roomBooking.* TO 'roombookinguser'@'localhost';
```

Figure 62: Prototype database table

Functional specification

This chapter will define our objectives and operations for the project overall while making a clear note of our considerations for usability for each objective.

Key:

- Specification
- Usability considerations

- System allows users to see which rooms are available and when - from this users can also see which rooms are booked.
    - Introduce a graphical representation for booked and vacant rooms.
- Allow for users to select a room and time to book said room.
    - Introduce a map system and a drop down menu(es) to select rooms and times for desired booking(s). Booking should also be cancelable by the one who made the booking in a similar fashion.
- Level of clearance should be logged to account - giving educators more privileges than society leaders.

    - Society leaders will have to go through a more rigorous confirmation process - this involves the coordinator confirming a proposed booking through their end of the application.
- Introduce a templating system for society leaders so as to streamline the risk assessment process.
    - This will utilise drop-down menus and other GUI elements to speed this process up.
- Room search should present the user with a list of facilities for a selected room.
    - In a similar fashion - the user can also search for rooms via the desired facilities.
- Confirming rooms should be done with a click of a button by the one who made the booking - otherwise the room is made vacant again.
    - Reducing input here increases usability of this function

Technical architecture

As a team we have decided to make a web-based application. Not only because it is the area of development in which we feel most confident, but also because it would increase accessibility for our stakeholders and proposed users as a web-based application would coincide with other web-based applications commonly used by educators and society leaders at Goldsmiths. Furthermore, a web-based application is not platform-specific, and can therefore allow for our users to access our app on both desktop and mobile devices, a common necessity among university applications like our proposed system [9].

The front end of our system will be handled with HTML, CSS and JavaScript. Our team is well versed in these languages - and this allows for more efficient allocation of resources, wherein we can focus on planning and development rather than learning a new language for this project.

For this reason also, we have decided to use MySql to handle our relational database model. Not only is it easy to use for secure storage of details as mentioned previously [22], but it is also the database system with which the members of our group are most familiar.

We have decided to focus more on making our web application responsive to different screen sizes and systems, and can therefore confidently discard the need for any platform-specific development systems such as Android Studio for mobile. We have also made considerations for streamlining this process - by including a styling library like Bootstrap - which is well documented and renowned for being effective and robust - we can more effectively allow for responsiveness in our application.

For our middleware, we have opted for NodeJS and Express to render our pages. This is not only due to the aforementioned familiarity our group already has with these systems, but also due to the fact that NodeJS allows for simple access to the servers [10] provided by Goldsmiths, which we plan to use to host our application and supporting database.

System Overview

As we are developing a system which relies heavily on a database - including interactions with said database and frequent amendments to it - we have agreed to follow the model-view-controller architecture. We decided to follow this architecture as opposed to the standard three-tier architecture as it follows a more triangular structure - with the model (our database) being able to feed directly back to our view (our rendered EJS pages) - while the controller (our JavaScript code) handles all other logic and communication between these systems [23]. This adequately follows the notion of separation of concerns and can therefore be considered an adequate architecture for our purposes.

We considered using the Single-Page Architecture - but felt this would add needless complications - especially when considering our login system - which would be far more simple to develop as a separate page while still retaining full functionality. Furthermore, we agreed that the pages would need to look different enough for each function of our system that the Single-Page architecture would not be suitable [3].

Leading on from this, we decided as a team that we needed a consistent structure to our relational database system to facilitate both our login system, as well as handling bookings, and also dealing with amendments to room and facilities data. Therefore, we decided to create prototypes for our database system that fit the mould of the model-view-controller architecture (see “Prototypes”).

Functional requirements

Functionality available to ALL USERS	
Log In	Log into account using a basic username and password system - done with text entry fields.
Create a booking	Click on a button to create a new booking
Select a date and time	- Use a calendar system to select the day on which the booking will take place. - Highlight the desired time for the booking using a drop-down menu. - Rooms that are unavailable at the given time frame will be greyed out in the selection.
Select room	- Rooms that are available won’t be greyed out on the GUI - and can be clicked on to select them. - Selection of rooms can also be done with a drop-down menu which is populated with the

	available rooms at the selected time frame. - Once the room is selected - a list of facilities will be provided for the user to aid in their decision for confirmation.
Search for rooms	- The user can tick boxes for the facilities they'd like to have for their desired room. - Rooms which do not have all the facilities specified will be greyed out in the GUI and not present in the drop-down selection list.
Confirm booking	The booking can be confirmed by clicking a button prior to the time of the booking - if confirmed the booking remains throughout the curation - otherwise the room is marked as vacant again.
View bookings	The user can see a graphical representation of which rooms are available at any selected time - occupied rooms will be greyed out.
Cancel booking	Users can select a booking they have made previously and mark the room as vacant - indicating they no longer wish to use the booking they had previously made.
Log out	Log out of account and stop accessing app.

Figure 63: Functional Requirements for ALL USERS

Functionality available to SOCIETY LEADERS	
Template risk assessment.	- Society leaders will be presented with a series of text fields and drop down menus when they wish to make a booking - this will allow them to input relevant data for a risk assessment when it comes to booking their desired activity. - The UI will consist of text fields for relevant information, tick boxes to indicate common risks, and drop-down menus to highlight options for more generic information.

Figure 64: Functional Requirements for SOCIETY LEADERS

Functionality available to EDUCATORS/LECTURERS	
View timetabled bookings	For regularly scheduled educational activities, such as

	lectures - the educator can see which room is in use for which activity and when. - The educator will use a calendar system to select the day - and a list of rooms and when they are in use for their regularly scheduled educational activities will be presented.

Figure 65: Functional Requirements for EDUCATORS/LECTURERS

Functionality available to COORDINATORS	
View list of booking requests from society leaders	Coordinators will be able to view a list of bookings that have been created/requested by society leaders alongside their attached risk assessments. They will be presented with information regarding which room, when it is being requested for, by what Society and which Society representative made the booking.
Confirm/deny bookings	Coordinators can click a button to confirm or deny a booking - at which point the Society leader is notified accordingly through the application.
Access and amend room data	Coordinators will have full access to the room data - where the given facilities can be updated/changed according to the current situation.

Figure 66: Functional Requirements for COORDINATORS

Non-functional requirements

Non-functional requirement	**How will we approach this?**
Performance - aiming for fast load times	- Reduce our file count - we should aim to keep our algorithms efficient and use code wherever possible instead of graphics. - When considering dynamic changes - we will make use of EJS to effectively and quickly display changes to the user.
Capacity - need to account for a large amount of information on a large number of rooms	- Host all data in a MySQL database to efficiently store and retrieve data on rooms/facilities. - Remove redundancies - only keep the necessary data - such as location, facilities, and room name. - Store data for all required rooms.

Security - certain users should not have certain functionality available to them	- Login system - password and username provide a layer of security for individual users. - Introduce a clearance level for users - only if you have the given clearance level can you edit the room data or decline a requested booking, for example.
Reliability	- As a web-based application, it should be hosted on a stable server with a consistent internet connection.
Scalability - we need the means to add more buildings/rooms in the future	- Storing rooms in a table-database structure based on the building will allow us to create more rooms and space representations using the same code.
Maintainability	- Coordinators can access the room data and add more as necessary - direct access to query and amend the database.
Accessibility - how can we make sure the people that need to use our system can actually use it?	- Images should have alt text. - Colours should account for common forms of colour blindness. - Text should be in a legible font and be large enough so that users with poorer eyesight can still read it.
Data integrity - how will we keep our data consistent and usable?	- Sanitise inputs - room data must follow a strict template (name, location) - and facilities assigned to a room must come from a predetermined selection. - Coordinators will only be able to input certain data types when amending the room data (e.g., only inputting alphanumeric characters when changing the room name).

Figure 67: Non-functional Requirements

Research summary

We decided as a group that we should follow the same iterative process when it came to gathering requirements that we planned to take for the entire project [4]. Therefore - we proposed the following structure to keep to when performing requirements gathering:

- Perform activity with users to gather raw data and information.

- Amend the MVP to fit the narrative gained from the previous requirements gathering activity.
- Reconvene with stakeholders to gain their opinions on the amended MVP and how we can better sculpt it to fit our user's needs.

