EMA – PROJECT REPORT

Matthew Mason: C6122243

BSC (HONOURS) COMPUTING AND IT: TM-470
AUGUST 21, 2023

TABLE OF CONTENTS

DEVELOPING A WEB APPLICATION BOOKING SYSTEM FOR A THREE-VENUE BOARDGAME CAFÉ BUSINESS

1. INTRODUCTION

The current reservation process at each venue is causing a number of problems; miscommunication of details, double bookings, recording incorrect information, loss of customers details, incorrect booking cancellations, etc.

Customers contact the venue where a staff member records the booking details and checks the request against their availability – a paper sheet attached to the staff notice board. Customers can also email the business and an administrator will liaise with individual venues to confirm and adjust bookings.

These problems have led to inconsistent and unreliable internal management and services, negative public perception and reputation, lost business, revenue and trust. The business has decided to rectify this by commissioning a booking system to counteract these dilemmas.

To solve this problem, the proposed solution is the development of a responsive web application for use on desktop and tables that allow staff to manage customer bookings. This provides a shared centralised management solution, accessible from anywhere and across multiple devices by utilising web-based technologies such as HTML, CSS, JavaScript and React. Multiple internet connection technologies including ethernet, Wi-Fi and cellular data provides redundancy to overcome the necessity of requiring an always-online connection to access the system and individually assigned credentials provide staff secure access to manage customer bookings, rearrange tables and dates, add comments to bookings and fulfil specific customer requirements.

A number of other solutions were considered but ultimately rejected:

- Installing dedicated software on PC's and tables, however, this would necessitate developing for multiple operating systems, requiring a wider specialist knowledge of multiple different subsets of languages and would significantly increase development, maintenance and infrastructure costs.
- Developing a hybrid application produced from a single code base that can be deployed across multiple devices, however this would notably place limits on functionality: access to hardware and sensors is only available via existing API plugins or additional development of new APIs, security concerns require additional development strategies and smaller mobile screen sizes affect usability, user experience, utility and efficiency.

For this project, I will be utilising my personal experience of managing booking as a restaurant host, the shortcomings of using a paper-based reservation system, plus my knowledge and experience from pervious modules to develop my skills in software engineering, database and API design and further explore and incorporate appropriate interaction design techniques.

A user-centred focus will help understand the domain while incorporating regular feedback will help produce artefacts such as prototypes, use cases, and infrastructure models to develop a solution and interface that meets the project's success criteria. It will allow me to produce a demonstratable booking system comprising a range of functionality including adding, modifying and cancelling reservations, setting up new venues, employees and exporting records. This will improve the customer experience by alleviating the venues problems and impact the business by providing a number of benefits:

- Manage multiple venues via a single web application interface.
- Prevention of double bookings.
- Reducing lost bookings and cancelled bookings.
- Reducing recording of incorrect information.
- Allowing bookings to be made outside normal operating hours to increase business.
- Multi-user management and permission levels for greater accountability.
- Historical data of reservations if problems arise.

This work is organised as follows: section 2 evaluates the projects impact on the business in terms of procedures, legal, social, ethical and professional issues, stakeholders, benefits, GDPR and its impact in the wider context of its adoption and effect on similar organisations in similar business sectors. Section 3 discusses related literature and its influence on the project. Section 4 discusses the project work conducted; development processes undertaken, data gathering techniques, interviews and analysis, compiled artefacts, iterative processes, design decisions and the projects outcome. Section 5 discusses further work. Section 6 discusses a review of the project management. Section 7 provides a critical evaluation of personal development.

2. Project Impact

While far from being a fully comprehensive working and maintainable web application, there is potential for continued development with enough resources to become a competitive and established product and this project provides a solid foundation of development techniques when it comes to building a unique reservation system. Even though the project was aimed at developing a bespoke system, it has potential to appeal with a wider audience by providing modularity, catering unique configurations for each customer and only paying for what elements they need making it much more affordable too. Continued user feedback has provided the greatest benefit to the development along with utilising unique gamification techniques to elicit requirements, engage participants and has played a pivotal role in understanding what users want from a booking system.

Continuous consideration of those who would be affected by the development of this project has included:

1. Customers: Legal issues would revolve around the storage and secure transmission of customer details, GDPR, etc.
2. Administrative personnel: They will need to be trained on GDPR so they can comply with legislation. Harmful effects as a consequence of the system is the ability to check logs for responsibility of mistakes.
3. Staff of the venue – The success of the software directly affects front of house staff by lessening the frequency of dealing with customer complaints from the mismanagement of reservations.
4. Managers: The responsibility of who was at fault is easier to narrow down in the event of problems as logs can be checked (time/date booking was made, any modifications made to the booking and by who at what time/date, etc) and situations can be managed accordingly and more appropriately.
5. Business owner: legal issues could be applicable such as the negotiation of contacted outputs and deliverables, cost of potential lawyers, legal ramifications of failing to deliver, etc.
6. Web developers: Unless contracted under the same company, web and software developers would need to collaborate together to modify the website and integrate the web application into the structure of the current website. This benefits the web developers by providing them with work and revenue but could provide harm if agreements and decisions cause disparity between designers and fails to provide integration and software that meets the users and stakeholders needs.
7. Software designers/developers: Similar to above, provides work, revenue, and opportunity to gain and maintain a potential new client for further development in the future. Harm could arise from exorbitant client expectations causing a conflict and damaging reputations.
8. IT and telecoms contractors/staff: Would be responsible for working with the software developers and website developers to agree on the best integration approach based on current infrastructure (on-site or cloud services, new equipment, responsibilities, who has access to what services and data? etc).
9. Competitors: other booking systems wouldn't see much if any impact from a bespoke system being developed unless it expanded into a commercial product later on.

Legal issues include responsibility of provided services, contractual agreements, service levels, staff policy and treatment, customer, client and staff data accessibility, GDPR, user involvement and ethical treatment, consent, age rating, licensing, etc

3. Related Literature

TerraVet: A mobile and web application framework for pet owners and veterinary clinic (Llaneta, et al., 2022)

The research-article covers the development framework of a web and mobile application for veterinarian clinics and pet owners to locate clinics using GPS, arrange appointments, provide online consultations and create e-cards to monitor and record pet health; this information is stored via MySQL databases, accessed via web APIs. This framework, which covers the following, will be utilised during my projects early development to help when generating:

- An Initial system architecture.
- Ideas for data gathering methodologies.
- Lifecycle model considerations.
- Software engineering practices such as use cases that lead to functional and quality requirement elicitation.

The section covering system features along with illustrations of the interface will provide a starting point for initial sketches and designs, the consideration of including a dashboard landing page upon logging in and access level and privileges of various user types (my project involves hosts, a variety of managers, booking operators, etc). While the web application differs significantly from my proposed project, the methods, ideas and system features provide a number of useful ideas for inclusion and influencing the direction of my project.

Applying Gamification to Prioritize Requirements in Agile Projects (Silva, et al., 2023)

The paper proposes a gamification method; the PRIUS (PRIoritizing User Stories) system, for engaging stakeholders' participation in prioritizing requirements. It combines game elements and a prioritization model (Weigers, 1999) that follows an 8-step sequence, estimating the benefit, penalty, cost and risk of a requirement to calculate its priority and produce a prioritization matrix.

The participants for my project are ideal for this method and requirements prioritization and design decisions would benefit greatly from implementing gamification techniques such as task completion leaderboards to determine efficiency and utility of potential designs.

The idea is to establish objectives for a given methodology (e.g., "OB03 More comments about the stories") to serve as a guide for adopting correct gamification elements such as leaderboard, roles, experience points, etc, defining expected behaviour of participants ("EB4 Make more comments on each proposed user story"), activities and the rewards for each one completed ("Be the first to comment on the story +5 points Behaviours: EB4"). The paper utilize user stories and gamifies engagement by displaying user rankings, who provides more comments, the users accumulated points and the current prioritization rank of user stories for

all to see. The execution of their tested hypothesis resulted in higher engagement from users and the quality of comments was positively influenced by the method used.

Relational Database Design and Implementation (Harrington, 2016)

This textbook follows a full tutorial from understanding the environment in which databases are used and required for operation, why they are needed and relationships between data, to models, design theory, and implementation. There are a number of case studies providing examples, an introduction to SQL, database security, and beyond.

There are a number of sections that will prove applicable to my project and influence its development:

- how database requirements are born from a systems analysis and development methodologies (prototyping, spiral, object-oriented analysis and design).
- Effects of poor database design, data modelling independent of specific theoretical data models, entity-relationships and ER diagrams, characteristics of columns and rows, primary keys, data dictionary tables, normalization, performance and partitioning.
- SQL, computer-aided software engineering tools, and case study examples.
- Concurrency control, security, and data quality.

A later element of my project will be the understanding, design and development of a database to store customer details, booking information, staff administration details and access levels. Each of the sections in this material alongside accompanying Codecademy courses will extend my knowledge beyond my level 3 learning providing a comprehensive, practical and theoretical framework to base my own database on.

Article - Different Types of Patterns for Online-Booking Systems (Teuber & Forbrig, 2004)

The paper aims to show that by analysing project tasks, users and objects, it's possible to generalise elements that can be applied to other systems to help provide common solutions to reoccurring interface design problems. An online booking system is used as an example. On a first scan some elements seem questionable (such as how does Paul know early registration causes people quite using the system early? There is no reference), despite this, it might benefit as a staring off point in terms of the types of tasks that might occur for my own system, at least, from a customer's perspective. They abstract user groups from the functionality available - based on the generalised tasks and extrapolate two types of user profiles: First time customers and registered customers. From there they use object-oriented concepts to determine objects, their attributes and relationships between them, finally using these analyses to produce a conceptual design.

Christopher Alexander first proposed the idea of patters to abstract a recognisable quality and apply it other designs. However, pattern languages can be more powerful, albeit less common,

as they incorporate a network of patterns that references each other to create a complete structure. (Preece, et al., 2015)A good example of this is Material Design: an adaptable system of guidelines, components and tools to support best practices of user interface design.

4. Project Work

To begin developing an initial solution, it was imperative to understand the design context and domain the web application would be deployed to, what competing products are available, and how these could be used alongside early and consistent user involvement, to improve and influence my design.

Researching the Design Context

Investigation of existing software using G2 (G2, 2023) yielded sales engagement platform and CRM systems such as Groove, Calendly and YouCanBook.me. They offered appointment scheduling and additional features beyond this projects scope; marketing/reporting tools, virtual zoom events, building branded websites, etc, despite this, the initial search provided preliminary usability and user experience goals (easy to use and learn, intuitive, user-friendly, efficient and helpful, etc) to be developed further post interview analysis and system features (create and accept bookings, customisable pre/post confirmation emails, user specific views, etc). Narrowing my research towards investigating existing establishments and their systems revealed a range of relevant software and user groups more appropriate to this project. These included (Names and locations of businesses have been omitted to protect staff identities in accordance with GDPR and the Data Protect act 1998):

- A 2-venue business with a 3rd in development
  System: Collins booking, and reservation system developed by TheAccessGroup. The system is used by designmynight.com, a hospitality discovery platform and provides three packages with increasing features, all including online reservation management, analytics and reporting.
- A 3-venue business
  System: ResDiary – provides various packages all with the same features of commission free bookings, reporting and CRM, the only difference is allowance of bookings per month which increases with more expensive packages.
- A single venue
  System: Bespoke system built into website.

By identifying, researching and utilising a variety of similar boardgame café establishments and the systems they used to manage customer bookings, I was able to identify typical system users, extrapolating their physical, sensory and cognitive abilities based on key activities of the system and the deployment environment, and create initial design sketches.

Key activities gathered, based on initial research include:

1. Creating a booking: Staff confirm and commit necessary booking details to the systems database.
2. Retrieve a booking: Staff retrieve matching bookings from searching reference numbers or details such as name and email address.
3. Modify a booking: Data for bookings can be altered and saved, overwriting the existing database entry.

