

APPENDIX

TABLE OF CONTENTS

Appendix.....	1
Project Management	2
Tasks & Subtask	2
Domain Modelling	6
Problem Statement.....	6
Success Criteria	8
Business Rules.....	10
Business Processes.....	11
Requirements.....	17
Use Case Models.....	17
Elaborated Use Cases.....	17
Functional Requirements.....	22
Design	33
Prototypes and Component Parts	33
Design Structural Models.....	44
Behaviour Models	47
Data Gathering.....	48
Identified Venues and Systems.....	48
System Users.....	48
Stage 1: Semi-Structured Interview Questions.....	48
Venue Observation Details	49
Participant Observation Details	50
2 nd Stage Interview	51
Literature Search.....	56
Glossary.....	60
References	63

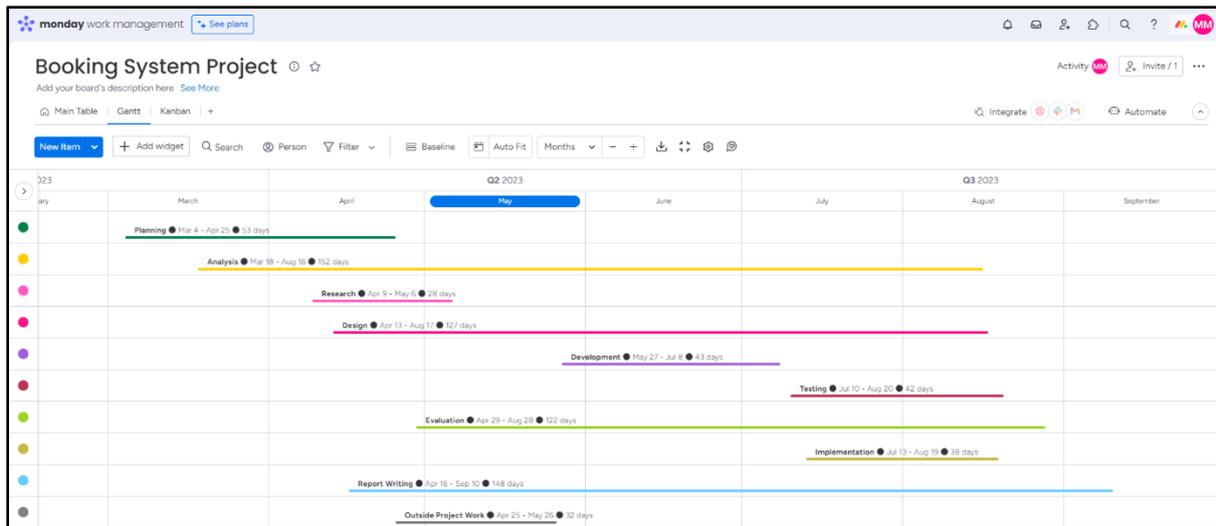
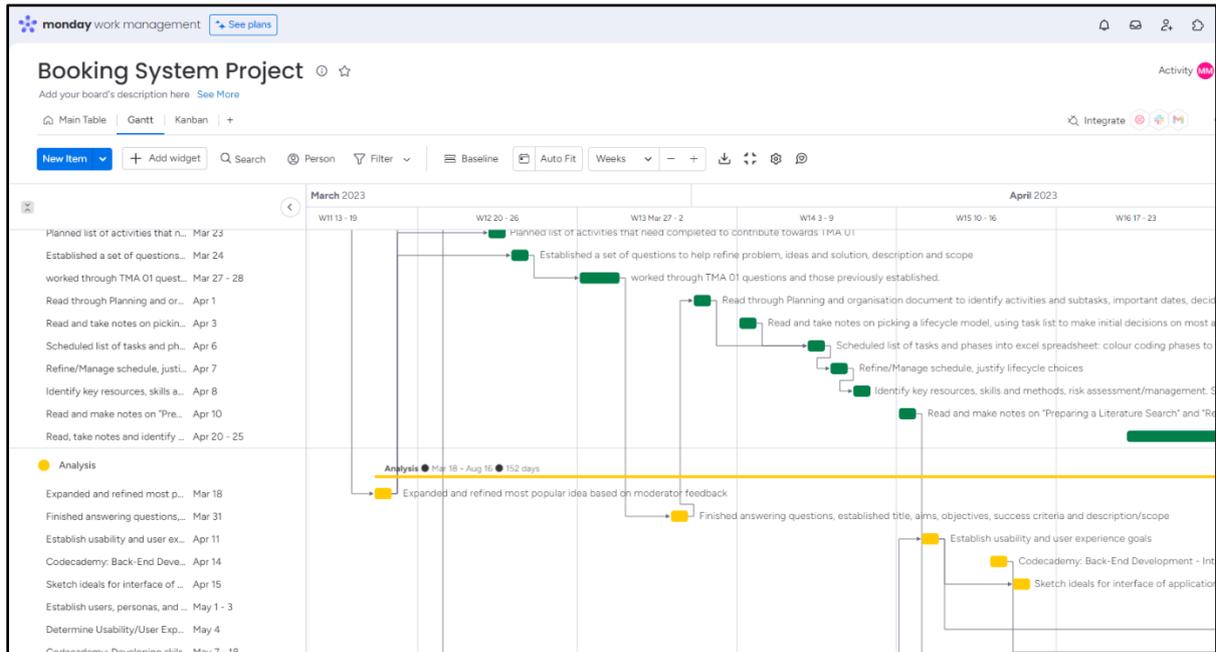
Project Management