This approach, while not only allowing us to ensure we are fulfilling our user's needs during the design process, also allows us to take a 'funelled' approach to gathering our requirements. While we agreed that open-ended interviews with stakeholders and potential users would be the best place to begin our gathering of requirements (for the aforementioned reasons), we also concluded that further methods that refine our specifications further were necessary for developing an MVP that was comprehensive for our goal for the system while also fulfilling all our user's needs. For this, we decided on questionnaires [8], followed by card sorting [18]. To garner an idea of what the system may include and to begin mapping out what features would go into our application and how respectively.

Interviews

From conducting our interviews (see "Interview transcripts" - Appendix), we managed to gather the following key themes that arose:

Thematic analysis:

- Availability
    - A big issue that arose in the interviews was a lack of information which rooms were available and when
        - Solution: a way to easily search for rooms - with a clear display on which rooms are and are not available.
    - Changing rooms relies on an unresponsive and unreliable communication system. The lack of cooperation leads to room bookings that do not satisfy the needs of the booker.
        - Solution: confirmation system - coordinators receive all bookings and can easily confirm or deny them.
- Timing
    - Two weeks waiting for a room booking to be confirmed causes complications.
    - Having to fill out a full risk assessment each time is a waste of time.
        - Solution: templating risk assessments alongside a simple confirmation system for room bookings should streamline this process.
- Usability
    - Google Forms has limited success at fulfilling the room booking purpose. It has the same issues surrounding repetition; while also being tedious to complete each time.

- Solution: confining everything to one application which has functionality to handle all aspects of the booking application will aid in this. Furthermore - it will give us the opportunity to add a simple and intuitive UI to this process to make booking rooms easier.

Amendments to MVP

From our thematic analysis - we decided that the following amendments should be added to R2:

- Template risk assessments
- Account system to log level of clearance
- Edit bookings - at which point they can be re-examined by the coordinator

Questionnaires

Room Booking

To further our research and analysis of what users may expect from a booking application, we extended into primary research where we created and conducted our own online questionnaires. This would better our discussions during prototype creations and we could take all of this information into account. This questionnaire has been given to Goldsmiths university students, including graduates but also students and graduates outside of Goldsmiths and other universities.
The questionnaire has a variety of both open and closed questions in order to gain direct answers but also have more in-depth answers.

Questions

After users input their name and email to ensure answers weren't fake, skewered or duplicated, the questions were split into 3 sections: likes & dislikes, wants & needs, expectations & experiences. This was to help us filter information when reviewing what changes needed to be made to prototypes.

Likes and Dislikes

1. What features do you like the most when using a booking system?
2. What makes an app/system user-friendly and how important is it?
3. Can you give some examples of booking apps that are user-friendly and why?
4. What are common problems you face when booking?
5. Are there any features within a booking app that are problematic or frustrating to use? If yes, why?

Wants and Needs

1. What is a common feature you would improve within booking systems?

2. Are there any functions that you wish were more common within booking applications? Why?
3. Do you like it when booking-apps can integrate with your other platforms? (Google calendar, Paypal payments, etc.) Why?

Expectations and Experiences

1. How do you feel about customer service and support provided by the booking applications?
2. How helpful are user reviews to you? (Answer from 1 to 5)
3. Overall, how easy is it to navigate booking apps? (Answer from 1 to 5)
4. Have you ever stopped using a booking app because of its poor functionality and interface? If so, why?

Answers - Summary

After looking over every response we have taken note of key ideas that the majority of users want that are realistic and relevant to implement into our own booking system for our university.

Likes and Dislikes

1. Users appreciate features that create a seamless experience while giving priority to user conveniences. User-friendly clear interfaces and navigation with a pleasing visual design is needed. In functionality, users want to easily select a room and select a specific time of use so the system is efficient. Users also tend to like visuals and short amounts of text - keeping it concise. This allows for quick comprehension and stops lengthy decision-making time. Being able to know real-time availability of rooms to avoid double-bookings and further issues.
2. User-friendliness tends to come from its design and elements such as colours, spacing and fonts. This will have to be experimented with heavily in our CSS. The process of booking should be easy without causing confusion, errors or extra unnecessary steps.
3. The DVSA driving test booking system was mentioned for its ease of use. Trainline and EuroStar simplify ticket booking with clear directions on platforms and times. Goals pitch hire provides easy navigation, and Airbnb stands out for its visual appeal, comprehensive property listings, and transparent information presentation. Booking.com has a clean interface, search filters, and user reviews for a straightforward booking process. Expedia's clear layout and bundling options enhance the user experience, while Uber, though not a traditional booking app, excels in simplicity - offering ride requests, real-time tracking and easy payment methods. All these systems have in common the clear information presentation and efficiency when being used.

4. Booking experiences can be ruined by various issues including but not limited to queues, aggressive fonts and old boring web designs. Delays during bookings can affect room confirmation, booking cancellations and slow response times to booking requests are also factors within the frustrating experiences. Lack of information such as sold out dates/times, visibility for available and unavailable times and unclear booking confirmations and cancellations can also add stress and encourage the user to take extra unnecessary steps to double check their procedures when it may have been unneeded. A huge concern users may face is finances - specifically the protection of financial information. Our university booking system will not need financial issues but it is best for our system to implement a privacy protection system within our project.
5. Although many responses have been said there aren't any problems, some other problems stood out to be problematic and irritating. One key feature would be having quick cancellation if needed - the hassle to call or email as a resort can be annoying and frustrating. Lots of pop-ups/notifications can be disruptive when booking, during the booking process it can be accidentally clicked which can ruin a booking process. Having fast responsiveness was a common response - it seems it is common for people to want to complete their booking process as fast as possible.

Wants and Needs

1. An improvement users would want to see implemented is a less bothersome/concise notification system: direct with confirmation and cancellation information to eliminate any uncertainties. A feature that could come with this is setting notification preferences so only relevant notifications can be sent to specific users. Visuals such as icons and colours can be implemented for more user-friendly and efficient booking experiences. Integrating calendar views to check availability across days and months is also convenient to have for users.
2. Clear direction on whether a room is available or already booked, a live-chat to contact customer service and perhaps an FAQ if not. A user mentioned being able to view the occupants of the room: this may cause a privacy concern so perhaps this will be for co-ordinators only. Perhaps a feature to show alternatives, personalised options based on previous bookings.
3. Integration between apps such as google calendar and paypal is very appreciated - especially any payment integrated features. It caters towards the ease of usage. Calendar apps are able to set reminders and streamlined payments options to make sure that payments are safe and quick during the transaction process.

Expectations and Experiences

Opinions on customer service are varied - most opinions lean towards neutral or negative because they either do not need to contact customer service or customer service has not

satisfied the user with what they needed. To have positive customer service experiences, users expect quick solutions, assistance and effective communication.

All users find user reviews extremely helpful, rating it 5/5. One user being the exception rating it 4/5.

All users find navigation around booking apps to be neutral or good - rating it either 3/5, 4/5, or 5/5. There have been no answers that are below 3/5.

Most users do not stop using a booking app because of poor functionality and interface; those who do mention that things that stop them from using apps are complex navigation, technical glitches, lack of features, and poor design. Some also stopped using booking apps because there isn't any availability when trying to book something.

University Room Bookings

To be more specific with our primary data research, another questionnaire was made specifically for university students who take role in societies as a committee member. Those members who have the responsibility of booking rooms within their respective university - Goldsmiths and others - have answered their own personal opinions and ideas with their booking experiences.

Questions

1. How satisfied are you with your current room-booking system? (Answer from 1 to 10)
2. How difficult is it to use the current room-booking system? (Answer from 1 to 10)
3. What aspects of your current room-booking system do you find to be user-friendly?
4. What challenges do you face when using your current room-booking system?
5. What features and functions do you believe is missing from your current room-booking system?
6. What features and function do you believe is useless/should be removed?
7. What devices do you use when booking a room with the system put in place?
8. Do you have any privacy or security concerns with your system?
9. Is your room-booking system integrated with any of your other university tools/platforms? (Answer Yes, No or Unsure/Haven't thought about it)
10. Is your room-booking system integrated with any of your other university tools/platforms? (Answer Yes, No or Unsure)
11. If yes, explain what it is and how it's integrated

Answers - Summary

1. Majority of the answers are negative, with only one answer towards the positive side.
2. Mainly neutral, one answer towards the negative side.

3. The current room booking experience isn't widely known for its user-friendly aspects according to its users, although the sign-up form has been mentioned for its clarity and for being straight-forward and clear.
4. Some challenges users face is the requirement of filling out the form and then waiting, which creates flexibility problems - with addition to separately filling in risk assessment forms.