4. Cancel a booking: Bookings can be removed from the system and database.
5. Navigation: Time and date display can be changed to view past and future bookings up to a specified limit.
6. Check-in: On arrival, customers bookings are located and updated to reflect attendance confirmation.
7. Update a booking: Change the status of a booking based on amount of time remaining.
8. Reply to enquiry: If a customer requests a booking at times where there is no availability, the details are sent to the venue as an email enquiry. Staff can reply with alternative options, add attachments, select preconfigured replies or write custom messages.

The system environment at each location follows an identical layout: A solid-oak adjustable height desk at the entrance is enclosed on three sides with a raised countertop. The booking system will be run from an ethernet connected laptop with Wi-Fi connection as backup and additional tablets running the web app will manage each section with options to filter the view to only those sections. There is adequate space for mobility impaired staff and adjustable seating provides an accessible comfortable operational environment.

Initial Design Sketches

Based on the design context, research of existing systems, key activities and users, I began sketching early potential and alternative designs for feedback during user interviews.

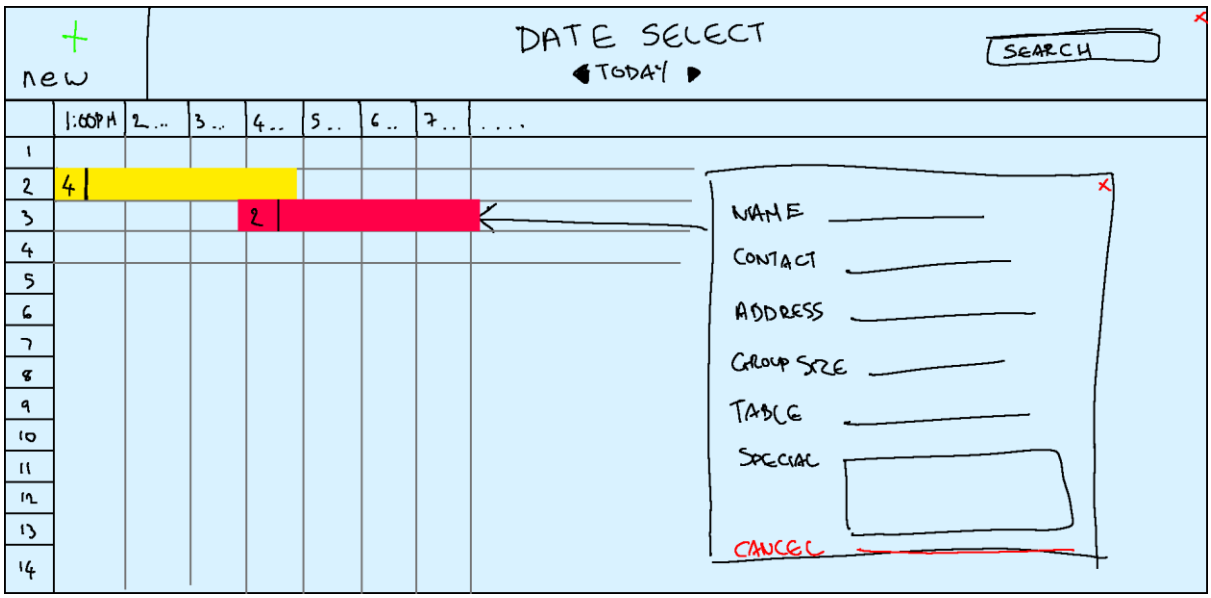

Figure 1: Horizontal Grid Design

Clicking the 'new' button displays a pop-out form allowing staff to enter bookings into the system, to the right, a date select/navigation option and the top right provides a search bar to input reference numbers or other details. The 'grid view', the most common way to organise data across booking software according to previous research, displays time horizontally across

the top and tables vertically on the left. Double clicking a booking in the grid brings up details for viewing and editing via a pop-out box.

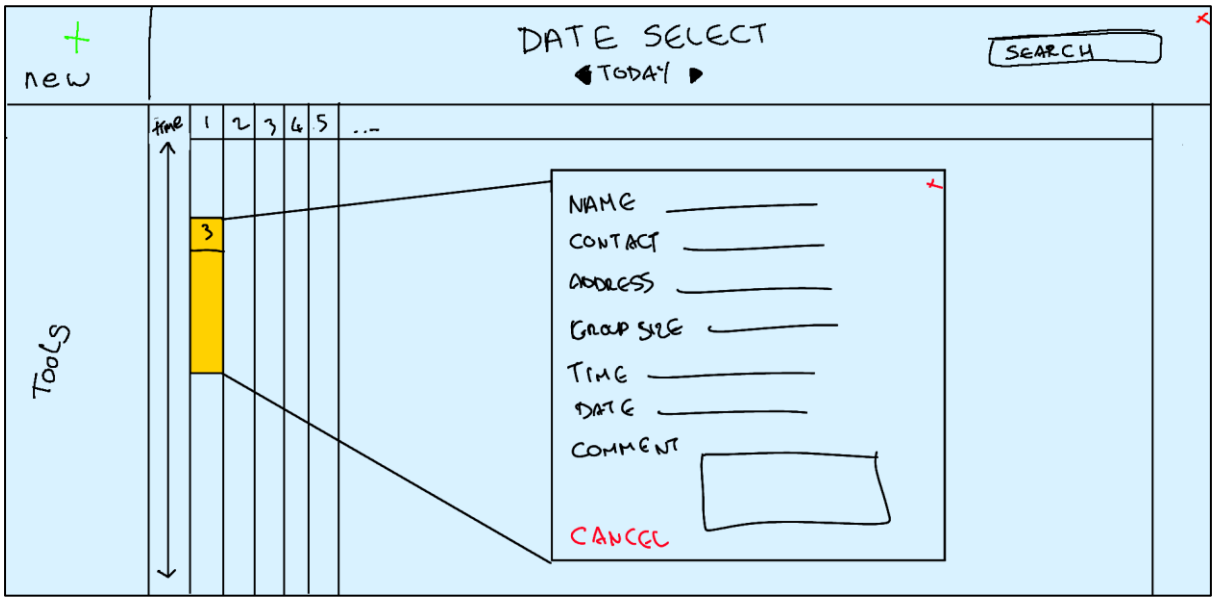

Figure 2: Vertical Grid Design

For the first alternative design, the top panel remains the same, however a minimizable tools panel on the left offers additional features such as different views. The grid is vertical with tables along the top and time horizontal on the left. The yellow block represents a booking component with details and text written vertically along its length.

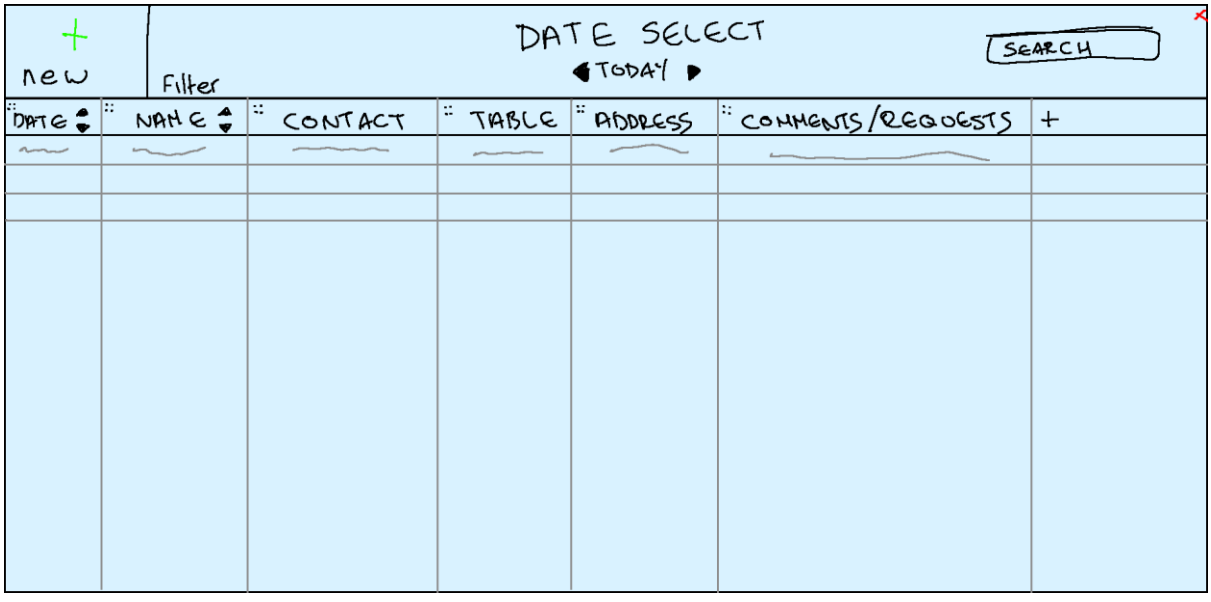

Figure 3: Spreadsheet Design

The 2nd alternative design favours a spreadsheet style similar to that of Microsoft excel. Columns can be rearranged via drag-and-drop into different positions and filters can be used for each column to organise its data by various options. Custom columns can also be added, and their size can be altered to fit the view.

User Interviews – Understanding the Domain

Selecting appropriate interviewees was determined by narrowing down the criteria to available participants, multi-site venues, and those where pre-booking is mandatory. This provided users who delivered better user representation, demonstrated various skill, abilities and experience levels and who closely matched the intended user base of the projects booking system.

Participants were required to read documentation detailing the study's purpose, length, participant obligations, how their data will be used and their rights for withdrawal. They were also required to sign a consent form, confirming their understanding and agree to or decline specific data gathering methods, e.g., audio recording. Photos of the system in use were not permissible due to GDPR and data protection.

Semi-structured interviews with management and staff bestowed qualitative and quantitative data, along with venue, participant and environmental observation details that were recorded when demonstrating specific procedures. These were used to iterate over and refine initially considered development expectations. Interviews were designed to discover highest to lowest employed operations, their complexity and difficulty, availability of multiple methods of completing the same tasks, positive and negative feelings about their existing system, user experience and availability of accessibility options.

Interviewees were also asked their opinions on initial design sketches with all venturing towards the horizontal grid design as their preferred favourite, prime reasoning being specific bookings were easier to locate compared to other designs. However, the ability to filter data in the spreadsheet design was a feature that many thought useful and could be implemented going forward. The vertical grid design wasn't conducive to determining booking details quickly due to the difficulty of reading sideways.

Success Criteria

The following usability and user experience goals were derived and developed from continuous analysis of: various lifecycle phases activities, interview data, and evaluating how the interface and the systems functions should support key activities and prioritised user requirements.

Effectiveness: The current systems process is slow, and staff continually mismanages bookings through loss and neglect. The new system needs to allow users to carry out work quickly, access information they need, and manage availability.
Q: Is the system capable of recording multiple bookings with a range of dates and times, tables, customer details and special requests that can be reserved and recorded into a database?

Test 1: How many participants can successfully and accurately enter a range of manually generated booking details in the system and retrieved them.

Test 2: How many participants can accurately identify and place a number of bookings and reminders?

Efficiency: Currently, due to bookings being stored physically, the booking process is tedious, inefficient, requires significant time to gather details and check for conflicts, and is prone human error. The system needs to provide support in carrying out tasks swiftly and should allow users to return to other jobs as quickly as possible.
Q: Can staff add a booking within 3 ½ minutes? This allows staff adequate time to gather and enter all necessary details whilst providing a profession and high degree of customer service.
Test 1 & 2: Acting as a customer (1. in person/2. phone), record the time it takes participants to enter a set of pre-determined data.
Test 3: Can staff follow a range of instructions, identifying and placing bookings and reminders within 3 ½ minutes whilst making no more than 3 mistakes?

Safety: The system should provide adequate validation so the user and administrator can only select or input appropriate data, such as not being able to input a date in the past. The system should provide a range of error prevention and measures to recover from errors such as an undo button.
Q: Is data validation provided to ensure correct dates, times, email addresses, phone numbers, and text entry fields can only accept specific data?
Test 1: Confirm only valid information can be entered for each user input.
Test 2: Confirm undo functionality works as expected in terms of undo limit and correct data types as decided between developers and users.