Tasks & Subtask

- Analysis of user feedback from prototypes.
- Analysis of user feedback from sketches.
- Codecademy: Back-End Development - API Development with Swagger and OpenAPI.
- Codecademy: Back-End Development - Back-End and feature testing and Database Basics.
- Codecademy: Back-End Development - Introduction to Back-End and Build a Back end with Express.js.
- Codecademy: Back-End Development - User Authentication & Authorisation.
- Codecademy: Back-End Development - working with Databases.
- Codecademy: Back-End Development: Adding a Postgre SQL Database.
- Codecademy: Back-End Development: Designing Relational Databases and Advanced PostgreSQL.
- Codecademy: Full-Stack Development - Common Attacks on Web Applications.
- Codecademy: Full-Stack Development - Connecting Front-End to Back-End.
- Codecademy: Full-Stack Development - Deploying Web Application.
- Codecademy: Full-Stack Development - DevOps Fundamentals.
- Codecademy: Full-Stack Development - Fundamentals of Operating Systems.
- Codecademy: Full-Stack Development - Web Security Fundamentals and Data Security.
- Comprehend and summarise main points.
- Contact Tutor with questions based on feedback for evaluation of assignment.
- Decide on data gathering techniques for requirements elicitation (Interaction Design 4th Ed - Chapter 10 and 7).
- Decide on data gathering techniques to gain feedback from user on low fidelity prototypes.
- Design changes to implement.
- Designing database for use with booking system.
- Determine how each source addresses the needs of the project.
- Determine Keywords.
- Determine project contexts – Users, Activities, Environment.
- Determine Usability/User Experience goals as success criteria.
- Develop database and APIs.
- Establish a set of questions to help refine problem, ideas and solution, description and scope.
- Establish stakeholders.
- Establish usability and user experience goals.
- Establish users and personas.
- Evaluate current progress with client.
- Evaluate project and progress.
- Evaluation of user feedback.
- Expanded and refined most popular idea based on moderator feedback.
- Feedback from high-fidelity prototype and evaluation.
- Finish answering questions, established title, aims, objectives, success criteria and description/scope.
- Identify current skills.
- Identify key resources.
- Identify methods.

- Identify skills required.
- Identify sources of information.
- Implement web app, database and APIs.
- initial introduction email to tutor with refined idea.
- Initial prototype designs sketches based on research.
- Literature Search based on feedback from assignment and evaluation questions with tutor.
- Literature Search.
- Plan list of activities that need completing to contribute towards TMA 01.
- Plan next design and implementation steps.
- Post initial ideas to project preparation forums.
- Post variety of different ideas for feedback on project preparation forums.
- Read abstract, scan and skim, use PROMPT.
- Read and make notes on "Preparing a Literature Search" and "Reviewing Literature".
- Read and make notes on project choice document.
- Read and take notes on picking a lifecycle model, using task list to make initial decisions on most appropriate model (different parts of project will utilise different models).
- Read and take notes on TMA 01.
- Read more thoroughly and evaluate the literatures necessity.
- Read through Planning and organisation document to identify activities and subtasks, important dates, deciding which task fits into which phase.
- Read, take notes and identify Legal, Social, Ethical and Professional issues and Equality, Diversity and Inclusion. Answer "Asking the Right Questions" section.
- Redesign low fidelity prototypes of web app based on data, evaluation, analysis and design sheet.
- Refine boundaries and scope of project.
- Refine goals, aims, objectives and success criteria.
- Refine/Manage schedule, justify lifecycle choices.
- Research similar products already available.
- Review and feedback with tutor (TMA 01).
- Review and feedback with tutor (TMA 02).
- Review and feedback with tutor (TMA 03).
- Review and feedback with tutor.
- Review Current Progress.
- Risk assessment and management
- Scheduled list of tasks and phases into excel spreadsheet: colour coding phases to tasks and project parts.
- Set up Git.
- Set up GitHub.
- Sketch ideals for interface of application: Customer facing and admin console.
- Sketch initial ideas and designs for system layout.
- Test system for errors/bugs.
- TM352 - EMA Assignment work.
- Use LSEPIs and EDI notes to ensure you are following ethical principles and code of conduct.
- Use notes, previous module, study guide, project choice, sample project titles and project ideas documents to produce initial project idea.

- User involvement and data/requirements gathering using wireframes and data gathering techniques.
- User involvement for feedback using low fidelity prototypes and data gathering techniques.
- Web application development.
- What systems are already available and what features do they offer?
- Worked through TMA 01 questions and those previously established.
- Write up Final Report.
- Write up report TMA 01.
- Write up report TMA 03.



monday work management [See plans](#)

Booking System Project

Main Table | Gantt | Kanban | +

New Item Search Person Filter Sort Hide ...

20 Items Mar 18 - Aug 16

Research

Item	Person	Status	Timeline	Dependency
What systems are already available and what features do they offer?	MM	Done	Apr 9	initial introduction email to tutor with refined idea
Literature Search 7	MM	Done	Apr 12 - 13	Read and make notes on "Preparing a Literature Search" and ...