5. It is missing detailed specifications of each room, including equipment and capacity which hinders planning for events within the room. A calendar displaying available rooms with a timetable is a needed feature. Absence of automated systems to confirm and remind is a limitation also.
6. While some of our stakeholders are unsure of what features are useless or unwanted, some suggest that emailing extra forms such as risk assessments to the coordinator is a waste of time.
7. All users are using computers to book university rooms.
8. Most answers are unsure or haven't thought about it, only very little concerns to privacy.
9. Most answers claim their systems are not integrated, only one answer is unsure.

Conclusion - Key Themes

After reviewing all the answers given from our questionnaires, it seems like majority of users focused on these points:

- Clarity: all functions of the booking system should be understandable
- Simplicity: limit the amount of steps necessary for a quick and easy booking experience
- Information: real-time information such as when rooms are already booked + information about the room's space and equipment. Possible FAQ page.
- Easy cancellation - a system that doesn't require hassle and stress when cancelling a booking if needed
- Reduce pop-ups: limit pop ups to what's necessary, have notification settings adjust to its user if notifications are needed
- Use examples of user interface from current existing booking applications, most systems are rated neutral or good
- External forms for our booking system should be limited or completely removed - possible integration if needed?
- Include user-reviews for rooms: all users find this useful
- Users did not really mention the following:
- Need or importance of integration between apps such as calendar

Proof of Questionnaires
Room Booking

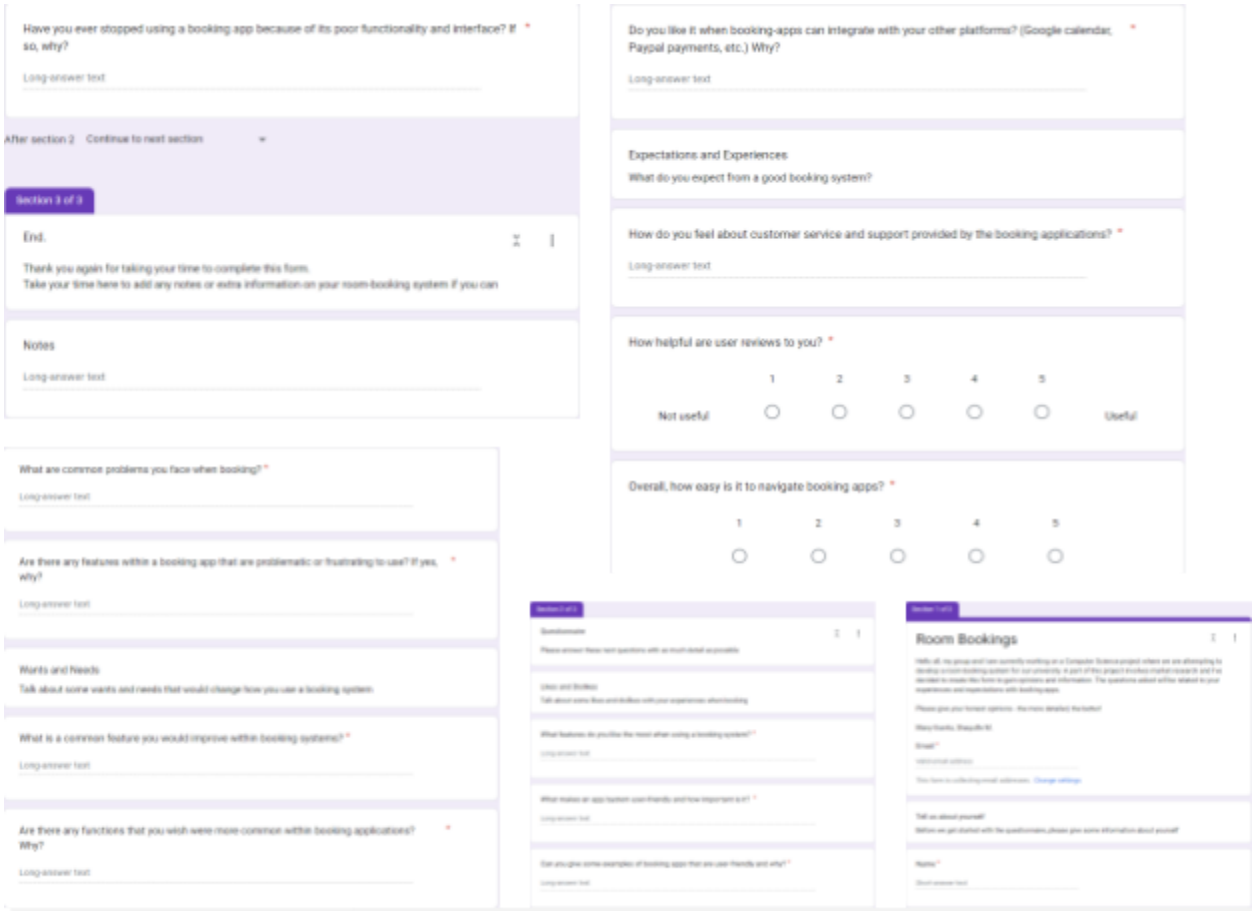


Figure 68: Room Booking Questionnaires

University Room Booking

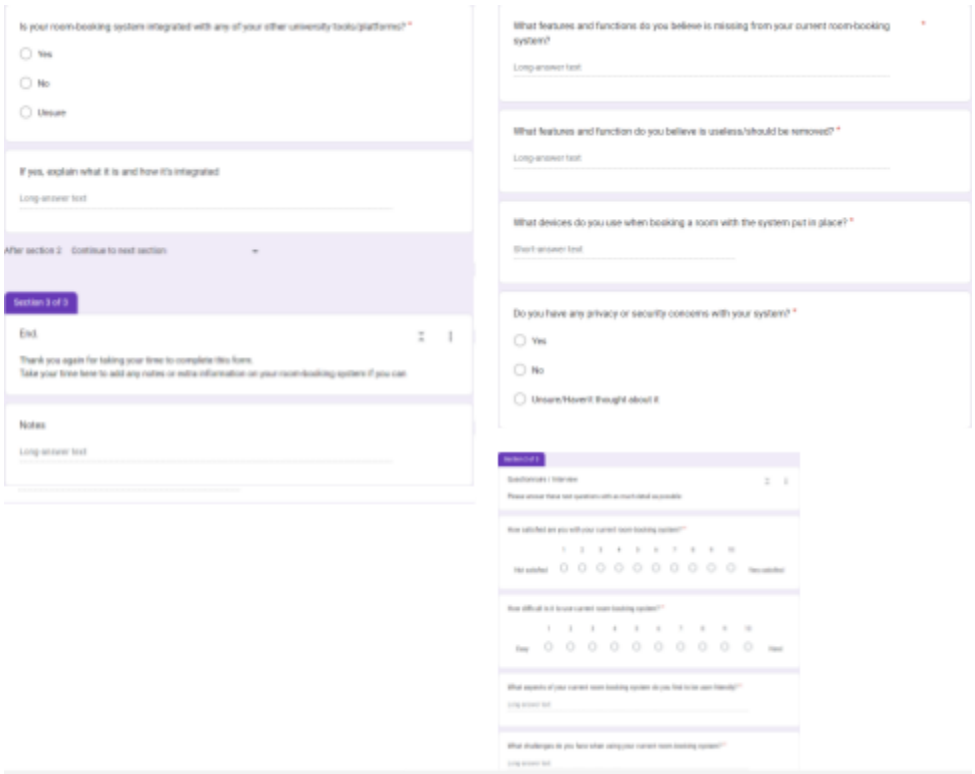


Figure 69:University Room Booking Questionnaires

Card sorting

After the production of R2 we conducted a card sorting activity to see how our stakeholders and related parties could give us an idea of how they would group features of our proposed solution [18]. We created 20 separate cards and nine separate categories (see Appendix - “Card sorting resources”) so that we could grasp a better idea of what layouts we could consider when creating our high-fidelity wireframes for R3. From this, we gathered 10 participants; all of whom are at college/university; and asked them to sort all of the cards into the given category.