Utility: The system should provide a range a range of functionality, appropriate to the users' needs and requirements.
Q: Can administrators carry out the range of tasks that allow them to manage bookings, retrieve guest information and respond to enquiries in a way that is logical or in a preferred way?
Test: Test that a range of tasks can be carried out using multiple methods: menus, buttons, drag-and-drop, etc.

Learnability: A top priority: should allow swift onboarding and understanding of system functionality. The layout should be cohesive, organised, and easy to understand.
Q: Can a new user learn to perform the basic tasks of making a booking, adjusting a booking and cancelling a booking within an hour (time is based on interview feedback)?
Test 1: Users must match the correct word with the correct icon that represents its functionality.
Test 2: Time how long it takes a user to remember the sequence of a set of essential tasks without making a mistake or additional guidance.

Memorability: The System should provide prompts and carefully considered design to enable recurring tasks without needing to refer to documentation or guides.
Q: Do the prompts and interface design support administrators in carrying out their task? Can they remember how to carry out a specific task after a set period of time such as 2 weeks?
Test: User must perform a complex task from memory without mistakes, using only the available interface design for assistance.

User Experience Goals:

- Helpful: Should allow easy management of bookings.
- Supportive: Should provide intuitive controls to carry out tasks in a variety of ways.
- Satisfaction: Responsive controls, system feedback and the use of essential functionality should evoke a feeling of satisfaction.
- Enhancing sociability: The system provides opportunities to bring people together.

Domain Modelling

Business Rules

1. If a guest has a booking, they must arrive within 30 minutes, or the booking will be marked as a 'no-show' and the space becomes available for other guests.
2. 30 minutes before the guests are required to vacate the table, staff must inform the guest that they have 30 minutes left until their booking ends. The staff in charge of each section will update the booking to reflect that the customer has been informed.
3. 15 minutes before the guests are required to vacate the table, staff must present the guest with an itemised bill for the session. The staff in charge of each section will update the booking in the system to reflect the guest acknowledgement and will continue to manage the payment through a separate system.
4. A table must be marked as clear after the guest have left to indicate it is ready for the next booking.
5. Tables support a specific number of guests, no less.
6. Tables can be combined to cater for larger groups where necessary and available.
7. A venue can have multiple sections (Bar, main area, garden, roof, private rooms, etc).
8. Bookings cannot exceed a specific length of time, determined by the venue manager.
9. Reminder emails will be sent out at least 48 hours before the guests booking time.
10. Guests cannot cancel the booking within 24 hours of the booking commencing.
11. For any booking, the minimum is required: first name, last name, party size, start time, date and contact details.
12. The last start time available for booking will be 1.5 hours before closing time. E.g., if the venue closes at 11:00PM, last bookings can start from 9:30PM. (This allows at least 30 minutes for customer overrun and precleaning to occur).
13. Cost of booking is dependent on:
    a. Length of session (minimum of 1 hour)
    b. Table type (interactive/special gaming tables cost more)

c. Group size.

14. The minimum booking length is 1 hour.

Business Processes and Use Case Model

The following activity diagrams represent the most common processes carried out on a daily basis, each providing an essential functional component of the system and acting as initiators for developing extended use cases, development and prioritization of functional and quality requirements, potential classes, and individual roles and responsibilities for conceptual and analysis diagrams. They are designed to help clients understand, confirm and communicate different scenarios and avenues of circumstances that can occur for each given point of contention and show the sequence of events for each process.

1. Check-in Guest (figure 4): The guest provides the host with their reference number and name so the correct booking can be located. If none exists, staff can add a walk-in or future booking, or the guest can leave and end the process. If the system finds the associated booking, the host confirms the details, updates it status and escorts the guests to their table.
2. Update Status: (figure 5): When bookings reach a specific time, the status and colour will be modified in the order: Arriving soon -> Check-in -> 30-minute warning -> 15-minute warning (payment) -> Cleared. Guests that don't arrive within 30 minutes of their reservation beginning will be marked as a 'no-show', removed from the display and be delt with at end of day.
3. Add New Booking (figure 6): Guests can attempt to reserve a table at a specified date and time lasting no longer than the maximum length of time specified by the venue manager. Each venue has a variety of sections with tables that seat a range of 2 – 8 people and can be combined to accommodate more. Price is offered based on party size and the use of the board game collection. Alternative options are offered for no availability. If offer accepted, guests provide additional necessary details, confirm the booking and are send an acknowledgement email. Online bookings are either accepted or guests fill out and return an enquiry form.
4. Modify Existing Booking (figure 7): The guest must provide their booking details for staff to locate the matching booking and only then can alterations be made. If there are any conflicts, alternatives are provided until and agreement can be made and if none are available the guest can cancel or keep their original reservation.
5. Cancel Booking (figure 8): Guest provide their details, either in person or via email, and staff locate the matching reservation. Once identified, the booking will be removed, and a confirmation email is sent to the guest.
6. Reply to Enquiry (figure 9): Emails are managed through the system and displayed in the enquiries tab. When an enquiry is received, staff can navigate between multiple views to retrieve alternative offers and respond with custom or pre-defined messages, add attachments and provide alternative venue/date/time options. When sending, staff can also choose to add a pre-confirmation booking component to the booking view page ('take') or confirm the booking and permanently add it to the booking page ('confirm'),