Subitem	Owner	Status	Date
Determine Keywords	MM	Done	
Identify sources of infor...	MM	Done	
Begin search	MM	Done	
Read abstract, scan and s...	MM	Done	
Comprehend and summa...	MM	Done	
Read more thoroughly an...	MM	Done	
Determine how each sou...	MM	Done	
+ Add subitem			

Literature Search based on feedback from assignment and evaluation questions with tutor	MM	Working o...	May 5 - 6	Review and feedback with tutor (TMA 01)
+ Add Item			Apr 9 - May 6	

monday work management [See plans](#)

Booking System Project

Main Table | Gantt | Kanban | +

New Item Search Person Filter Sort Hide ...

Phase	Items	Status	Timeline
Planning	16 Items	Done	Mar 4 - Apr 25
Analysis	20 Items	In Progress	Mar 18 - Aug 16
Research	3 Items / 7 Subitems	In Progress	Apr 9 - May 6
Design	9 Items	In Progress	Apr 13 - Aug 17
Development	5 Items	In Progress	May 27 - Jul 8
Testing	7 Items	In Progress	Jul 10 - Aug 20
Evaluation	16 Items	In Progress	Apr 29 - Aug 28
Implementation	6 Items	In Progress	Jul 13 - Aug 19
Report Writing	4 Items	In Progress	Apr 16 - Sep 10
Outside Project Work	2 Items	Done	Apr 25 - May 26

Domain Modelling

Problem Statement

Developing a web application booking system for a three venue boardgame café business.

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.

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.

Business Rules

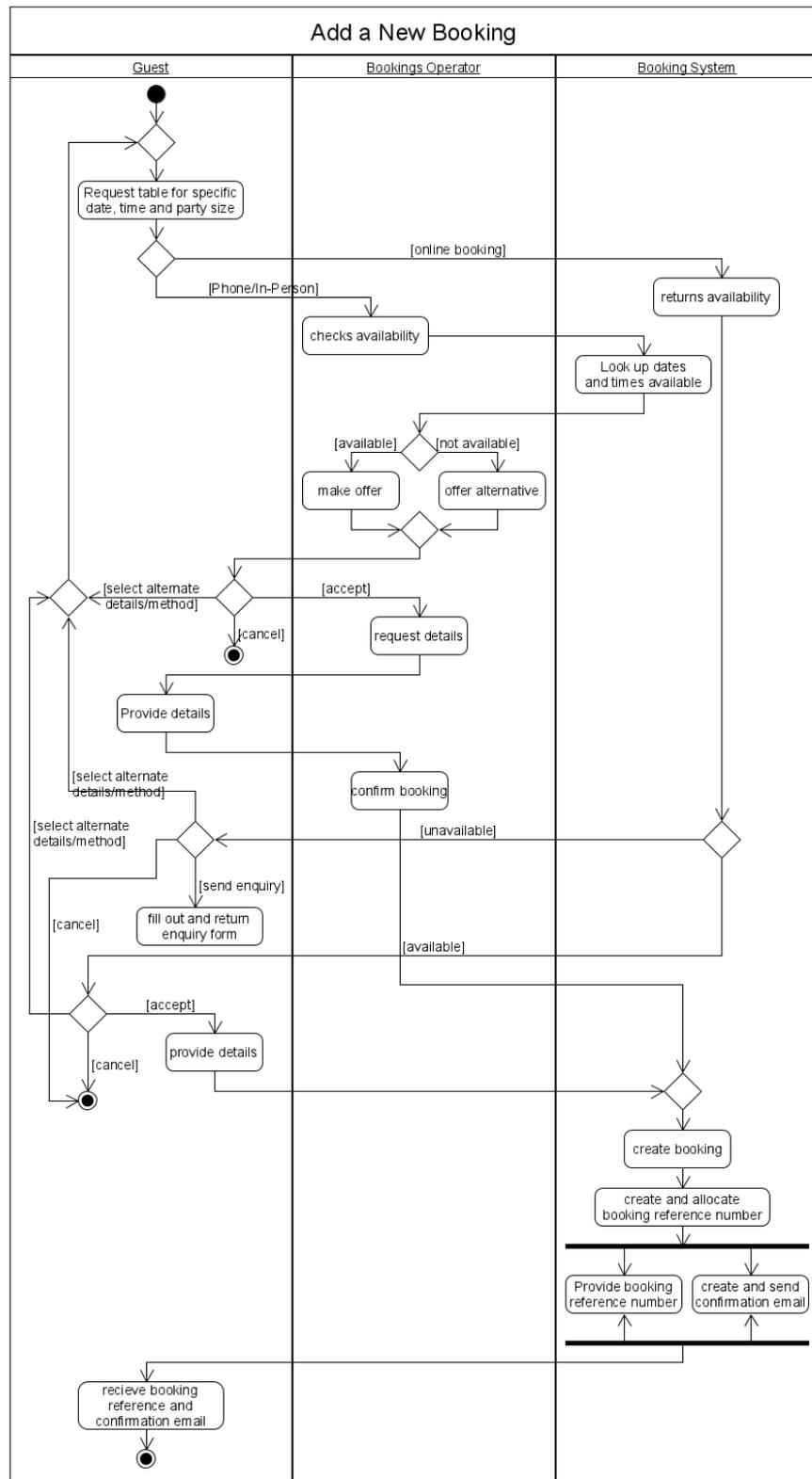
1. If a guest has a booking, they must arrive within 30 minutes, or the booking will be marked as a 'no-show' and the space becomes available for other guests.
2. 30 minutes before the guests are required to vacate the table, staff must inform the guest that they have 30 minutes left until their booking 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 24 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.
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.