For this activity, we used online software called UXtweak - as it is free, and collects data automatically. This data from the activity is listed below:

Cards	Booking process	Navigation	User account	Search	Integration	Notifications	Performance	Accessibility	Usability
Room type	70%	0%	0%	20%	10%	0%	0%	0%	0%
Select room	60%	10%	0%	10%	0%	0%	0%	10%	10%
history of bookings	60%	0%	20%	0%	0%	0%	0%	0%	20%
Availability	50%	0%	0%	0%	0%	10%	20%	20%	0%
Date/Time	40%	0%	10%	10%	10%	20%	0%	10%	0%
Facilities	40%	30%	0%	10%	20%	0%	0%	0%	0%
Calendar	40%	0%	0%	0%	20%	20%	0%	10%	10%
Capacity	30%	0%	0%	10%	10%	0%	30%	0%	20%
Map view	0%	40%	0%	0%	0%	0%	10%	10%	40%
Home page	0%	30%	30%	0%	10%	0%	0%	0%	30%
user profiles	0%	0%	80%	0%	10%	0%	0%	10%	0%
Register	10%	10%	70%	0%	10%	0%	0%	0%	0%
login page	0%	10%	60%	0%	10%	0%	0%	20%	0%
Search	10%	10%	0%	80%	0%	0%	0%	0%	0%
Filter	20%	0%	0%	50%	10%	0%	10%	0%	10%
scalability	0%	0%	0%	0%	30%	0%	30%	10%	30%
alerts/notifications	0%	0%	10%	0%	20%	70%	0%	0%	0%
Loading times	0%	0%	0%	0%	0%	0%	80%	0%	20%
response time	0%	0%	0%	0%	10%	20%	40%	0%	30%
Font/colours	0%	0%	20%	0%	10%	0%	0%	60%	10%

Figure 70: Card sorting data

From this; we saw that it would make the most intuitive sense to place the majority of features in a single booking screen as this was the most popular category for the card sort. Furthermore, we should place a larger portion of our focus on streamlining the navigation process and allowing for easy access of details and information in a user's account.

This activity also showed us that usability is not a direct consideration of our proposed users; so we should still aim for a high level of usability and an easy to use UI for our system, while also ensuring we implement as many useful features as possible.

Performance also seemed to be an important consideration for this system when it comes to the opinions of our proposed users. Therefore, we have decided to make it a priority when it comes to non-functional considerations.

Ethical audit

As we will be handling user data such as passwords, we aim to ensure that we will have adequate security for each user's information. We aim to take an approach that will minimise the need to take extensive user details for each account - and this is particularly

important in a university setting; where there is lots of data attached to each individual. Ensuring that we only take the data needed to allow for the functionality of our app while ensuring that that data remains secure and at an adequate level of encryption is crucial. Login credentials will be mandatory. Not only to ensure security of user data but also to ensure that the centralised data stored in our main database is only accessed by those with clearance logged to do so. Our prioritisation of data security is further demonstrated by our choice to use mySQL - as it is reputable for having robust security and being simple to maintain [22]. Data leakage could have serious implications for a University application, and that is another reason why we are taking security as a priority when it comes to ethical considerations.

While our system may not be designed specifically for people with disabilities, we believe that to be as inclusive as possible we should ensure that our application has a colour scheme that complies with AAA contrast accessibility [27] - while also having a font that is legible, and large enough for people with dyslexia (see “Design - Prototyping; Prototype, Set #4 (R4, application flow, hi fidelity wireframes and clickable prototype)). We aim to have a UI that is accessible to the greatest number of users possible.

We believe our ethical considerations should help our application be more accessible and secure for the students at Goldsmiths.

Evaluation plan

Type of test	Test name	What will be tested?	Test cases for development		Measurement of success	
			While developing	After conclusion of development	While developing	After conclusion of development
Functional	Systems test	The whole application/system is tested vigorously against our defined functional requirements. The actual functionality, capabilities, and shortcomings of the whole system can be judged against the defined system specification.	Test whole classes/ constructor functions and interactions between the classes. The user should be able to make books - bookings should be saved in the database, etc.	Run the entire system as an application. Test all features for functionality and shortcomings against the functional requirements.	Individual specifications must function in isolation.	The entire application functions as expected.
Functional	Unit tests	Each specific section of code is tested for its specific purpose - be	Test lines and functions and methods	Test interactions between methods and functions.	Specific sections function. E.g.,	Specific functions must work as

		that specific methods, functions; or queries, etc.	individually to ensure they work. E.g., testing the buttons to request a booking.	E.g., ensuring that a booking request is correctly received by a coordinator and can then be confirmed or denied.	do the buttons redirect you to the correct pages?	intended. E.g., does clicking the button to request a booking send that request to coordinators correctly?
Functional	User functionality tests	The user will test the system and ensure it flows correctly as is according to their needs.	Stakeholders and users can test the individual modules of the application to ensure we are on the right track for developing the application flow.	Stakeholders and users can perform an entire black box test of the entire application to test user flow and functionality.	The application must align with the user's needs as to the point it is at in development.	The system must solve the overall problem and satisfy the overall concept.
Non - Functional	Stress	Test to see how the system performs when under pressure that exceeds that of the predefined specification.	Make many bookings from many users at the same time to see if they can all be received by the coordinators.	Many users all trying to book the same room at the same time while other users browse for that time and see if the room is available.	Test for when the process starts to leave data behind or inserts the data incorrectly.	Test to see how much stress the system can be put under before it fails.
Non - Functional	Usability	Test will determine how usable the system is - will a standard user be able to make easy use of the system for its intended purpose?	Usability can be tested for individual modules.	The whole application can be criticised for usability.	The feature mapping makes sense - the way things are laid out for the user is intuitive and does not cause confusion.	Test usability with all stakeholders - and ensure the system is entirely usable for people with varying levels of aptitude in IT.
Non - Functional	Accessibility	Test will determine whether the system accommodates disabilities and other users with different needs when it comes to operating an application like this one	Different users should test individual modules of the system to ensure we retain functionality while accommodating for accessibility.	The whole system can be tested by different users with varying accessibility needs.	Particular aspects of the system must be accessible for all users.	The whol;e system must account for accessibility for all users.

Figure 71:Evaluation Plan

Project management

Gantt charts

Main Gantt chart for amending MVP (Repeated)

<table>
<tr><th></th><th>Week 1</th><th>Week 2</th><th>Week 3</th><th>Week 4</th><th>Week 5</th><th>Week 6</th><th>Week 7</th><th>Week 8</th><th>Week 9</th><th>Week 10</th></tr>
<tr><td>Documentation and logging</td><td colspan="10">Continuous</td></tr>
<tr><td rowspan="3">Research</td><td></td><td colspan="3">Define research methods</td><td></td><td colspan="4">Research</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Interviews</td><td>Surveys</td><td colspan="2">Card sorting</td><td></td></tr>
<tr><td></td><td></td><td></td><td colspan="2">R1</td><td></td><td>R2</td><td>R3</td><td></td><td>Final MVP ammendment: R4</td></tr>
<tr><td>Design (UML, specifications, diagramming)</td><td></td><td></td><td colspan="3"></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Wireframing/protoyping</td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Low fidelity</td><td colspan="2">High fidelity</td><td></td></tr>
<tr><td>User testing/feedback</td><td></td><td></td><td></td><td></td><td></td><td colspan="5">Update wireframes, prototypes and MVP according to user feedback</td></tr>
<tr><td>Finalisation of MVP</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2"></td></tr>
</table>

Figure 72:Gantt chart for amending MVP

Gantt chart for requirements gathering

ID	Name	Start Date	End Date	Duration
1	Interview with Kelvin	Nov 06, 2023	Nov 08, 2023	3 days
2	Interview with Sahaf	Nov 06, 2023	Nov 08, 2023	3 days
3	Creation of questionnaires	Nov 13, 2023	Nov 16, 2023	4 days
4	Questionnaire write-up	Nov 17, 2023	Nov 17, 2023	1 day
5	Creation of card sorting activity	Nov 20, 2023	Nov 21, 2023	2 days