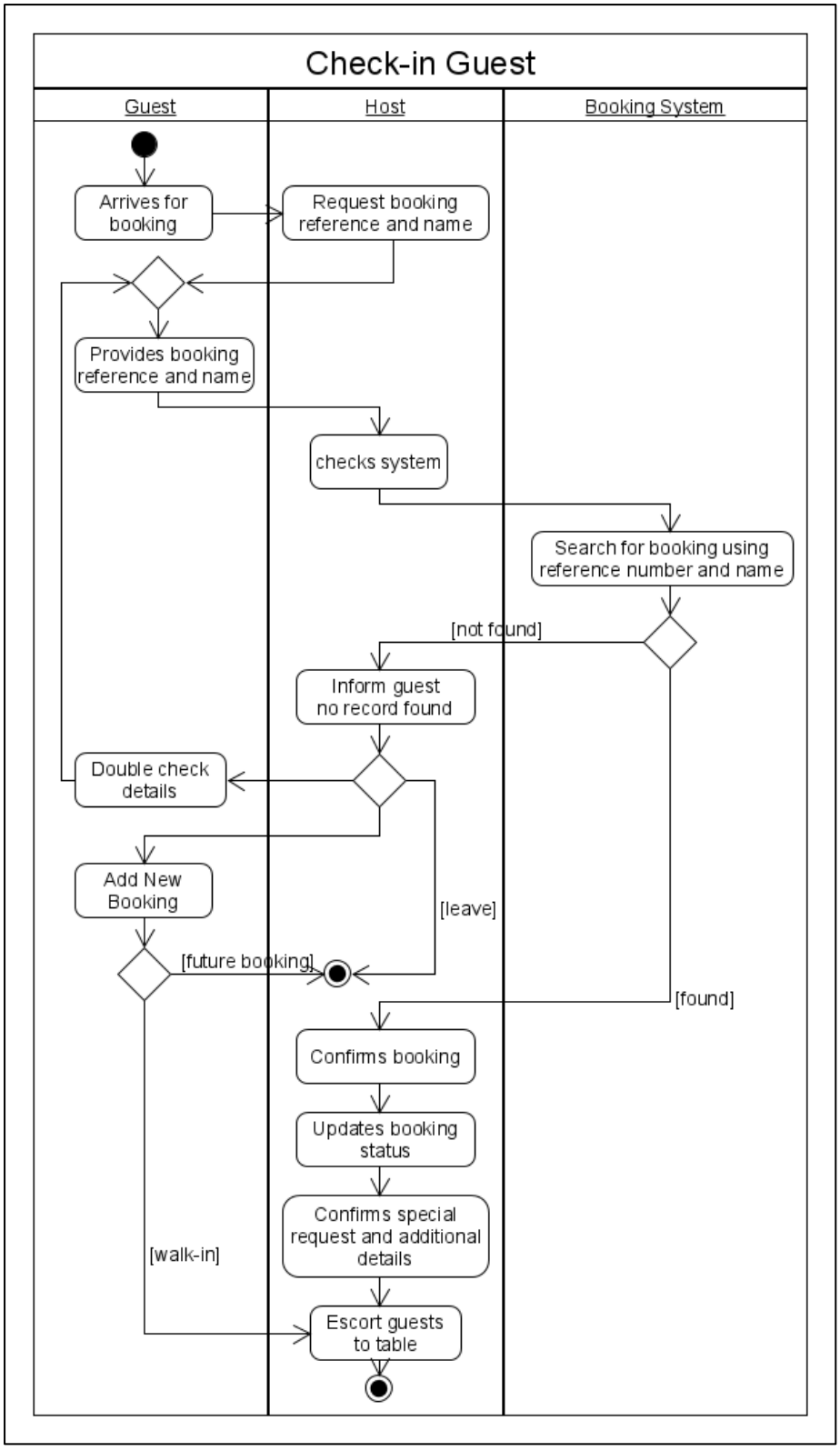

Figure 4: Check-in Guest

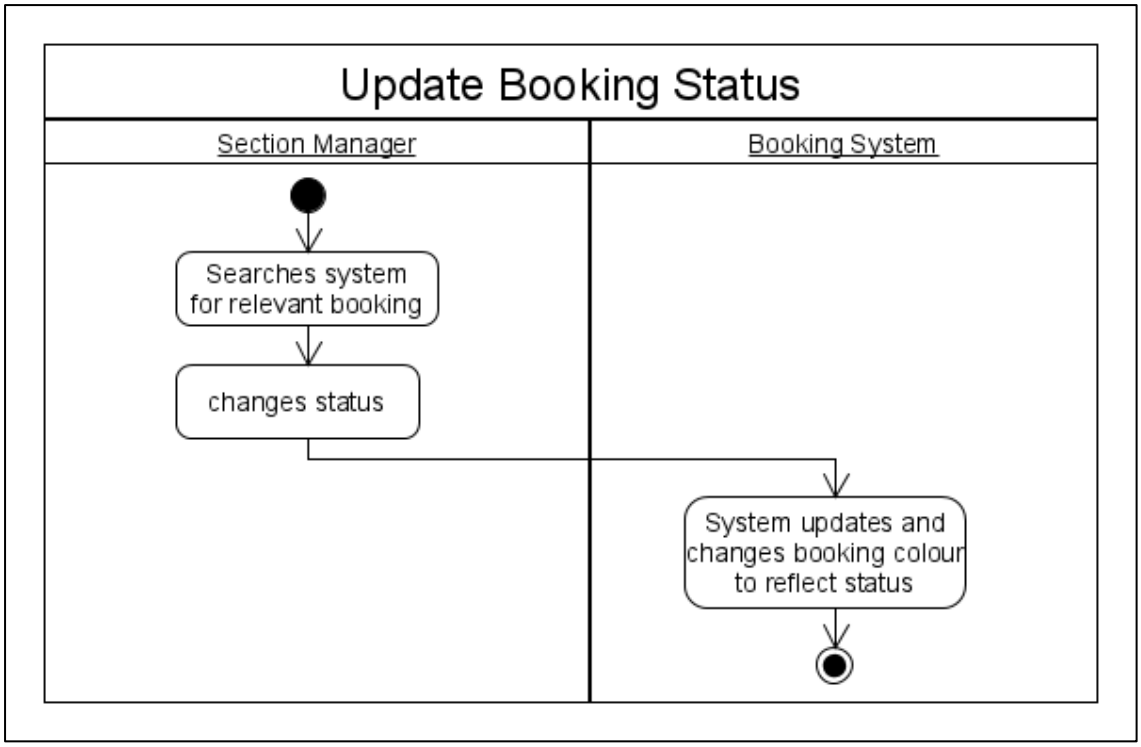

Figure 5: Update Booking Status

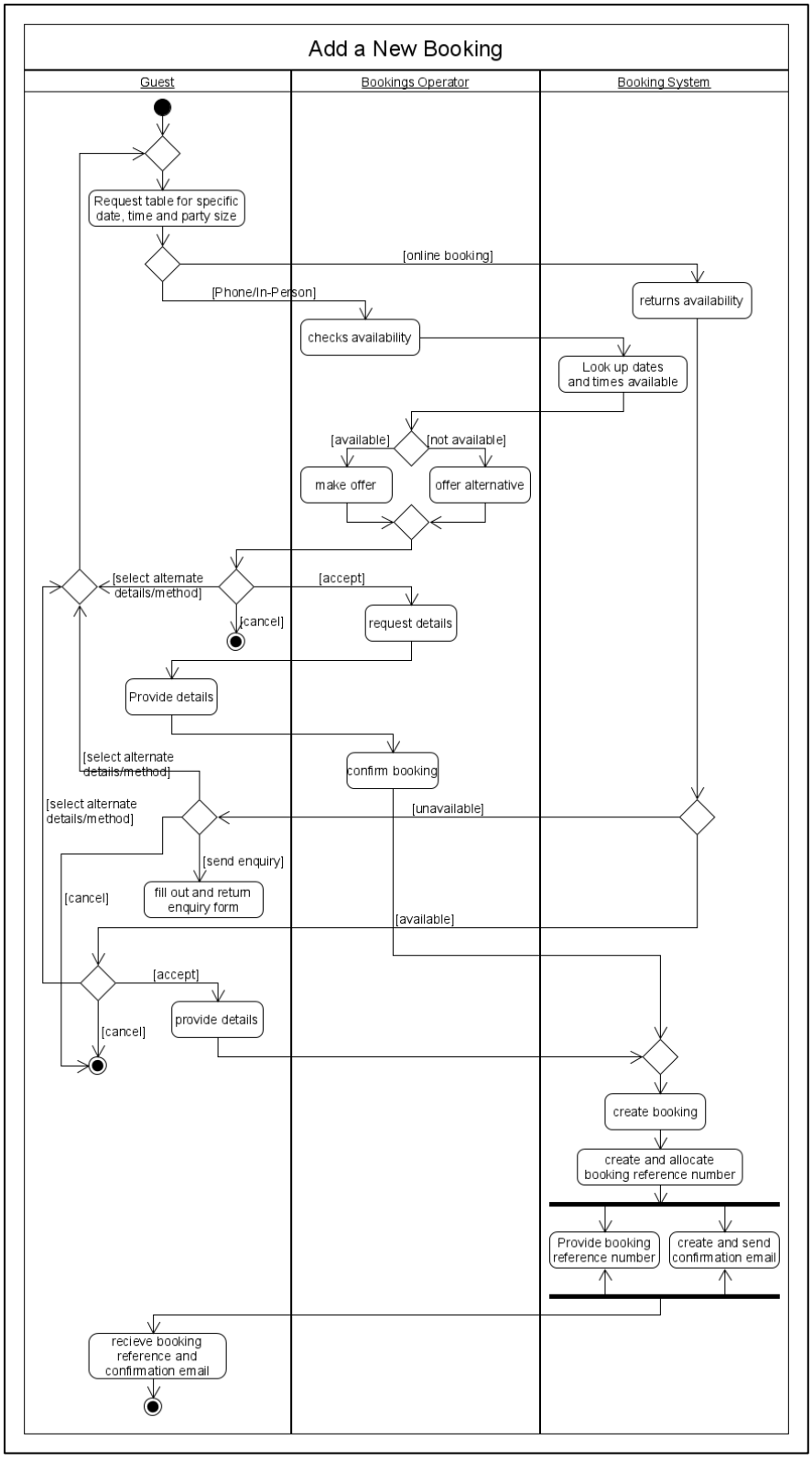

Figure 6: Add New Booking

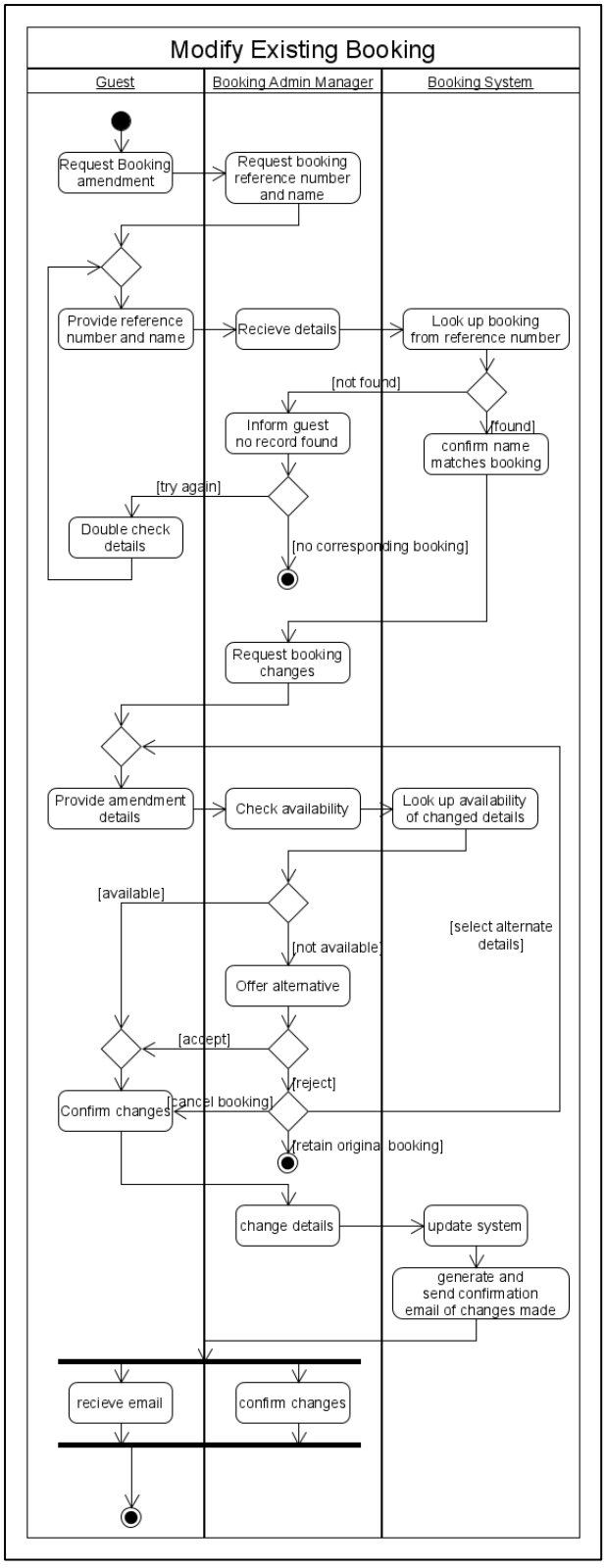

Figure 7: Modify Existing Booking

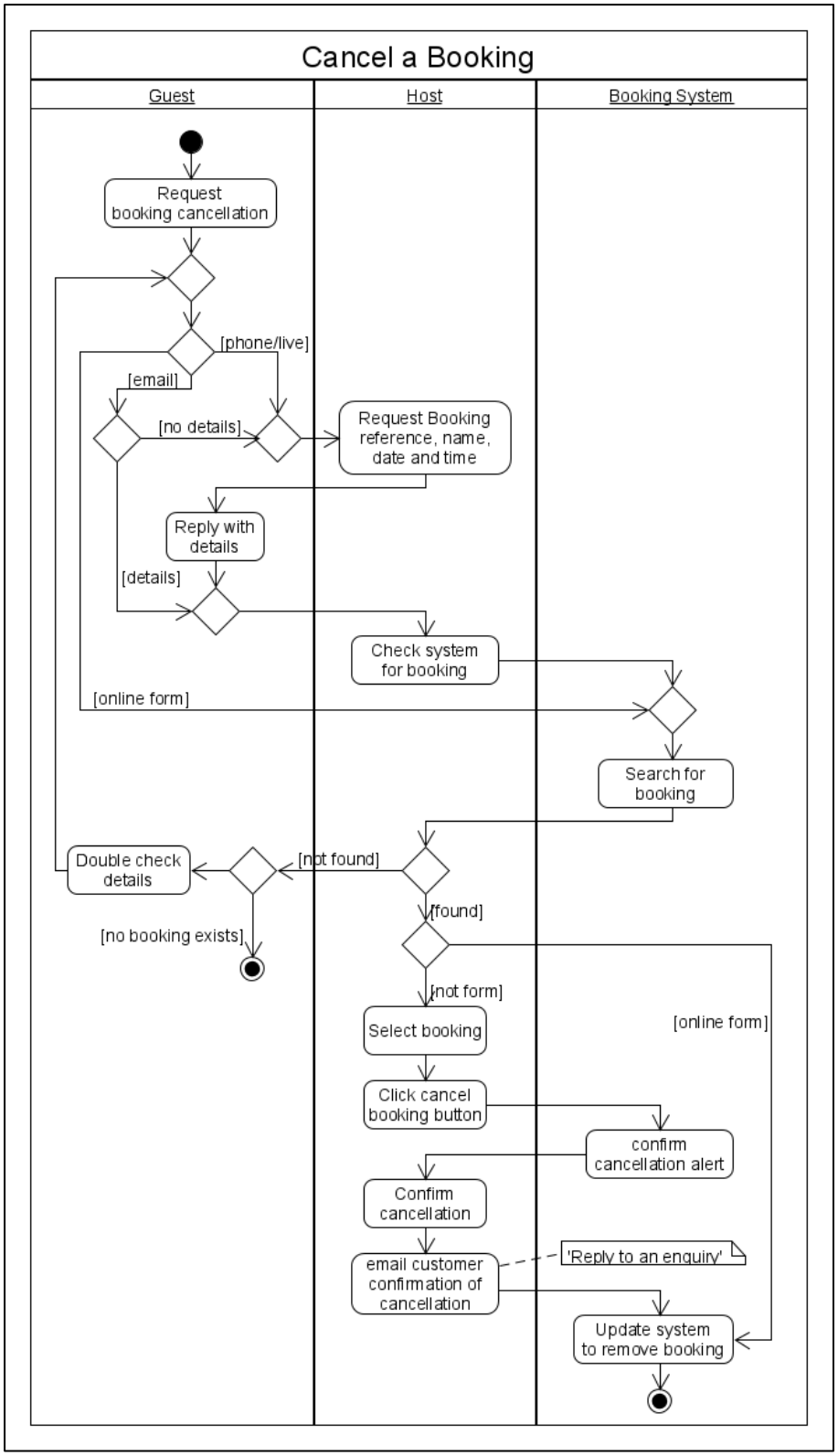

Figure 8: Cancel a Booking

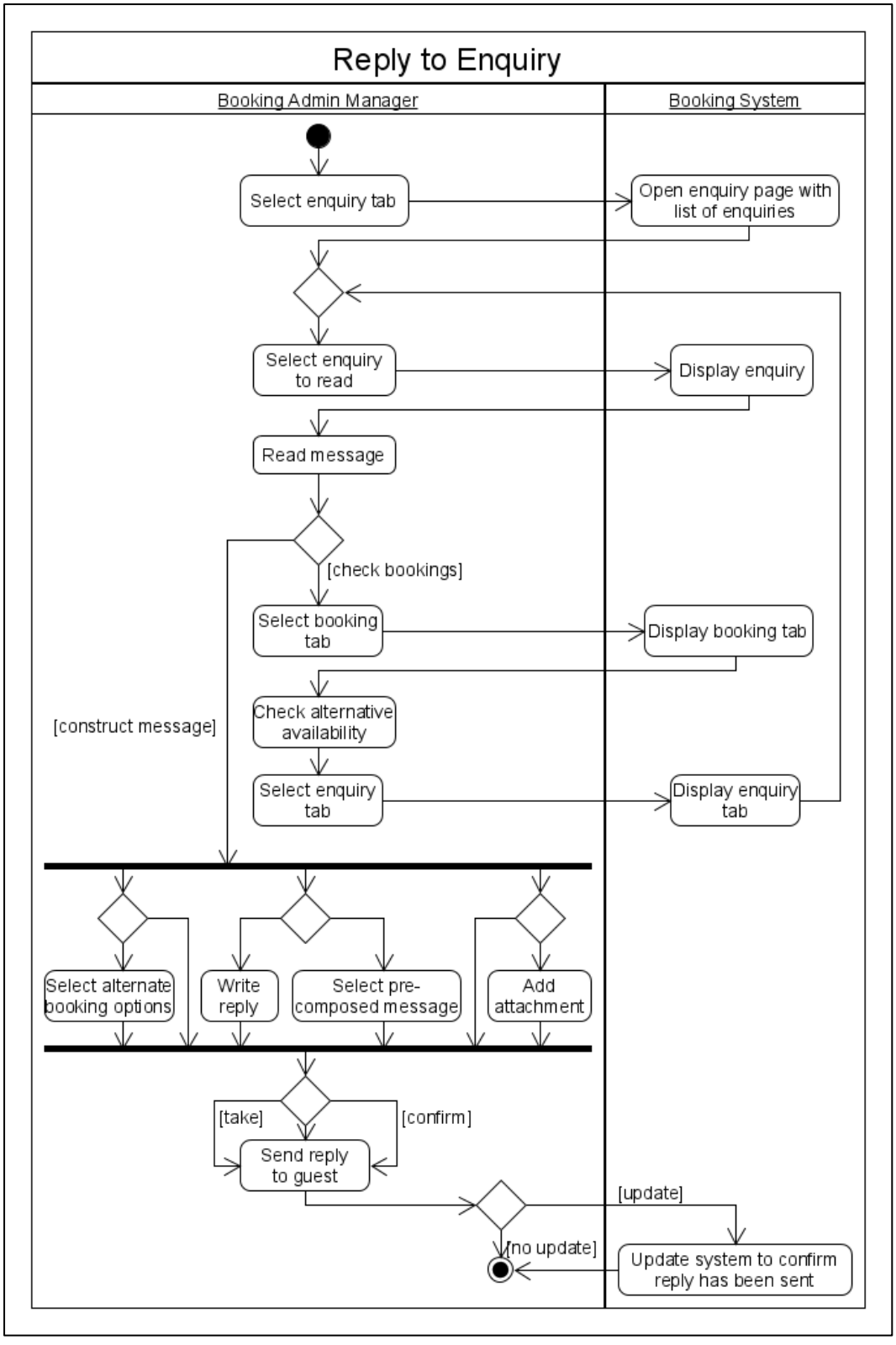

Figure 9: Reply to Enquiry

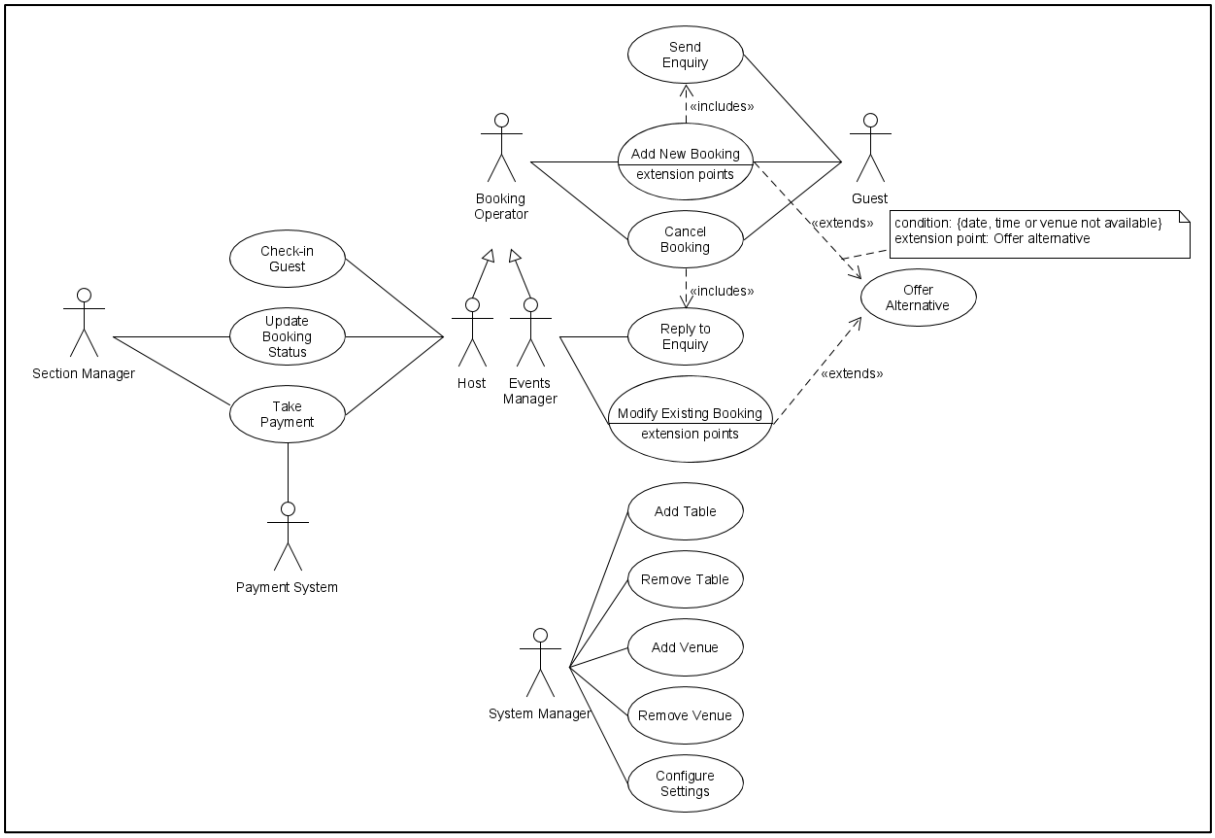

Figure 10: Use case model for initial business processes

The use case model in figure 10 is derived from analysing the business processes, factoring out common extensions and identifying generalised roles. Use cases for a system management role have also been derived from the business processes as the system requires venues and tables to exist before other processes can be completed. This also provides potential candidate classes for conceptual and analysis models in preparation for the implementation phases.

Elaborated Use Cases and Functional Requirements

The following use cases provide formal declarations of essential processes and establishes a potential basis for each implementation phase's goal, represented by the included functional requirement.

Identifier and name: UC2 *Add a new phone/in-person booking*
Initiator: *Guest*
Goal: A table in the venue is reserved for a guest
Precondition: None
Postcondition: A table of appropriate size will be reserved for the required date and time, and the table will no longer be available for that period.
Assumptions: The initiator is not known to the system. The guest has valid contact details. The volume of the environment allow for both staff and guest to have a conversation at a normal level.

Main success scenario:

1. The guest chooses to make a reservation.
2. The guest informs the member of staff of the date, time, and group size for the booking.
3. The member of staff searches the system for availability within the scope of the provided details.
4. The member of staff informs the guest that a table is available and the cost of admission with and without access to the venues boardgame selection.
5. The guest accepts the offer.
6. The member of staff requests additional details.
7. The guest provides their full name, contact information, any special requests and any games they wish to reserve.
8. The member of staff enters and confirms the details and submits the booking into the system.
9. The system creates the booking and allocates a unique booking reference number.
10. The system provides the guest with the reference number and sends an email to the guests provided email address that include the bookings confirmation details.

Extensions:

4.a A table is not available within the parameters of the given details.
4.a.1 The guest accepts an alternative offer of a combination of different venues, dates and times.
4.a.2 The guest does not accept the initial alternative and requests another alternative options.
4.a.3 The guest does not continue with the booking process.

8.a The guest requests for a booking with immediate availability.
8.a.1 The member of staff must also select the walk-in checkbox.

Requirements Number	*FR14*
Event/Use Case	UC1: Step 6
Description	The system shall create a booking from the provided parameters
Rational	Creates a booking in the system. The table at the associated date and time are no longer available. Any reserved board games are also no longer available.
Originator	Interviewee
Fit Criterion	A reservation will be created in the system for a given period on the provided date.
Customers dis/satisfaction	1 (Very Satisfied)
Priority	1 (Extremely Important)
Conflicts	All data needs to be of a valid type and a table must be in the corresponding section.

Supporting materials	Use Case 1 & 2, FR1, FR 2, FR3, FR3, FR8, FR9, FR10, FR11, FR12, FR13, FR25
History	15.07.2023 FR Created

Identifier and name: UC4 *check-in guest*
Initiator: *Guest*