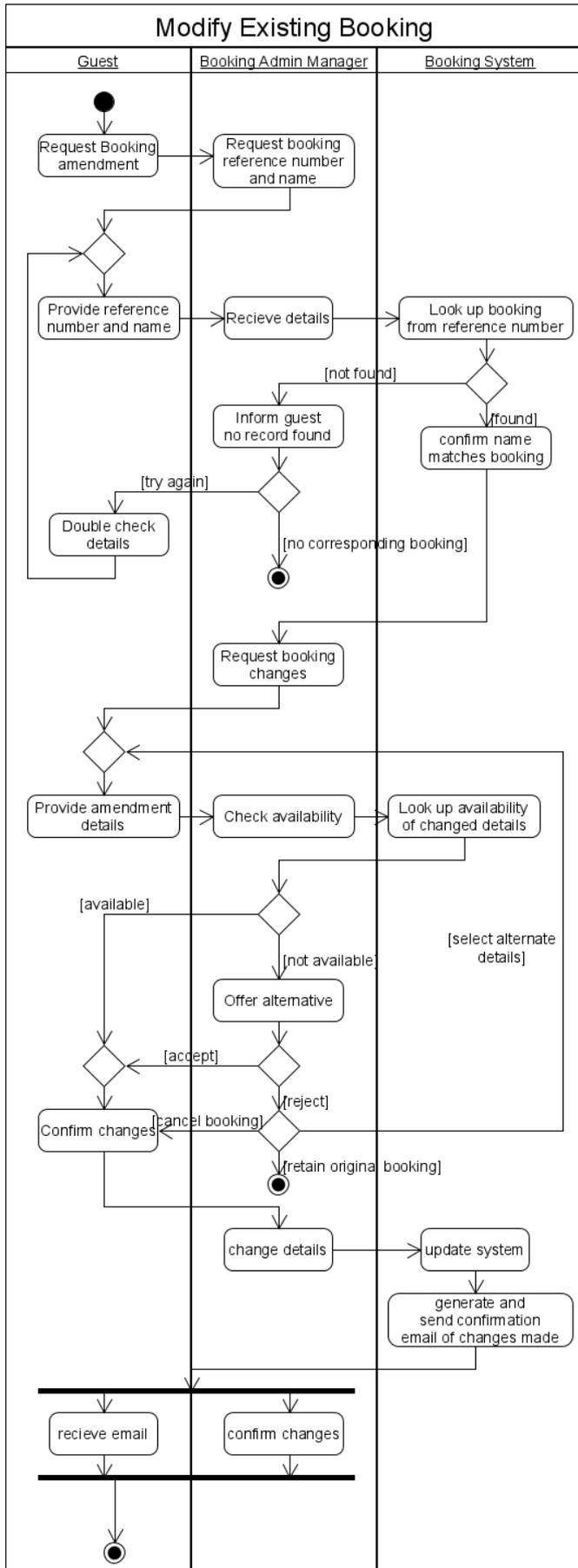
Business Processes

1. Add a new Booking:

The booking process being when a guest attempts to make a booking, reserving a table for a specified date, group size and period lasting no longer than the maximum length of time specified by the venue manager.

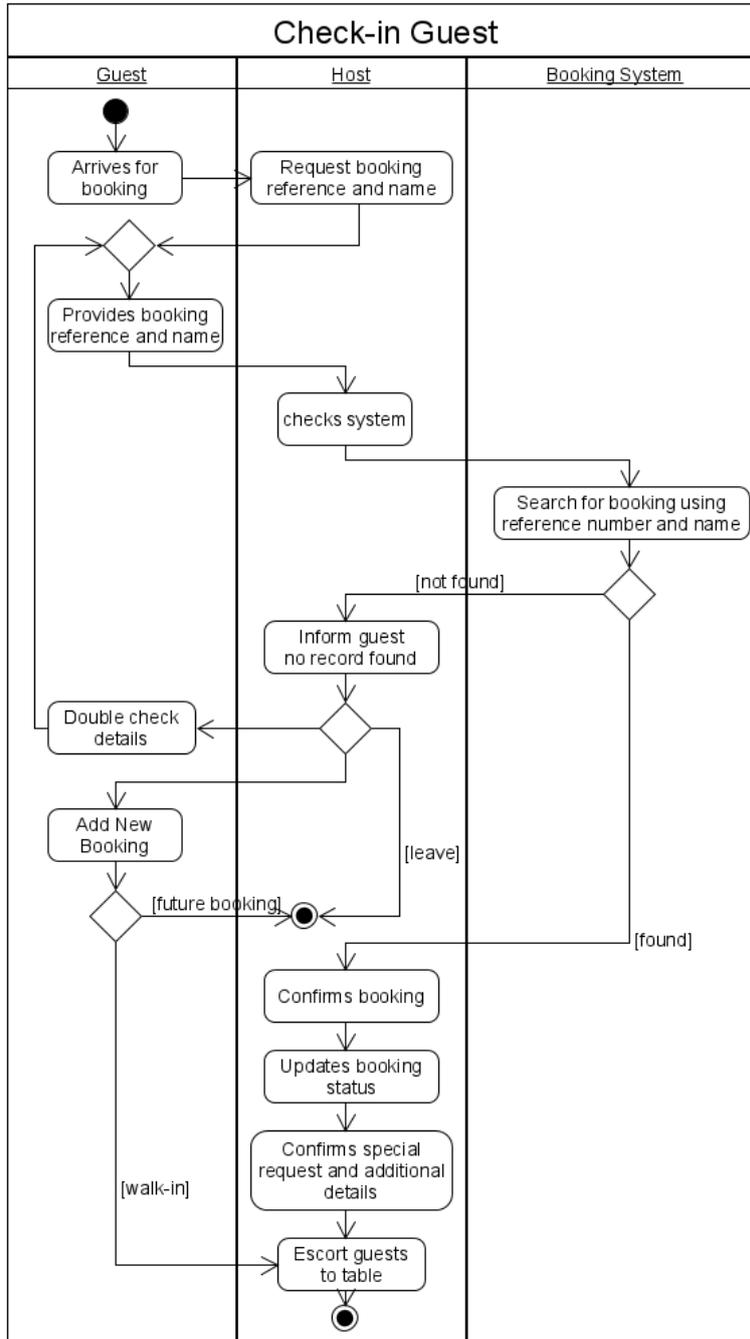
Each venue has a variety of sections with tables that seat a range of 2 – 8 people and tables can be pushed together to accommodate more. A price for the booking is offered based on the guest's party size and the use of the venues board game collection. If the guest accepts the price, they are required to provide additional details which the system will use to confirm the booking. The required details are as follows: Date, Start and end time, Full name, Party size, Table number, contact details (phone or email), Additional notes or special requests, Games to reserve.