6	Card sorting data collection	Nov 22, 2023	Dec 03, 2023	12 days
7	Card sorting write-up	Dec 04, 2023	Dec 05, 2023	2 days
8	Creation of R2	Nov 13, 2023	Nov 19, 2023	7 days
9	Creation of R3	Nov 20, 2023	Nov 26, 2023	7 days
10	Creation of R4	Dec 04, 2023	Dec 08, 2023	5 days

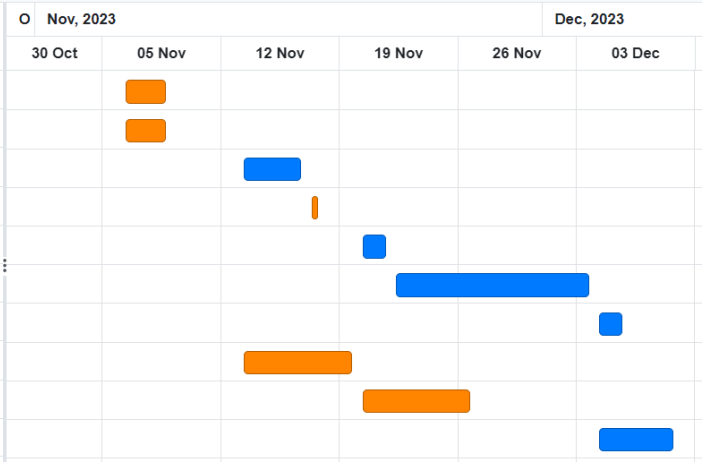


Figure 73:Gantt chart for requirements gathering

Gantt chart for prototyping

ID	Name	Start Date	End Date	Duration
1	paper wireframes	Dec 04, 2023	Dec 08, 2023	5 days
2	Feedback on paper wireframes	Dec 09, 2023	Dec 09, 2023	1 day
3	digital low fidelity wireframes	Dec 10, 2023	Dec 13, 2023	4 days
4	Feedback on digital lf wireframes	Dec 14, 2023	Dec 15, 2023	2 days
5	High fidelity wireframes	Dec 16, 2023	Dec 19, 2023	4 days
6	Feedback on high fidelity	Dec 20, 2023	Dec 21, 2023	2 days
7	Database prototyping	Dec 04, 2023	Dec 09, 2023	6 days
8	ER Diagram	Dec 09, 2023	Dec 09, 2023	1 day

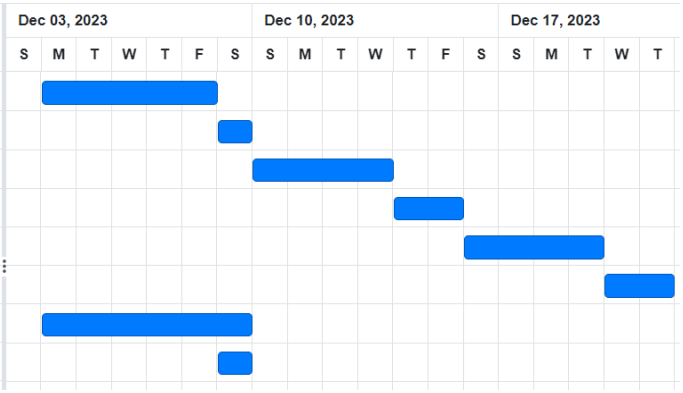


Figure 74:Gantt chart for prototyping

Development Methodology

The Agile methodology is often considered superior to Waterfall for developing a room booking app due to its adaptability to change, regular user involvement, and iterative development approach. Agile's flexibility allows for adjustments to requirements as they evolve, crucial in a dynamic domain like room booking. The iterative nature of Agile, with short development cycles, facilitates the early delivery of functional increments, enabling frequent stakeholder feedback. This contrasts with Waterfall's sequential approach, where each phase must be completed before moving on, potentially delaying user feedback. Agile's emphasis on risk management and early issue detection contributes to more effective problem resolution, reducing the likelihood of major setbacks. Additionally, Agile's quicker time-to-market and continuous customer collaboration contribute to higher customer satisfaction, ensuring that the final product aligns more closely with user expectations. Ultimately, the choice between Agile and Waterfall depends on the specific project requirements and characteristics. Overall - we chose a methodology that fit more of the aspects of an agile methodology as opposed to a waterfall methodology as we believed it would align more with our focus on usability and our practices surrounding user-centred design [19].

Organisation

Our group worked well when it came to organising resources and work. We maintained frequent contact through our Discord server throughout the week to provide one another with updates on progress and changes to the plan made each Tuesday. Furthermore, we centralised the storage of our notes, diagrams and documents through OneDrive and Google Docs. This meant that any information available to one member of the group was

also available to all members of the group - ensuring that we were all as informed as possible on not only the current schedule of work moving forward, but also on suggestions and information gathered from the weekly seminars.

Evaluation

In the project definition stage, we decided to include personal logs from each term member, analysing our strengths, weaknesses, and tendencies etc. Carrying out these logs in the initial stages was invaluable. We gained the opportunity to not only learn how cohesively we can function as a group but also track our progress as well. Having laid down that initial groundwork we discussed how we would track our progress for our second stage Project Proposal.

Upon analysis of our Product Definition with comparison to the needs of the Proposal stage we found that we had exceeded the requirements of the definition which made this stage a different sort of task altogether. While we recognised that much of our completed work would require some revisions, (also that we needed to delve deeper into market research, prototyping and stakeholder analysis). We also recognised that principal amounts of research and information would be able to carry forward into this document (we had overestimated the requirements of the first task and covered more ground).

With this in mind, we decided that instead of individual logs we would track our progress of our entire group in by-weekly sprints through reflective logs. Each sprint would headline goals and point towards tasks we need to complete. This was designed to be accompanied by task assignment (if necessary) and a short group reflective text at the end of the sprint [12].

Our initial sprint set the tone for our second phase which was designed to focus on assessing our proposal needs and expanding upon research efforts. We had the enjoyable task of reviewing our past achievements and recognise the progress we had made in the definition phase of development.

Yes, we successfully anticipated our workload in upcoming sprints and and yet, while overall this sprint was successful, we would argue that we did not go far enough towards evaluating the exponential increase in workload as we developed this document. This might have meant we had underestimated the task but at the same time we did feel that most of the tasks that were “added” in our pipeline came about as we were completing tasks rather than through ignorance or oversight.

Our next (second) sprint created a new level of intensity for us. We found that task allocation and adjustments were pivotal to our success here as we faced sickness within

the group. At this point there was a good level of momentum and cohesion regarding our vision for the project. This is reflective in how closely our wireframing, specifications and scope line up in terms of our proposal goals and concept [15].

The final sprint has been the most challenging so far. While our issues with external factors affecting our productivity are well understood, we feel that they have compounded to an uncomfortable level at this stage of our development. Specific team members have been handling the lion's share of work at this stage, and while this does not discredit (not would we want to) the hard work everybody has done up until now, this final sprint was simply too much of a task to be left to only a few members of the team to work on. Granted most of the team has been present for discussions, input and valuable consultation, it is simple availability that has been lacking this time around.

To the credit of those who were unable to participate to a greater level in our final sprint; it is a testament to how much of the foundation the entire team laid down weeks ago that made our final sprint's goals an approachable (albeit oversized) task. From wireframing, to market research and justification of the project development methodology, all of this has been consolidated and appropriated to a point where prototype databasing and high-fidelity prototypes have been constructed on the basis of our research across the development of the last 10-11 weeks.

Conclusion

In conclusion, we are confident in our vision for the development of our application. We feel we have successfully scoped and defined our project and pushed our definition further across the last 6 weeks to a point where we can envision and prove the concept of our proposed application using our MVP as a signpost for our development [14]. As a team, we have maintained a shared vision throughout and have seldom needed to refocus any of the information that individual team members have gathered and concatenated into our document. It is this strength and confidence we have in each other that has carried us this far and we believe that we can bring this attitude to the development stage of our application. In contrast, our biggest concern going forward is to fully address how we prioritise our development time. The next stage is crucial, and we cannot afford to repeat the mistakes we may have made now.

Reference list/Bibliography

[1] Abraha˜o, S., Gravino, C., Insfran, E., Scanniello, G. and Tortora, G. (2013). *Assessing the Effectiveness of Sequence Diagrams in the Comprehension of Functional*

Requirements: Results from a Family of Five Experiments | IEEE Journals & Magazine | IEEE Xplore. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6193111.

[2] Alonso, S., Kalinowski, M., Viana, M., Ferreira, B. and D.J Barbosa, S. (2021). *A Systematic Mapping Study on the Use of Software Engineering Practices to Develop MVPs | IEEE Conference Publication | IEEE Xplore*. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9582602.

[3] BasuMallick, C. (2022). *What Is a Single-Page Application? Architecture, Benefits, and Challenges* |. [online] Spiceworks. Available at: https://www.spiceworks.com/tech/devops/articles/what-is-single-page-application/.

[4] Bigelow, S.J. (2020). *7 techniques for better Agile requirements gathering*. [online] SearchSoftwareQuality. Available at: https://www.techtarget.com/searchsoftwarequality/tip/7-techniques-for-better-Agile-requirements-gathering.

[5] Bolloju, N. and Sun, S.X.Y. (2012). Benefits of supplementing use case narratives with activity diagrams—An exploratory study. *Journal of Systems and Software*, 85(9), pp.2182–2191. doi:https://doi.org/10.1016/j.jss.2012.04.076.