Goal: acknowledge a guest has arrived and amend the booking.
Precondition: There is an existing booking in the system with a valid unique booking reference number.
Postcondition: The system has been updated to acknowledged the guests arrival, they have been escorted to their table and it is no longer available for the duration of their booking.
Assumptions: The party size does not exceed the tables maximum capacity. The guest can navigate the venue using its accessibility facilities where necessary and easily access the physical space surrounding the table.

Main success scenario:

1. The guest arrives for their booking.
2. The host requests the guests booking reference number or name.
3. The guest provides their booking details.
4. The host searches the system for the booking using the supplied credentials.
5. The host confirms the booking.
6. The host updates the booking status to acknowledge the guest's arrival.
7. The host acknowledge and confirms any special requests and necessary details.
8. The host escorts the guest/s to their table.

Extensions:

5.a There is no booking in the system with matching provided details.
5.a.1 The guest decides to leave and stops any continued processing.
5.a.2 The guest decided to continue by beginning the 'Add New Booking' process (see UC2 *Add a new phone/in-person booking*).
5.a.3 The guest double checks they have provided the host with the correct details and reverts back to step 3 in the main success process.

Requirements Number	*FR20*
Event/Use Case	UC3: Step 6
Description	The system will update the bookings tab view to display all bookings on the same date
Rational	Any modifications, requests, and availability can be seen in a single view
Originator	Developer
Fit Criterion	The bookings tab view will show all bookings on a specific date

Customers dis/satisfaction	2 (Satisfied)
Priority	1 (Extremely Important)
Conflicts	N/A
Supporting materials	Use Case 3
History	18.07.2023 FR Created

Requirements Number	***FR18***
Event/Use Case	UC3: Step 4
Description	The system shall check for a booking with a matching unique booking reference number
Rational	Validates a user against a specific booking.
Originator	Developer
Fit Criterion	A valid booking reference shall be accepted
Customers dis/satisfaction	3 (Neutral)
Priority	3 (Important)
Conflicts	N/A
Supporting materials	Use Case 3
History	18.07.2023 FR Created

Requirements Number	***FR19***
Event/Use Case	UC3: Step 6
Description	The system will display the details of a booking
Rational	Details of a booking will be available to view for validation and comparison.
Originator	Developer
Fit Criterion	A bookings information will be revealed
Customers dis/satisfaction	2 (Satisfied)
Priority	1 (Extremely Important)
Conflicts	N/A
Supporting materials	Use Case 3
History	18.07.2023 FR Created

Identifier and Name: UC5 *Update booking status*
Initiator: *Booking System*
Goal: To alter the booking in the system to reflect its current status.
Precondition: There is an existing booking in the system. The booking has reached a specific

point relative to its end time. There is a specific sequence up updates as follows: Arriving soon -> Check-in -> 30 Minute warning -> 15 Minute warning and payment -> cleared.
Postcondition: The booking status and colour have been updated in the system.
Assumptions: The user understands concepts such as time and can differentiate between colours representing several status categories.
Notes: Each booking type will have its own customisable sequence of events and timings. The following main scenario is for the 'general admission' booking type.

Main success scenario:

1. The booking system requires a booking status change.
2. The section manager searches the system for the corresponding booking.
3. The section manager selects the corresponding booking.
4. The section manager selects the next update in the sequence.
5. The section manage confirms the change.
6. The booking system updates the display to reflect the changes made.