2. Modify an existing booking: The modification process begin when a guest contacts the venue to amend an existing booking. The guest must provide their booking reference number and name to identify the correct booking before moving forward. The host will find the booking in the system, request the details which require amending and determine if the alteration can be made. If there are no conflicts, the host will make the changes, confirm the guest is happy and commit the new booking details in the system. If the changes conflict with other bookings, alternatives are provided to the guest who can confirm or reject until an agreement is made. They cancel the booking or keep the original booking details.

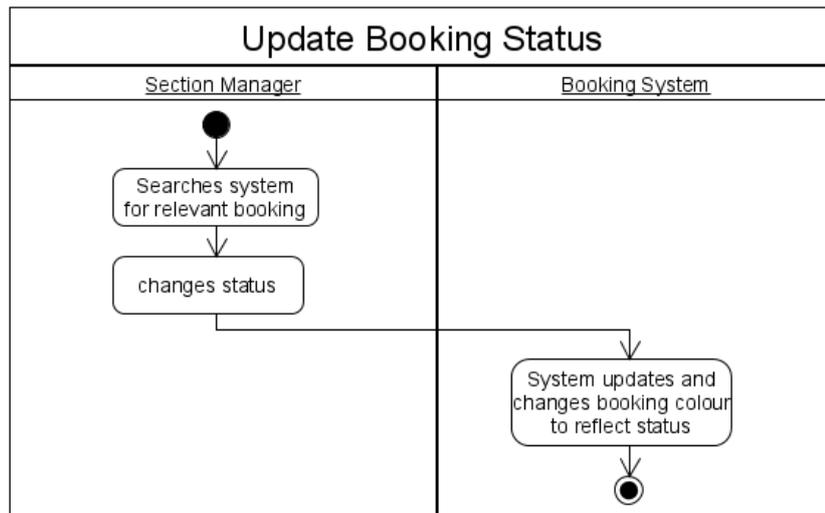
(Not modelled: future functionality - online booking modification: The guest makes amendments to their booking using their reference number and name, if there is availability, the system automatically makes the adjustments and sends a confirmation email, if there is no availability, they are presented with an enquiry form to fill out and submit or they can exit the page).



3. Check-in Guest: The check-in guest process begins when a guest wants to make an amendment to their booking. The guest provides the host or system with their booking reference number so it can be confirmed the guest has a valid booking. If the system returns no booking, the guest can start the new booking process for immediate or future reservations, try again with different details or cancel the process and leave. If the system finds the associated booking, the host will confirm the details, update its status, and escort the guest/s to their table.