[6] Bourgeois, J. (2015). *Room Bookings*. [online] www.em.admin.cam.ac.uk. Available at: https://www.em.admin.cam.ac.uk/what-we-do/estate-operations/workplace-services/room-bookings.

[7] British Dyslexia Association (2018). *Dyslexia friendly style guide*. [online] British Dyslexia Association. Available at: https://www.bdadyslexia.org.uk/advice/employers/creating-a-dyslexia-friendly-workplace/dyslexia-friendly-style-guide.

[8] Connor Brooke (2019). *The Pros and Cons of 10 Information Gathering Techniques for Projects*. [online] Business 2 Community. Available at: https://www.business2community.com/strategy/pros-cons-10-information-gathering-techniques-projects-01437727.

[9] Davidson, T. (2022). *Native App vs. Web App: Are Native Apps Still Relevant? | Clean Commit*. [online] cleancommit.io. Available at: https://cleancommit.io/blog/native-app-vs-web-app-are-native-apps-still-relevant/#:~:text=Web%20apps%20are%20typically%20developed.

[10] expressjs.com. (n.d.). *Using Express middleware*. [online] Available at: https://expressjs.com/en/guide/using-middleware.html.

[11] Fox, J. (2021). *Accessible content design for emojis • UX Content Collective*. [online] UX Content Collective. Available at: https://uxcontent.com/accessible-content-design-for-emojis/#:~:text=Use%20emojis%20that%20translate%20well.

[12] Kaur, K., Jajoo, A. and Manisha (2015). Applying Agile Methodologies in Industry Projects: Benefits and Challenges. *2015 International Conference on Computing Communication Control and Automation*. doi:https://doi.org/10.1109/iccubea.2015.166.

[13] Kosari, A. (2022). *10 Essential Guidelines for Colorblind Friendly Design - UI/UX/Website/Data*. [online] COLORBLIND GUIDE. Available at: https://www.colorblindguide.com/post/colorblind-friendly-design-3.

[14] Lenarduzzi, V. and Taibi, D. (2016). MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product. *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. doi:https://doi.org/10.1109/seaa.2016.56.

[15] Livermore, J.A. (2007). *Factors that impact implementing an agile software development methodology*. [online] IEEE Xplore. doi:https://doi.org/10.1109/SECON.2007.342860.

[16] Park, Sangwon & Yin, Yizhen & Son, Byung-Gak. (2018). Understanding of online hotel booking process: A multiple method approach. Journal of Vacation Marketing. 25. 135676671877887. 10.1177/1356766718778879.

[17] PERERA, K. (2019). *LECTURE HALL SCHEDULING AND TIMETABLE MANAGEMENT SYSTEM* . [online] Available at: https://dl.ucsc.cmb.ac.lk/jspui/bitstream/123456789/4407/1/2016MIT056.pdf.

[18] Sherwin, K. (2018). *Card Sorting: Uncover Users' Mental Models for Better Information Architecture*. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/card-sorting-definition/.

[19] Thesing, T., Feldmann, C. and Burchardt, M. (2021). Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project. *Procedia Computer Science*, [online] 181(1), pp.746–756. doi:https://doi.org/10.1016/j.procs.2021.01.227.

[20] Um S and Crompton JL (1992) The roles of perceived inhibitors and facilitators in pleasure travel destination decisions. Journal of Travel Research 30(3): 18-25.

[21] Vermeulen IE and Seegers D (2009) Tried and tested: the impact of online hotel reviews on consumer consideration. Tourism Management 30(1): 123-127

[22] w3resource. (2019). *MySQL security - w3resource*. [online] Available at: https://www.w3resource.com/mysql/mysql-security.php.

[23] Warade, A. (2020). *Three Tier Architecture vs MVC Architecture*. [online] www.c-sharpcorner.com. Available at: https://www.c-sharpcorner.com/blogs/three-tier-architecture-vs-mvc-architecture2.

[24] www.eventmapsolutions.com. (n.d.). *Developing Booker with the University of Cambridge*. [online] Available at: https://www.eventmapsolutions.com/success-stories/creating-booker-with-the-university-of-cambridge.

[25] www.roombookingsystem.co.uk. (n.d.). *Room Booking - Features for Schools*. [online] Available at: https://www.roombookingsystem.co.uk/features-schools.asp.

[26] www.schoolbooking.com. (n.d.). *Feature Timeline - SchoolBooking*. [online] Available at: https://www.schoolbooking.com/timeline.

[27] www.w3.org. (n.d.). *Understanding Success Criterion 1.4.6: Contrast (Enhanced)*. [online] Available at: https://www.w3.org/WAI/WCAG21/Understanding/contrast-enhanced.html.

Appendix

Bi-weekly sprints:

Weeks 1 – 2: CURRENT ASSESSMENT OF NEEDS, STAKEHOLDERS AND MARKET RESEARCH

- Reflect on the previous task - scan through document for things we did well discuss [ALL]
- Identify method of progress tracking [Noah]
- Look at UMLs and current specifications - improve? [Jake, Kelvin]
- Adjustments to scope [Ben]
- Can we expand on market research? [Sahaf, Spike]

- Investigate prototyping [Shaquille]

Reflection:

The past two weeks went well. We realised how much progress we made in the Project Definition after analysing the document. Because of this we feel that this initial sprint of the second phase has been lighter than we expected. We have successfully managed to agree upon progress tracking and begun expanding upon our work with market research and tightening up the scope.

Outside of this task we have all had more assignments to tackle which has made these two weeks something of a respite for us to get our bearings under the workload. Leading up to the Holidays this does mean more intense development for the next sprint/s but morale is currently holding strong.

Weeks 3 -4: MORE ON STAKEHOLDER AND RESEARCH, DIAGRAM DEVELOPMENT, SPECIFICATION

- Sprint direction (allocation of tasks) [Ben]
- Scope Adjustments [Ben]
- Interview structures [Shaquille, Noah]
- Construct surveys [Shaquille]
- UML reworking [Jake]
- Tighten up the market research
- Database structuring [Jake]
- Prototyping wireframes R1 [Shaquille]
- Cross document referencing [Ben]
- Contacting stakeholders [Kelvin, Sahaf, Noah, Shaquille]
- More Market research (scale down, appropriate) [Spike, Sahaf, Noah]
- Surveys and interviews [Noah, Shaquille]

Reflection:

As expected this sprint was much more intense for us than the first. The biggest challenge we faced was identifying points to adjust in our current document with cross referencing the first. We spent much time analysing the previous document which started to bring about levels of uncertainty amongst the team Having learned from our last assignment we discussed as a team and we ended up successfully allocating tasks.

The first half of this sprint a few illnesses occurred. This combined with some impending assignment deadlines meant we had to reorganise our allocation of tasks beyond our allocation. Shaquille was invaluable during this sprint. He made the principal forms necessary for surveys/interviews which helped us to validate, scope and justify the goals of

our project. He also spent time while sick constructing designs for the wireframes during times where he was unable to come in. Again this was invaluable in the first week of this sprint.

The biggest drawback for us this sprint was how exponential our tasks seemed to become under our increased workload with the added pressuring of assignment deadlines. We discovered more tasks as we began working through this assignment. This meant a lot of work needed to happen while other tasks and life commitments were keeping us from focussing on this document as much. This thankfully hasn't created any contentious attitudes between group members but the added stress has definitely affected our productivity.

Overall we feel confident in our ability to complete this assignment despite the roadblocks we have had to workaround and we look towards our next sprint to tighten up our work before the final submission.

Weeks 5/6: Citation, revisions, editing and high fidelity diagrams

- Second set of wireframes R2 [Shaquille, Jake]
- R2 wireframe revisions [Ben]
- Development Methodology justification [Kelvin]
- R3 wireframes with changes from feedback from R1 and R2 [Jake]
- Discuss R4 - high fidelity UI production[Shaquille, Ben, Jake]
- Accessibility (colours, fonts, other considerations) [Ben]
- R4: High fidelity diagrams & prototype [Jake]
- Citations [Noah]
- Finish and review creative logs [Noah]

In review:

Similar to the last sprint, our tasks became exponentially more complicated and time sensitive, especially with the added deadlines of other modules. Unfortunately (along with the myriad of deadlines) this has meant there has definitely been a lack of overall group participation as compared to the previous sprints. Many of the tasks have regrettably fallen to “the usual suspects” and this is an unfair practice and precedent to set so early in our production. We need to work harder to ensure every single member is present and responding to interactions about the progress of our development and engaging with the responsibilities of producing work. .