Requirements Number	***FR28***
Event/Use Case	UC4: Step 6
Description	The system shall update the bookings status
Rational	A bookings status and colour is modified to match a specific point in its lifetime.
Originator	Developer
Fit Criterion	The reminder status and colour matches a specific time period of the booking
Customers dis/satisfaction	1 (Very Satisfied)
Priority	1 (Extremely Important)
Conflicts	FR18: A booking must exist for it to be updated
Supporting materials	Use Case 4
History	18.07.2023 FR Created

User Interviews – Evaluating Prototypes

Having iterated over, derived and developed multiple extended use cases and requirements, it was imperative to ensure that before proceeding further, I was confident the work completed thus far was contributing with positive impact towards the final product. Utilising an iterative, agile, user-centred project development approach, the most effective method was to continue involving potential users and conducting a 2nd stage interview designed to address a number of purposes:

- Confirm and adjust existing rules, processes and requirements.
- Establish missed/additional processes, rules and requirements.
- Gather prototype feedback for analysis, evaluation and redesigns.

- Test success criteria using gamification techniques.
- Make sure the correct product is being built.
- Building of additional models and continued work

The participants, all of which were familiar with both playing boardgames and using booking systems, facilitated designing the interview to incorporate gamification techniques (Silva, et al., 2023) to encourage participation through cooperative and competitive testing methods and determine if:

1. Everyone is able to understand the software to an acceptable level by evaluating interview results against a number of test criteria.
2. Users are having positive responses to engaging with the prototypes and the results of tests are meeting success criteria.

The interview was split into five parts and was developed around the idea of each user gradually building their own unique interface design from multiple components. Each part was designed to be tested against multiple success criteria and as a way for users to accumulate points depending on how well they performed individually and cooperatively.

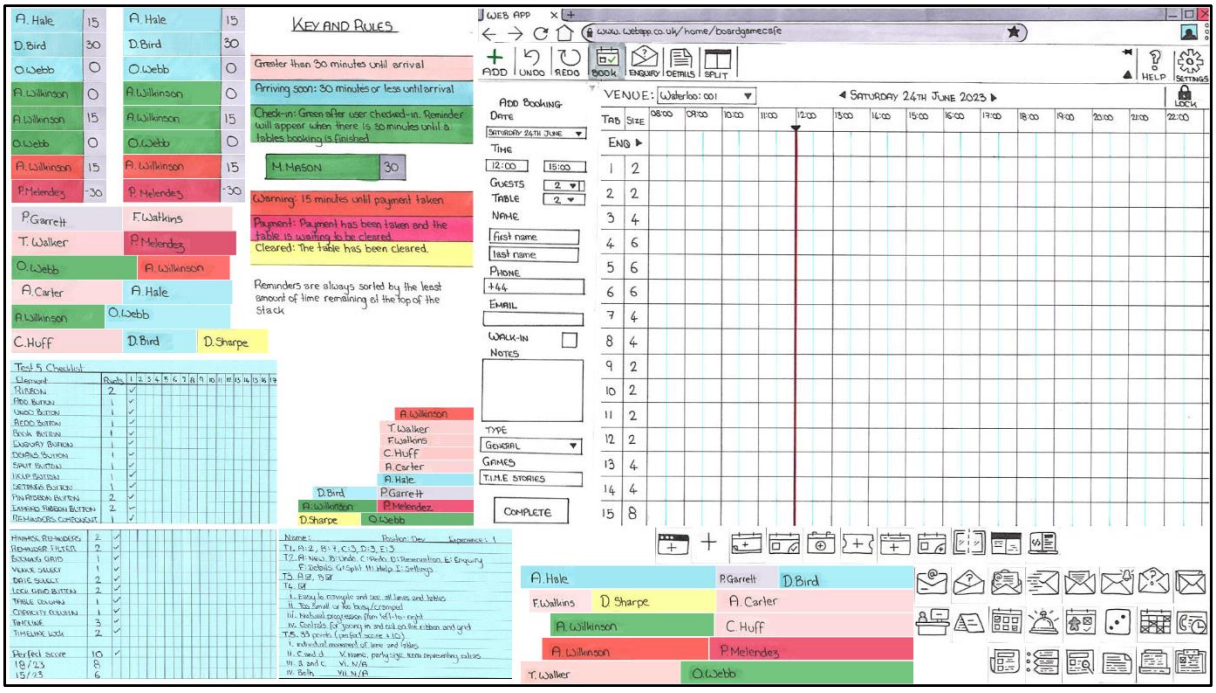

Figure 11: Components - Top left going clockwise: small and large reminder components, Key and rules of reminder status and colours, web browser bar, web application ribbon, Fullscreen grid view with add booking details component, ribbon icon groups representing functionality, Zoomed-in grid booking components, participant answer card example and compact booking grid components (above), participant identification checklist back and front, Fullscreen grid components.

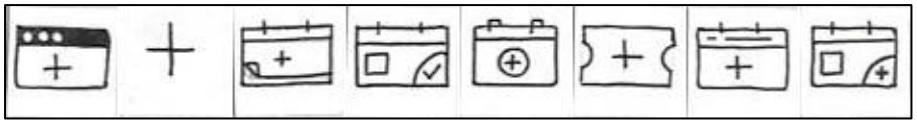

Figure 12: Icon group A1 - A8

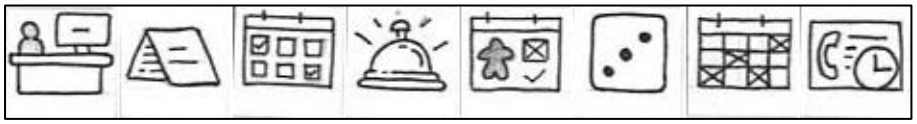

Figure 13: Icon group B1 - B8

Figure 14: Icon group C1 - C6

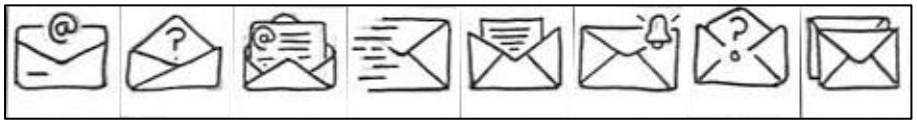

Figure 15: Icon group D1 - D8

Figure 16: Icon group E1 - E3

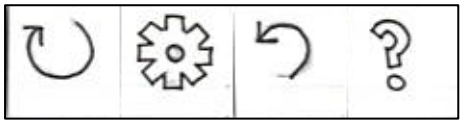

Figure 17: Icon group Redo, Settings, Undo, Help

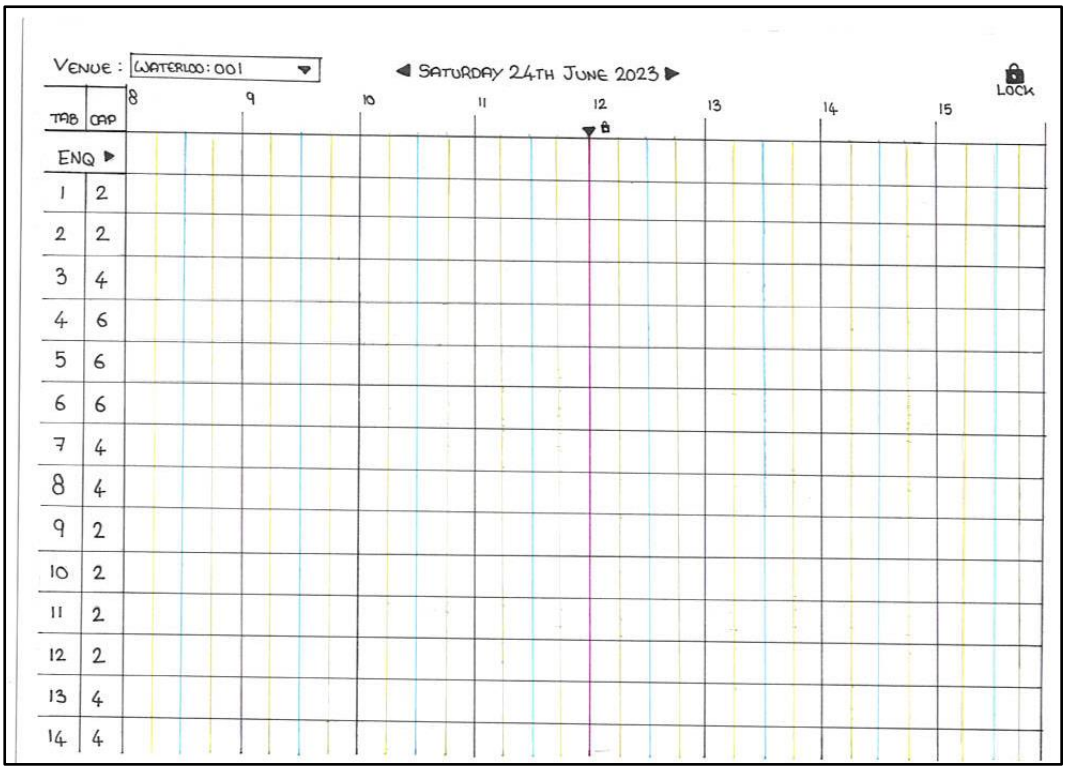

Figure 18: Zoomed-in grid component.

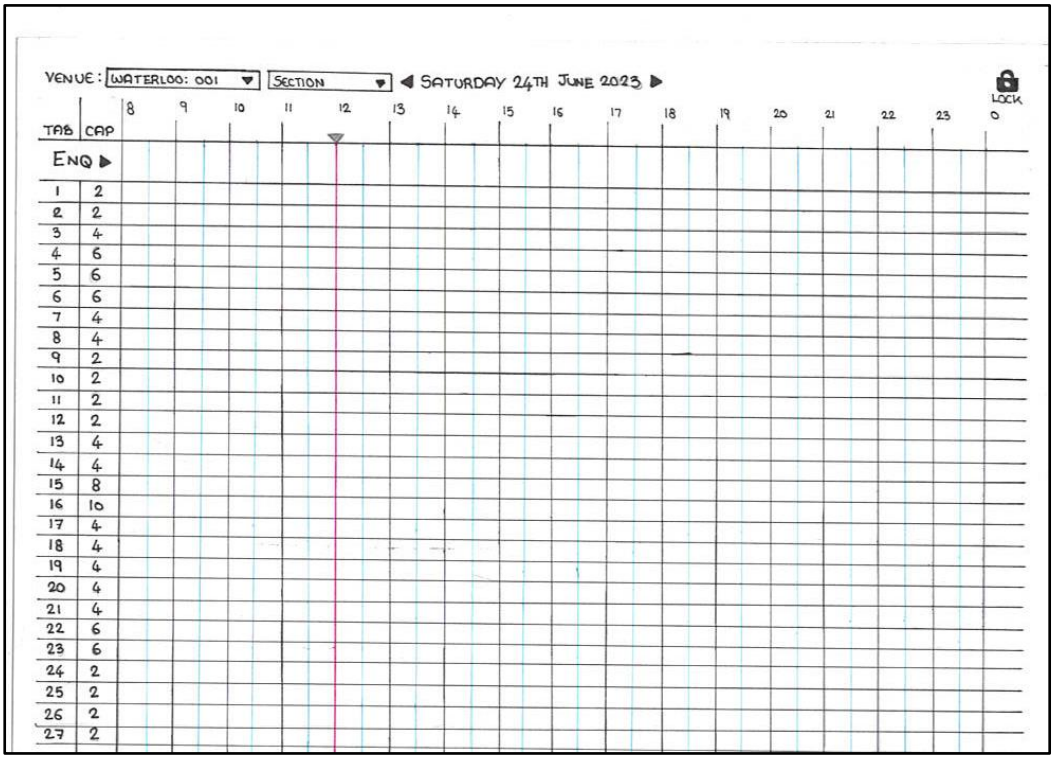

Figure 19: Compact grid component.