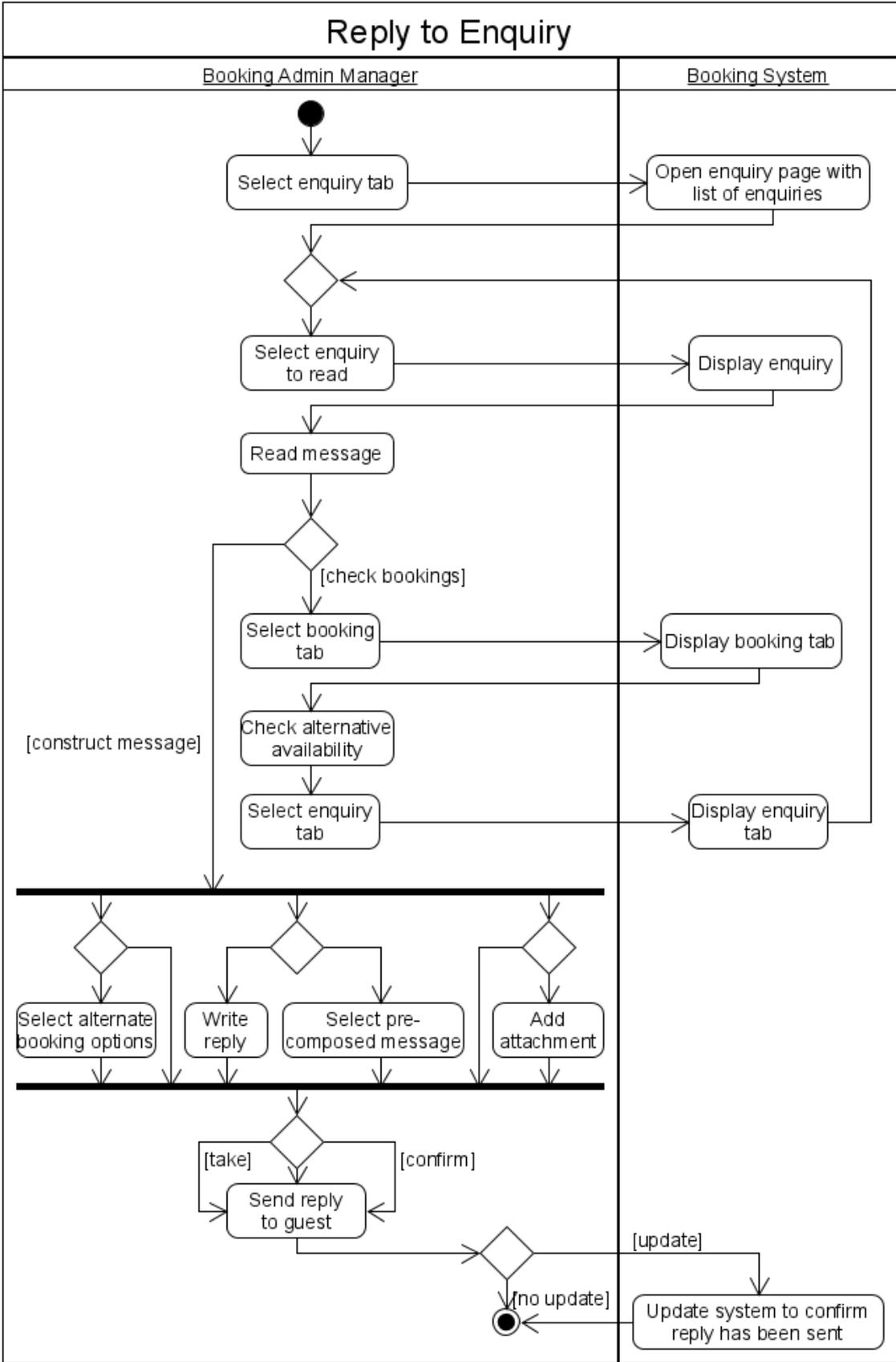
4. Update Booking Status:

The update booking status process begins when the booking reaches a specific point in its lifetime and changes in the following order: Arriving soon -> Check-in -> 30 Minute warning -> 15 Minute warning & payment -> Cleared. There must already be a valid booking in the system. At specific times during a booking's lifetime, the status must



be changed by the section manager based on the time left until the end of the booking. This will update its status and colour to reflect the changes. If a guest doesn't arrive within a specified time of their bookings check-in time, it will be marked as a 'no-show', removed from the display and be dealt with at the end of the day by a booking admin manager.