Interviews:

Interview structure:

Q. How will we conduct interviews?

A. In person, with an interviewee being given free reign to speak in response to talking points

Q. Who will we interview?

A. Society Leads and Lecturers.

Q. How to get access to these people?

A. Two societies lead in our group. Multiple social links provide us access to society leads at Goldsmiths and other Universities. Email through lecturers.

Q. How will this affect our scope?

A. Allows us to adjust the requirements by priority of things they wish to see or struggle with in room booking and management. This will also inform things such as low fidelity to high fidelity.

Q. How will we follow-up?

A. These interviews will enable us to create a survey in response to the interviews/ low fid feedback which we can then feed back into our plan and high fidelity versions..

1. Our proposed interview structure:

The Current System

Q. As a lecturer/society member how do you feel about your involvement in the room booking process and timetabling process and what things,

Q. Are there any problems you face with the system? Scheduling issues, overbooking, allocation?

Q. Is there any functionality that you feel the current system is missing?

Q. Does the current format for accessing and utilising the system satisfy your needs?

Q. Is there a format you would appreciate more (e.g., transferring from Google forms to a dedicated web application, etc.)

Q. Do you feel there are any unnecessary aspects of the system that should be removed? What are they?

We have decided to use open-ended questions like these as they pose less of a risk of limiting our stakeholder's opinions when it comes to gathering requirements. Because of this - we can garner a better idea of what our final solution will entail and this should allow us to more effectively amend our MVP. Furthermore - it will give us a better platform for moving towards more focused requirements gathering: such as surveys, questionnaires and card sorting activities. Closed questions have also been included for direct answers so during our coding development stages we know what our definite focuses are. The importance of including a mixture allows us to understand the user's main needs and wants. Once this is completed we can then focus on the more complex features we wish to implement whilst keeping our end-user's satisfied.

2. Survey follow up:

To follow up on the results of the interviews we will construct a survey that can ideally be taken by lecturers and society leads but could be taken by anyone who may have had experience with booking a room, experience, or something through an online application.This survey will be accompanied by an overview of the proposed system vs current system.

Current System:

- Google docs form and risk assessment form
- Drop down text menu for event type
- Drop down text menu for room type
- Form for naming the event
- Form for the time period.
- Contact form
- Email correspondence - lengthy wait time

Short overview of our new system , its features and goals

The proposed system:

Essential to our system will be -

- Catalogue of rooms by type, size and features - visually represented
- View Available/pre booked/ commonly booked rooms
- Booking cancellations
- Confirmation

Additionally:

- Check- in system
- Facilities search (keyword search)
- Templates for risk assessment

Q. How important is a visual aid, when booking a facility?

Q. How important is it to have the availability of rooms dynamically updated?

Q. How important is a facilities search to a booking system?

INTERVIEW TRANSCRIPTS:

Khar Chew

INTERVIEW BEGIN

Speaker 1: So could you please tell me a little bit about yourself, your name, what university you go to, any societies that you are part of, and what position you hold?

Speaker 2: My name is Khar Chew. I am a second-year student at Goldsmiths University of London. I am currently the president for the Anime Games Society.

Speaker 1: Okay. So, we are developing a room booking system for societies and for lecturers. We're going to ask you a couple of questions about how you feel about the current system to help us inform our development process. As a society member, how do you feel about your involvement in the room booking process and the timetabling process for your events?

Speaker 2: That position, that role for room booking is mainly in charge of the secretary. However, I am partially responsible when picking the rooms suitable for events.

Speaker 1: Are there any problems that you face with the current systems, such as scheduling issues or overlooking the allocation of space?

Speaker 2: Yeah, it's a lot of issues, like having to book the room two weeks in advance and it takes time for us to get a response. By the time we get a response, we don't get the room we want, and they end up giving us a different room, which is completely unsuitable for our event. And when we submit a new request, replying to that saying we want the room to change to a new, more suitable one, they don't come back to us in time or they say, "You should have booked it two weeks in advance." So it's uncooperative.

Speaker 1: Okay. So given those problems that you face with the system, do you feel that the current format for accessing and utilising the system satisfies your needs?

Speaker 2: It does, the main function is to book a room, which does its thing. Timing for schedules is just hard because we don't know which room is available or not.

Speaker 1: So what functionality do you feel the current system is missing?

Speaker 2: A calendar where it shows which room would be booked and when it's available, so everyone can see which room has been booked, and you can plan ahead to make sure that that room is guaranteed to be yours at least.

Speaker 1: Do you feel that there are any unnecessary aspects of the system that should be removed, and what are they?

Speaker 2: For me, the health and safety forms become repetitive, which isn't really necessary since most societies fill out new forms for the same event types. Which is really not needed.

Speaker 1: So the current format for the booking processes is through Google Forms, is there a format that you would appreciate using more than Google forms?

Speaker 2: Probably a redesign - using a separate app/website which has a UI. Just easier to understand and more accessible and simpler and straightforward to look at and understand. Because on the form, it's just a straight paragraph of just putting details in, which gets a bit annoying.

Speaker 1: Okay. Thank you. That's all the questions we have for you today.

INTERVIEW END

INTERVIEW BEGIN

Syed Sahaf

Speaker 1: So could you please tell me a little bit about yourself, your name, what university you go to, any societies that you are part of, and what position you hold?

Speaker 2: My name is Sahaf. I am a second-year student at Goldsmiths University of London. I am currently the secretary for the Anime Games Society.

Speaker 1: Okay. So, we are developing a room booking app for the university. We're going to ask you a couple of questions about how you feel about the current system to help us inform our development process. As a society member, how do you feel about your involvement in the room booking process and the society's scheduling?

Speaker 2: As the secretary, I'm the one who handles the room booking process. In our society, it's very cumbersome as we don't know exactly when rooms are free. We're not given a clear time to able of when rooms are free. So we just have to guess which rooms are free, which times it's free and, and it takes a while to get feedback on these room booking requests to see when they are free.

Speaker 1: Are there any problems that you face with the current systems, such as scheduling issues or overlooking the allocation of space?

Speaker 2: Even when we get feedback, a lot of the time they give us rooms that aren't suitable if we didn't get the room we wanted. This feeds into scheduling issues, not just booking and allocation.

Speaker 1: Given those challenges, do you feel that the current format for accessing and utilising the system satisfies your needs?

Speaker 2: It's too overcomplicated and cumbersome. We need a better system in place.

Speaker 1: So what functionality do you feel the current system is missing?

Speaker 2: A calendar system, seeing which rooms are free at certain times and days. Maybe a map to see where rooms are would be useful.

Speaker 1: Are there any unnecessary aspects of the system that should be removed, and what are they?

Speaker 2: Maybe the fact that we have to complete multiple different forms, and the email system for everything is really cumbersome. Combining forms and integrating risk assessments with the booking system would be useful.

Speaker 1: Compared to the current Google forms, is there a particular format that you would appreciate more?

Speaker 2: Maybe a webpage or something like that? It would be more efficient as part of the room booking system.

Speaker 1: Okay. Thank you for your time. That's all the questions we have for now.