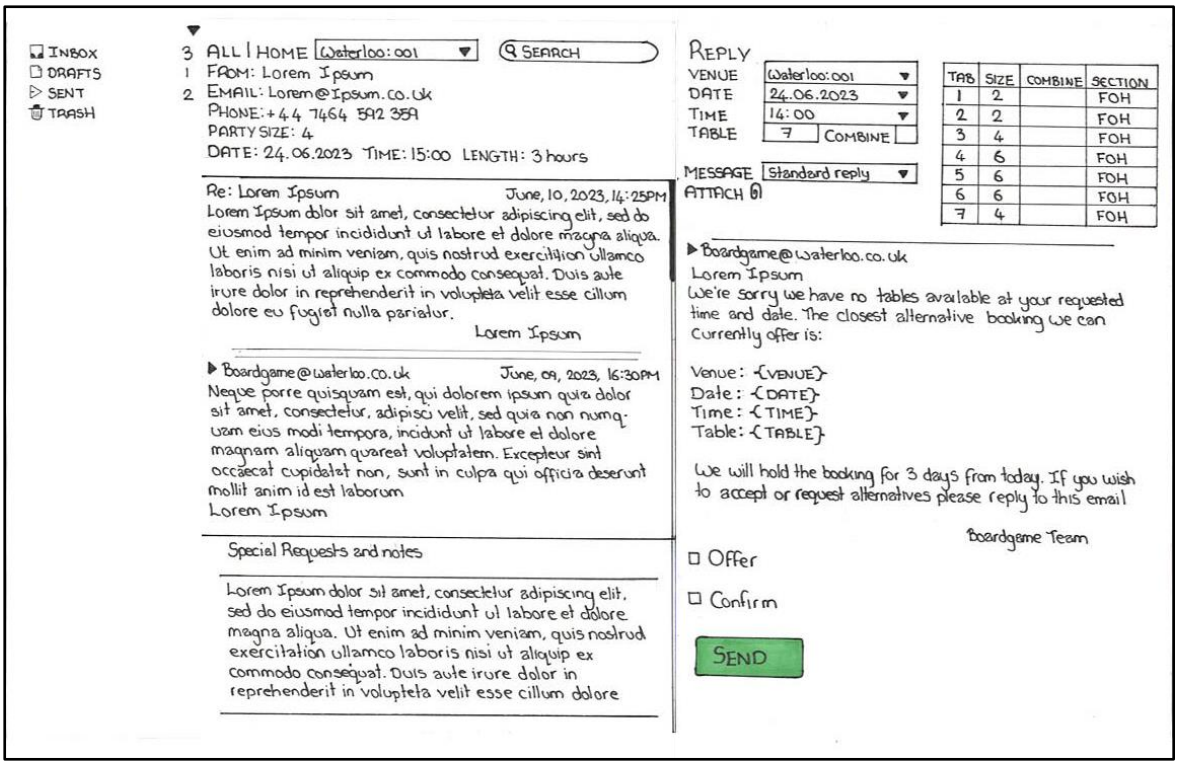

Figure 20: Enquiry view components.

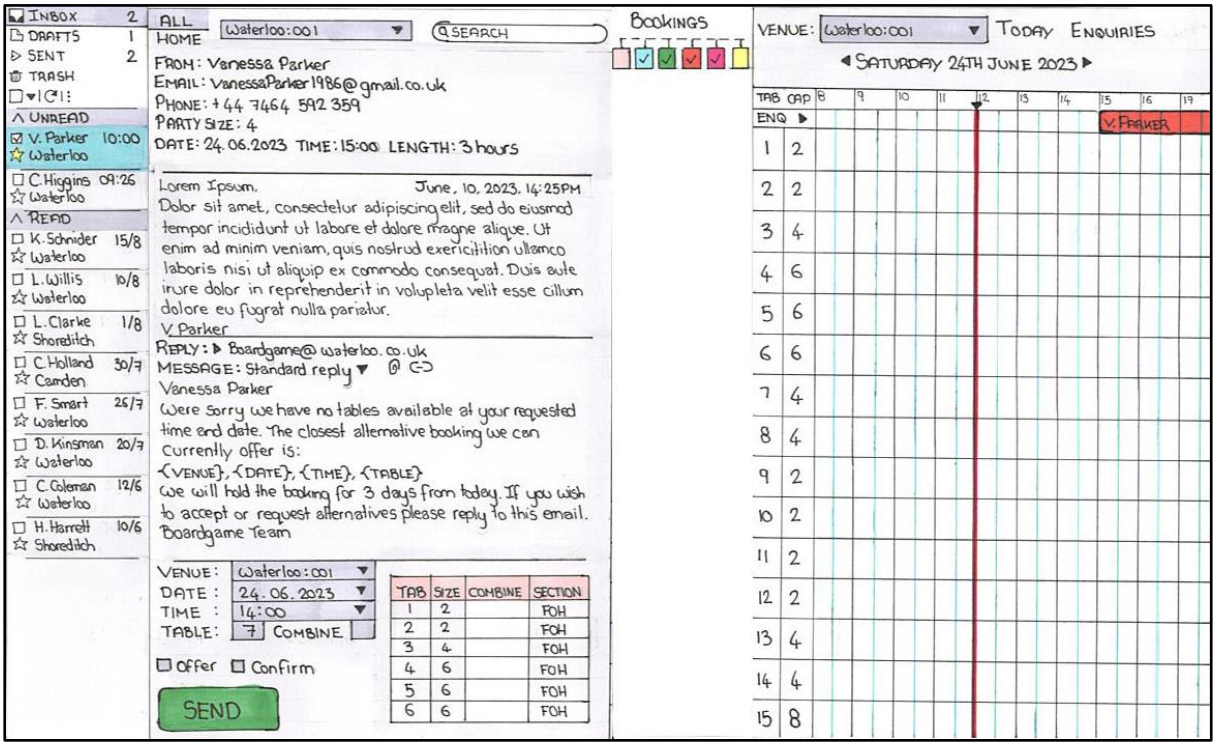

Figure 21: Split screen view components.

Each interview was conducted as follows:

Part 1: Beginning the interview:

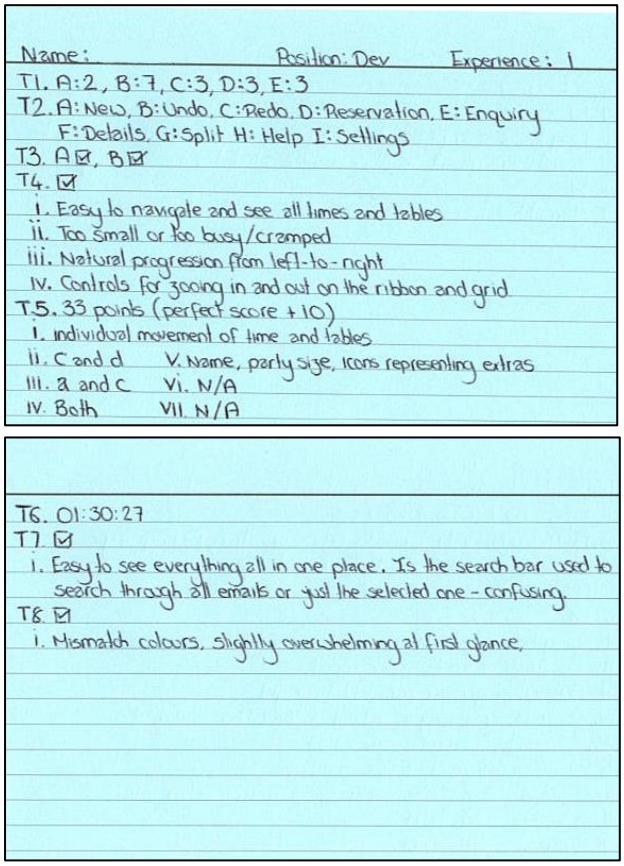

Figure 22: Example answer card (Top-front, Bottom-back)

Participants were provided with information sheets, consent forms and explained the purpose of the study. Each participants information and answers were recorded on individual cards and audio was recorded where permission was granted.

Each user was also asked to rate their experience of using their current system on a scale of 1 – 5.

1.I have none to very little experience with the system (such as you've only done a training session)
2.I have some experience with the system (Had a shift or two as host)
3.I have experience and feel comfortable with the system (Has completed a number of shifts as host)
4.I am experienced and feel comfortable with the system (Knows most of the systems functionality)
5.I am very experienced and extremely confident with the system (Knows most system functionality and configuration settings).

Part 2: Buttons, Icons, Skeuomorphism, and affordance

Test 1: Users were presented with categories of icons and asked to describe each set in a single word. The points allocated are based on the majority word receiving the highest points.

Test 2: Present each user with a number of words and their chosen icon and ask them to match one word to each icon. The word with the majority vote will receive the highest points.

Success Criteria: Learnability: Tests users understanding of buttons, cognitive abilities (decision making, perception, attention, and problem solving) and how well the icons represent their function/use. It also provides feedback on my understanding of the domain and users, ensuring the software being built is correct in terms of requirements and usability/user experience goals.

Part 3: The Grid View:

Test 3: Users were given component parts for the booking view and asked to assemble them in their preferred configuration.

Test 4: Users are given the option to change icons and associated words from part 2 to place on the ribbon in their respective positions.

A short question and answer section establishes why the configuration was chosen, was they liked and didn't like, why they placed components in specific positions and what could be improved or changed.

Test 5: Users are tested on their ability to recognise, identify and categorise all the components and elements of their chosen configuration. Points are uniquely allocated for each item correctly identified with additional bonus points reaching different targets.

A second question and answer section covering expectations regarding grid functionality such as zooming and navigation, essential and additional elements attached to booking components and how those are represented in their existing system.

Success Criteria: User experience goals (Helpful/Supportive): Utility and Safety: Attempts to establish a common layout, details about the addition elements (only realised from prototype design) might function, look, and work.

Part 4: Testing Effectiveness and Efficiency:

Test 6: Users are given the necessary components for their selected grid are asked to follow a set of instructions which involve placing them on the grid and reminder component in the correct order. The test is timed initially on how long it takes before the user understands the key and rules then how long it takes to complete the instructions. The test should finish with and a specific configuration with points awarded based on fastest times and each mistake being penalised by a single point.

Part 5: Enquiries and split view initial impressions:

Test 7: Users who have experience with replying to enquiries are asked to assemble the enquiry view components and asked their initial impressions of the elements that make up the view.

Test 8: Advanced users (4 or 5) are provided with the split screen elements and asked to assemble them in their preferred configuration and for their impressions and any changes they would make.

Part 6: Reset

Thank participants for their involvement, provide users with story cards for any additional system requirements or thoughts to be returned during the next user participation and feedback session, and reset all the elements.

The results from the interviews provided both qualitative and quantitative data; common decisions and choices, understanding of elements and their function, additional requirements and changes to the interface to name a few.

Tests 1, 2 and 4 showed that given a set of icons, in the context of the booking system the most popular choice of icons and words to represent their meaning were:

- A3: Add
- B3: Reservation
- C2: Information
- D7: Enquiry
- E2: Split
- Undo, Redo, Help and Settings

Test 3 showed that the most popular choice for grid was the zoomed-in view. It provided a level of detail that made it easy to see all the bookings when quickly scanning, determine the time using the different coloured vertical dividers, and was easy to see the details of the booking components on the grid. It was also found that military time was preferred by most as it provided a clear representation of the time than simply a number at the top of the grid. The compact view was determined to be good when looking at the details of the day overall, but too difficult to see the details of individual bookings when searching and slightly overwhelming for those who were less experienced. The full screen view also proved to be popular but left no space around the edges of the grid and felt cramped with the other elements.

The most popular reminders component was the basic filter as it was much easier to understand the priority of updating guests and the status of bookings. The placement was also placed left of the grid for the majority as it felt natural reading left to right for participants.

Test 5 revealed that 100% of users understood at least 18 out of the 23 elements listed with a small number of more experienced users achieving a perfect score.

Comparing test 6 results against the success criteria indicated that 90% of participants were able to complete the test in under 3:30 minutes with fewer than 2 mistakes, and were able to understand the test components, instructions and functionality within 2 minutes.

Finally, initial feedback on the enquiry view and split screen view was that elements should have a little more space between them to distinguish them better such as between the message details and message body, and the search bar should be moved to above the inbox to indicate users are searching for specific enquiries, otherwise, the layouts provided everything necessary to manage enquiries and work in split screen.