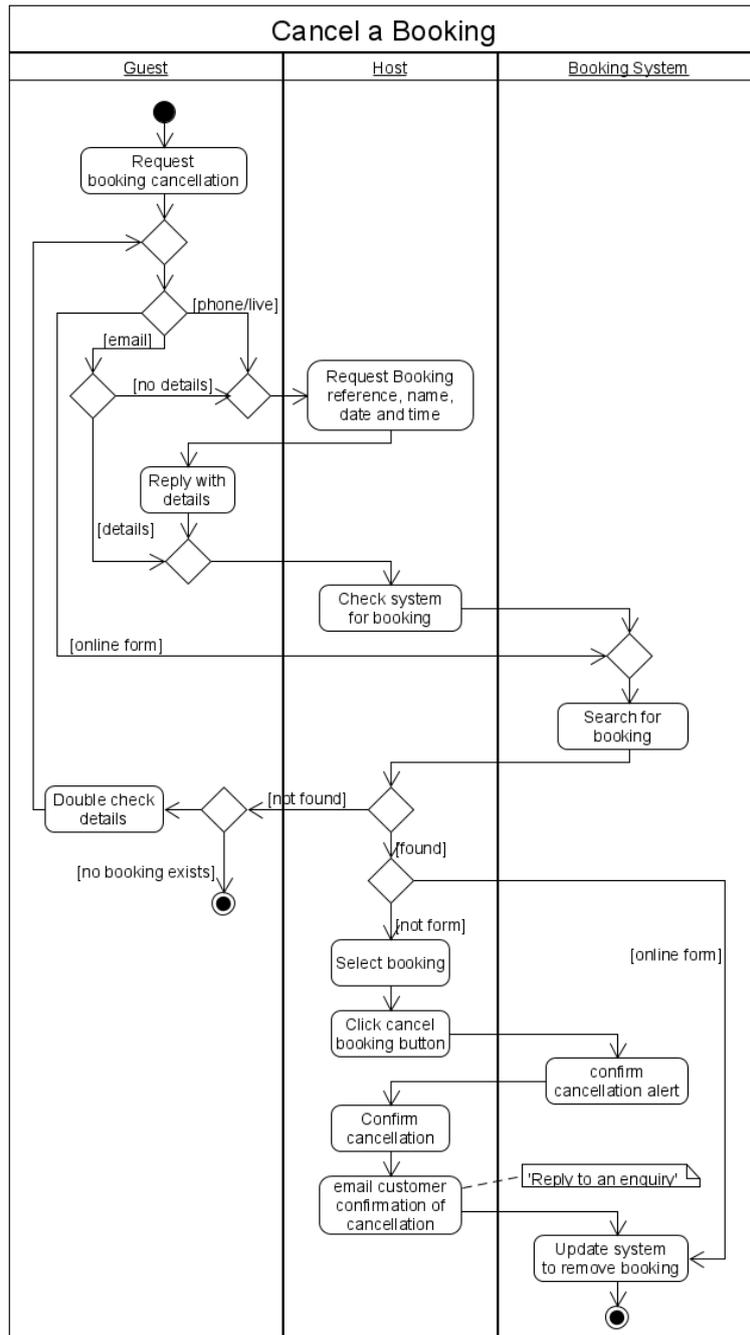
5. Reply to an enquiry: The reply to enquiry process is initiated by the booking admin manager. All emails are managed through the system which formats them and displays them in the booking system as enquiries. If a guest attempts to make an online booking and the selected date and time has no availability, the guest completes the offered enquiry form which is forwarded to the booking system. All future communication is conducted through the guest's email address and the booking systems enquiry tab. To view all enquiries, the booking admin manager selects the enquiries views tab whereupon they can then select an individual enquiry from the list to view the message contents, history, details, special requests, and 'reply to' options. The booking admin manager can select precomposed replies, alternative booking options, add attachments, then 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').



6. Cancel a booking: The cancel booking process is initiated by the guest and the system must have a pre-existing booking saved in the system. The guest will provide their booking reference number, name and contact details to confirm their identity and to locate the correct booking. There are a number of ways a cancellation can be received by email:

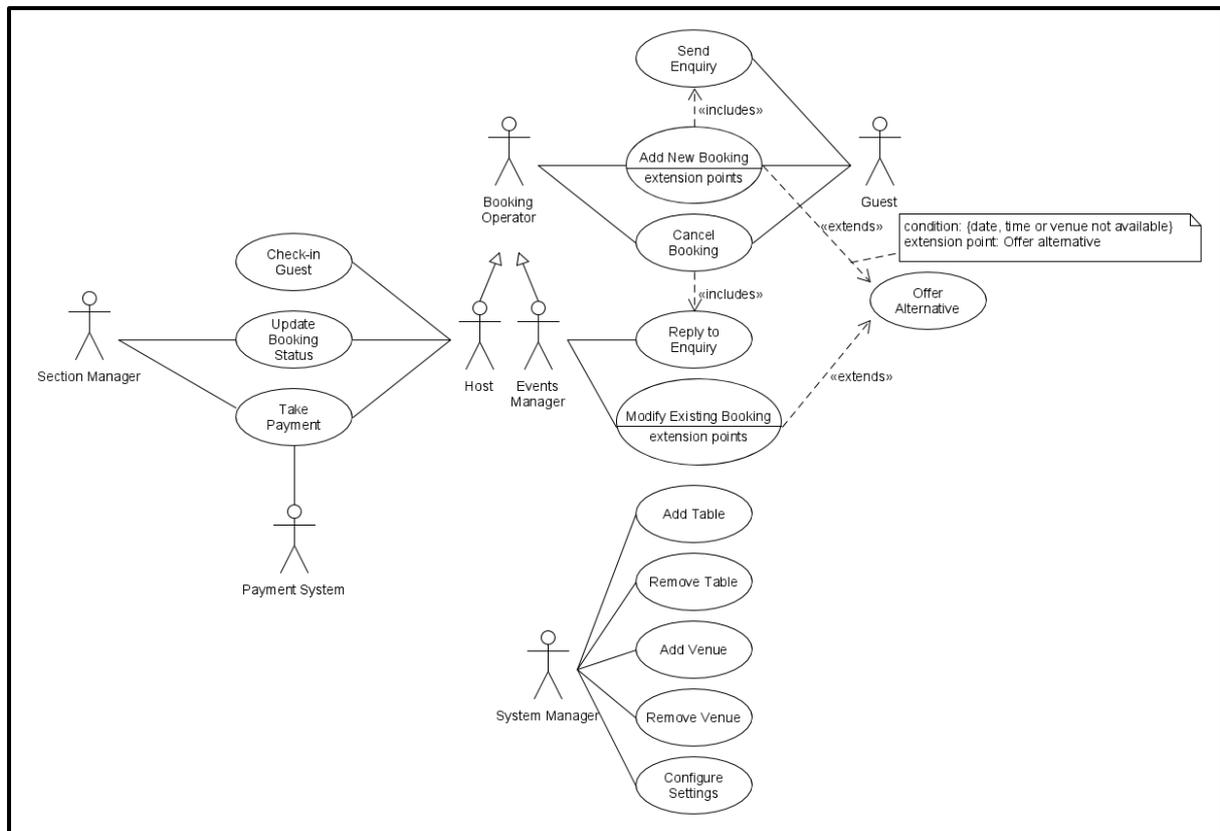
The guest can reply via their confirmation email that already includes all the necessary details. The guest can submit an online form that requires the necessary details before it can be sent. The guest can send an email directly to the businesses contact email address in which case, it is forwarded to the booking system so the relevant staff member can reply to the guest requesting them to provide the necessary details in a follow up email.