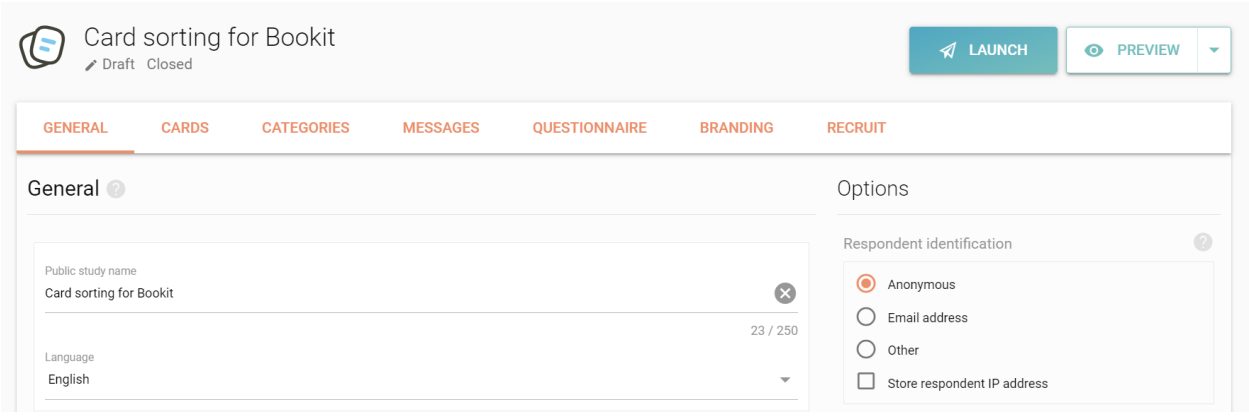
INTERVIEW END

Card sorting resources

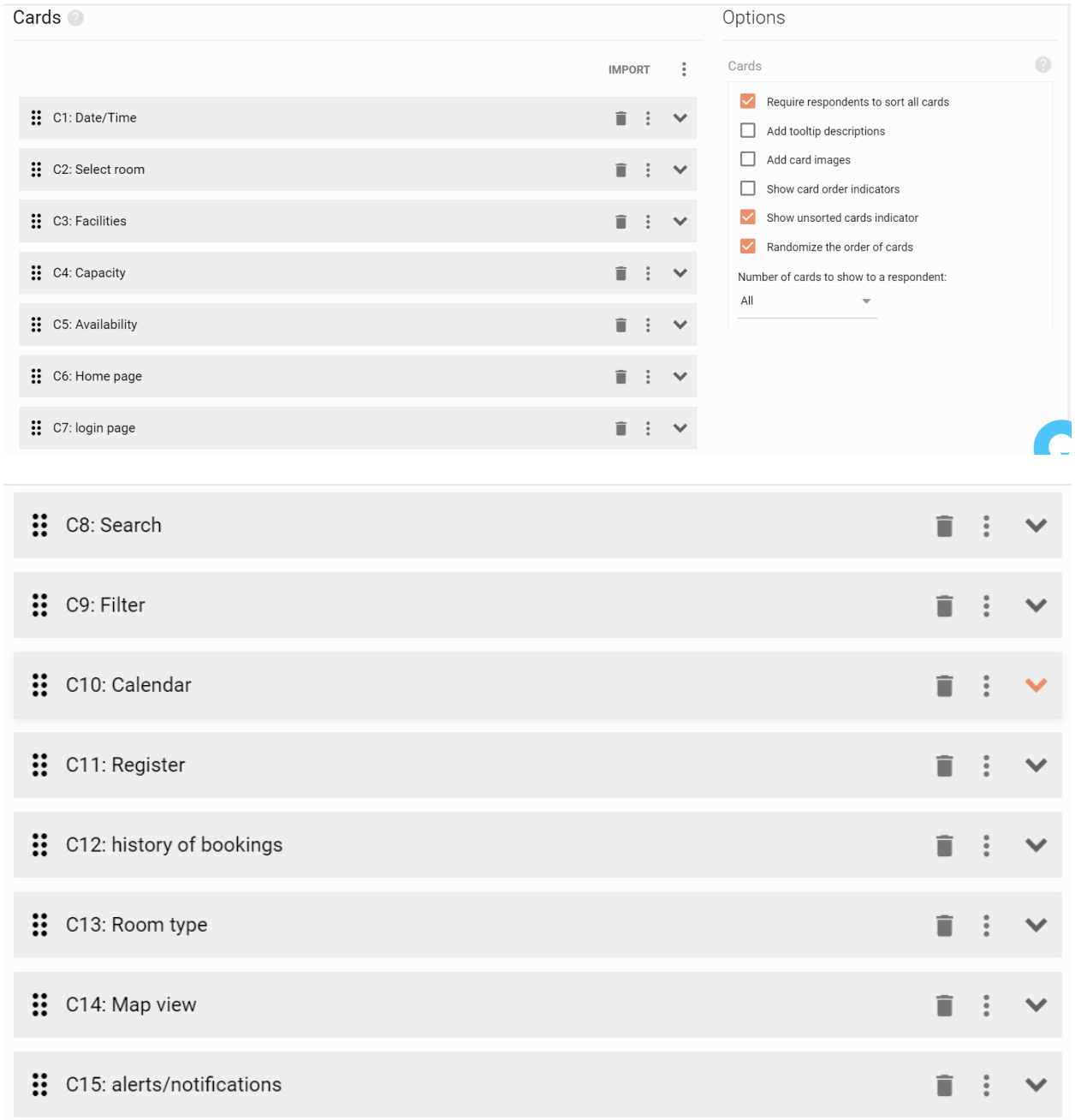
Raw data:

Cards	Booking process	Navigation	User account	Search	Integration	Notifications	Performance	Accessibility	Usability
Date/Time	4	0	1	1	1	2	0	1	0
Select room	6	1	0	1	0	0	0	1	1
Facilities	4	3	0	1	2	0	0	0	0
Capacity	3	0	0	1	1	0	3	0	2
Availability	5	0	0	0	0	1	2	2	0
Home page	0	3	3	0	1	0	0	0	3
login page	0	1	6	0	1	0	0	2	0
Search	1	1	0	8	0	0	0	0	0
Filter	2	0	0	5	1	0	1	0	1
Calendar	4	0	0	0	2	2	0	1	1
Register	1	1	7	0	1	0	0	0	0
history of bookings	6	0	2	0	0	0	0	0	2
Room type	7	0	0	2	1	0	0	0	0
Map view	0	4	0	0	0	0	1	1	4
alerts/notifications	0	0	1	0	2	7	0	0	0
Loading times	0	0	0	0	0	0	8	0	2
response time	0	0	0	0	1	2	4	0	3
scalability	0	0	0	0	3	0	3	1	3
user profiles	0	0	8	0	1	0	0	1	0
Font/colours	0	0	2	0	1	0	0	6	1

UXtweak overview



Cards


Cards
IMPORT
C1: Date/Time
C2: Select room
C3: Facilities
C4: Capacity
C5: Availability
C6: Home page
C7: login page
Options
Cards
Require respondents to sort all cards
Add tooltip descriptions
Add card images
Show card order indicators
Show unsorted cards indicator
Randomize the order of cards
Number of cards to show to a respondent:
All
C8: Search
C9: Filter
C10: Calendar
C11: Register
C12: history of bookings
C13: Room type
C14: Map view
C15: alerts/notifications

C16: Loading times

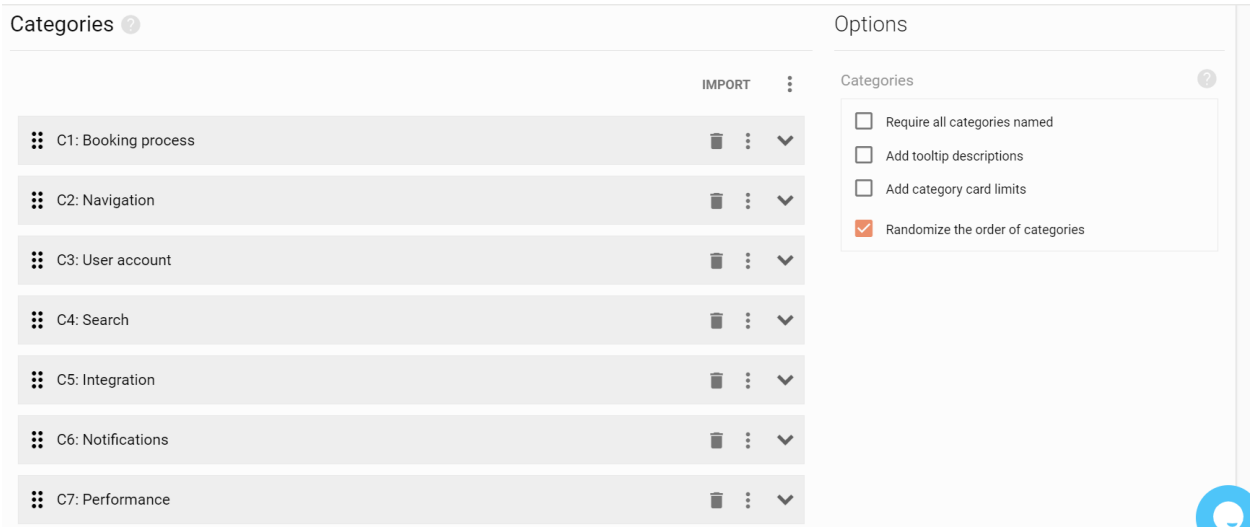
C17: response time

C18: scalability

C19: user profiles

C20: Font/colours

Categories



C8: Accessibility

C9: Usability

+ ADD ANOTHER CATEGORY