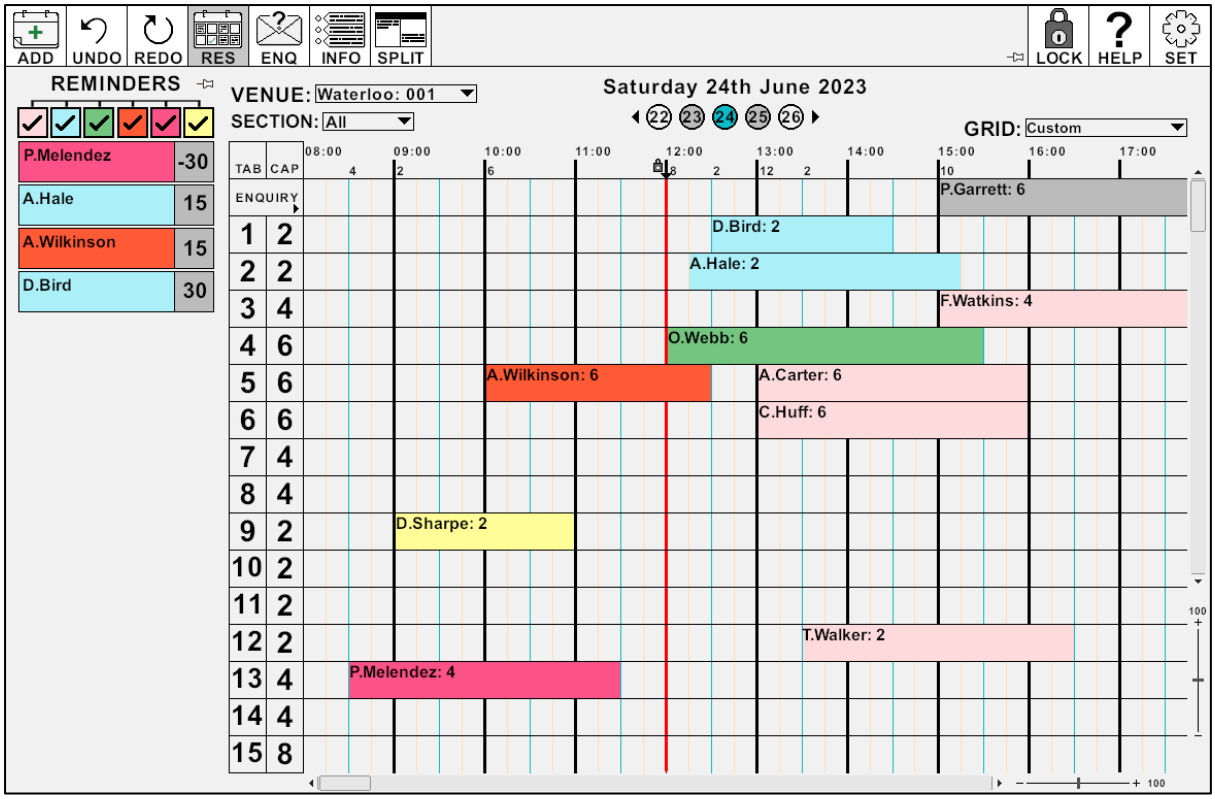

Figure 23: Redesigned interface of the booking system reservation view

Figure 23 (above) displays the redesigned interface of the systems reservation view based on the analysis, and evaluation of interview and test data. Additional zoom and navigation controls have been added, and elements from multiple grids and icons have been combined to better represent the majority of user's feedback.

Structural Design Models and preparing for implementation

Before the implementation phase and any coding can begin, I developed a number of object diagrams to assist in building a structural design model, prioritising development goals and assigning the responsibilities of operations to specific classes. The following diagram provides the initial structure of the application, classes and their attributes and the associations between them. The process involved systematically identifying potential classes via the problem description and developed artefacts, use cases, and functional requirement then filtering them providing justifications for class.

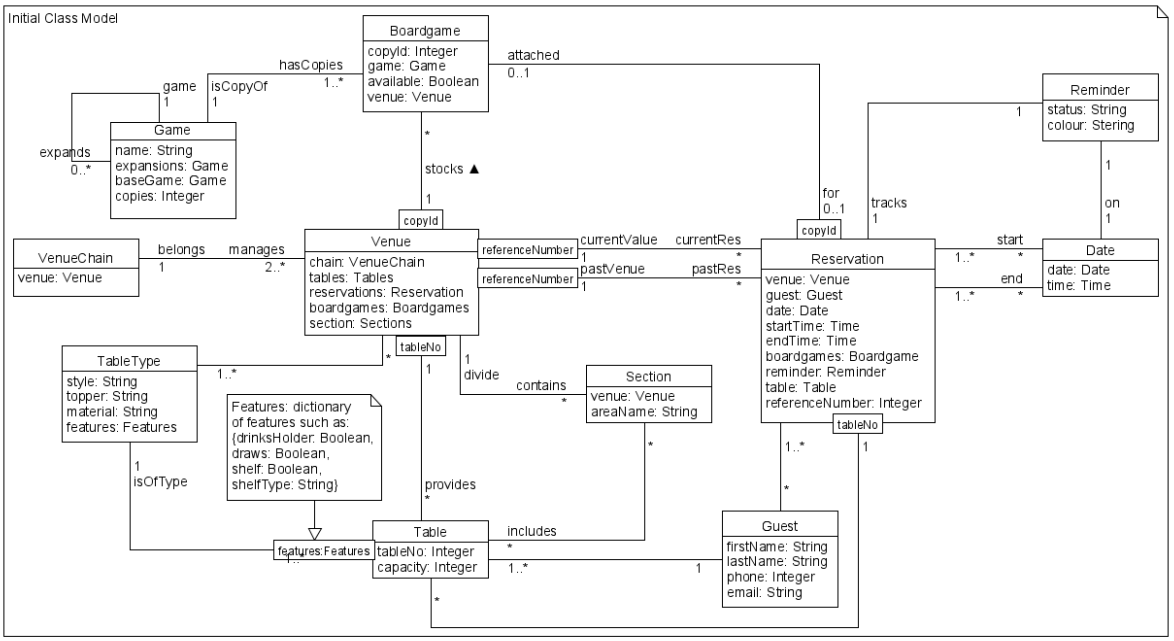

Figure 24: Initial structural diagram for the booking system

Implementation phase 1:

Rather than focusing on implementing the design of the interface the goal of this initial sprint is to implement basic system configuration functionality. Without this, it won't be possible to implement any other requirements so this takes priority. The following functional requirements will need to be completed during this sprint which is expected to last a week:

1. FR21: A new venue will be added to the system.
2. FR22: Adds a new table at a specific venue.
3. FR29: An existing venue will be removed from the system.
4. FR23: Removes a table at a specific venue.
5. FR13: Adds a new section at the specified venue.
6. FR30: Remove an existing section from the system.

Implementation phase 2:

This iteration will focus on implementing use case 2's success scenario: Add a new booking, and its relevant functional requirements including:

1. FR1: The system shall obtain a venue name from the user.
2. FR2: The system shall obtain a date from the user.
3. FR3: The system shall obtain a time from the user.
4. FR4: The system shall obtain a party size from the user.
5. FR8: The system shall obtain a full name from the user.
6. FR9: The system shall obtain an email address from the user.
7. FR10: The system shall obtain a phone number from the user.
8. FR12: The system shall obtain any special requirements from the user.

9. FR25: The system shall obtain a table number from the user.

Implementation phase 3:

With some of the core elements now implemented, the next phase is aimed at implementing the interface components to display controls and visual representations of stored data. This includes using web programming languages and frameworks to build out the interface, implementing use case 4: check-in guest and the following functional requirements:

1. FR18: The system shall check for a booking with a matching unique booking reference number.
2. FR20: The system will update the reservation view to display all bookings on the same day. This includes developing and managing the Date class.

Implementation phase 4:

The goal of this iteration will be to implement use case 5's main success scenario: updating a bookings status including FR28: The system shall update the booking status.

5. Review of Project Work

My initial goal was to have a working prototype developed and prepared for feedback after completing a number of implementation phases and while my project didn't quite reach this stage it came very close. I knew that the workload would be significant, trying to build a solution using software engineering methodologies while simultaneously focusing on a user centred approach to the interaction design aspect curtailed the time I could spend on each discipline. At this point of the project the only further project work would be to conduct the sprint phases outlined previously and conduct a final evaluation of the working prototype with the previous participants via a 3rd interview stage, testing success criteria to ensure progress is being made as intended. As an estimate, the project would require at least a month to complete these final activities, a week for each sprint, while also designing and conducting the final evaluation and testing interviews. I feel the quality of the work produced has enabled me to being development on a solution that will confidently meet all the success criteria.

Further additional development outside the scope of the project would be to continue iterating through the different lifecycle phases to implement the various different management views for dealing with enquiries, the info view and split views, adding reporting functionality and modules aimed at delivering services to different hospitality sectors such as restaurants and hotels.

6.Review of project management

The projects chosen lifecycle model provided effective management, allowing me to iterate and switch easily between the different phases of planning, analysis, research, design, development and evaluation. The most difficult aspect I found was trying to plan out all the tasks of the project as my approach was intended to be agile and adapt to change easily when necessary. I found this to stifle my ability to let the project determine what techniques and methods to use as having everything planned down to the amount of hours I'd spent on specific tasks counterproductive, and not conducive to an agile approach. Despite this, I managed to adjust my project plan and tasks through the iteration of phases to better suit the projects needs. If I were to do the project again, I would divide up my time to focus on producing a working example quicker and ensure that I'm allocating time appropriately. One shock I had was the realisation of the amount of work and effort required to constantly go back and forth between the documentation to ensure continuity and ensure no contradictions occur between any of the documentation. As I am only one person, this took up a lot more time than anticipated and in the future, I would aim to keep projects a lot more agile and attempt to produce only documentation that is truly instrumental to the project.

7. Review of personal Development

One skill I feel I've developed considerably is planning, conducting, analysing and evaluating interviews. From determining the purpose and what results I want to achieve, to understanding how I can relate questions to answer specific criteria important for the projects development. I found I've been able to manage and conduct interviews in a context that's appropriate for the style of users while still being fun and providing necessary data. While the project didn't fully meet my objectives in terms of a demonstratable product being produced, I do feel better equipped to manage a project, from identifying the time it will take to conduct specific tasks and reach project milestones to identifying and scheduling time sensitive tasks. I have put a significant amount of time, energy and work into this project and while I am proud of all that I have achieved I am disappointed that I didn't find the opportunity to develop my skills of database management and incorporate it into the project. This is something that on completion of this project, I will surely rectify.

Word count (8804)

REFERENCES

G2, 2023. *Highest Rated Online Appointment Schedulkking Software.* [Online]
Available at: https://www.g2.com/categories/online-appointment-scheduling?tab=highest_rated
[Accessed 13 April 2023].

Harrington, J. L., 2016. *Relational Database Design and Implementation 4th Edition.* 4th ed. s.l.:Morgan Kaufmann.

Llaneta, J. C. E. et al., 2022. *TerraVet: A mobile and web application framework for pet owners and veterinary clinic.* Harbin China, ACM, pp. 19 - 24.

Preece, J., Rogers, Y. & Sharp, H., 2015. In: *Interaction Design: beyond human-computer interaction.* 4 ed. s.l.:Wiley, p. 443.

Silva, D. et al., 2023. *Applying Gamification to Prioritize Requirements in Agile Projects.* Tallinn Estonia, ACM, pp. 1498 - 1507.

Teuber, C. & Forbrig, P., 2004. Different Types of Patterns for Online-Booking Systems. *TAMODIA 2004: Proceedings of the 3rd annual conference on Task models and diagrams,* 15 November.pp. 91 - 97.

Weigers, K. E., 1999. First Things First: Prioritzing Requirements. pp. 1 - 6.

APPENDIX

For a full list of additional artefacts relating to this project, please see the appendix.