The booking can then be identified, selected and cancelled by clicking on the cancel booking button. The system will send a prompt to confirm cancellation before being removed from the system and any necessary views.



Requirements

Use Case Models



Elaborated Use Cases

Identifier and name: UC1 Add a new online 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 guest is using a web browser.

Main success scenario:

1. The guest chooses to make a reservation.
2. The guest selects the venue, dates, times and party size.
3. The booking system provides the availability of the request.
4. The guest decides to proceed with the offer.
5. The guest provides their full name, contact details, games to reserve, section, and special requests.
6. The booking system creates a booking and a unique booking reference number.
7. The booking system provide the booking reference number to the guest.

8. The booking system creates a confirmation of the booking and sends it to the guest.

Extensions

3.a: A table is not available, and an enquiry form is offered.

3.a.1: The booking system offers the guest an enquiry form to request alternative dates

3.a.2: The guest provides details (as per main scenario part 5) and requests a reply from staff at the venue with alternative booking options.

3.b: A table is not available and starts the process again using different details.

3.c: A table is not available and the guest calls or attends the venue in person to make a booking.

3.d: A table is not available and the guest cancels the booking process.

4.a The guest declines to proceed with the offer.

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.

Identifier and name: UC3 *modify existing booking (phone/in-person)*

Initiator: *Guest*

Goal: The details of a guests booking are amended

Precondition: There is an existing booking in the system with a valid unique booking reference number.

Postcondition: The existing bookings details have been changed to reflect the guest's requested amendments.

Assumptions: The environment allows for a conversation at a normal volume.

Notes: The system displays all of the bookings for the matching calendar day within the main bookings view, centring and highlighting the found booking (details of how the system functions are to be part of the solution later – this note is just a reminder for now – please remove at a later date).

Main success scenario:

1. The guest chooses to make an amendment to their booking.
2. The host requests the guests booking reference number and name.
3. The guest provides the necessary details.
4. The host searches the system using the provided details for the correct booking.
5. The system confirms the booking exists.
6. The system displays the booking.
7. The host confirms the date, time and party size details of the booking with the guest.
8. The guest confirms the details.
9. The host requests amendments to be made.
10. The guest provides amendments to be made.
11. The host looks up availability within the scope of provided details and commits the necessary changes to the booking.
12. The system accepts the changes and updates the booking.
13. The system generates a confirmation email of changes and sends it to the guest.

Extensions:

5.a The staff member informs the guest there is no booking in the system with the provided details and the guest chooses not to continue with the modification process.

5.b The staff member informs the guest there is no booking returned with the provided details and requests the information again to double check the correct details have been provided.

5.b.1 The guest provides an alternative booking reference number.

5.b.2 The staff member searches the system using the new provided details.

5.b.3.1 The system confirms the booking exists.

5.b.3.2 The system cannot find a booking matching the new details: return to step 5.a.2

5..b.3.3 The system cannot find a booking matching the new details: Exit search

11.a Requested changes provide no availability withing the parameters of the given amendments.

11.a.1 The guest accepts an alternative offer of a combination of different venues, dates and times.

11.a.2 The guest does not accept the initial alternative and requests another alternative options.

11.a.3 The guest does not continue with the modification process.

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.

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 manager confirms the change.
 6. The booking system updates the display to reflect the changes made.
-

Identifier and Name: UC6 *Reply to enquiry*

Initiator: Booking admin manager

Goal: Send a reply to a guests booking enquiry

Precondition: None

Postcondition: An email has been successfully delivered to the guest

Assumptions: The enquiry is to request discussion of booking an event, alternative options as original booking details were unavailable, or to cancel a booking.

Main success scenario:

1. The Booking admin manager selects the enquiry tab to open the enquiry page.
2. The Booking admin manager selects an enquiry from the list of those available.
3. The Booking admin manager reads the enquiry.
4. The Booking admin manager writes a reply.
5. The Booking admin manager sends the message.
6. The system updates to confirm a reply has been sent.

Extensions:

3.a The Booking admin manager toggles between the enquiry view and main booking view to check for alternative availability.

4.a The Booking admin manager selects alternative booking options to send as part of the reply.

4.b The Booking admin manager selects a precomposed message as opposed to writing a custom reply.

4.c The Booking admin manager adds an attachment to the reply.

5.a The Booking admin manager selects the 'Take' option to add a provisional booking component to the main booking grid view. This will remain for a length of time specified by the manager until it is automatically removed.

5.b The Booking admin manager select the 'confirm' option to permanently add a booking component to the main booking grid view.

Identifier and name: UC7 *Cancel a booking*

Initiator: Guest

Goal: The booking is removed from the system

Precondition: There is a valid booking in the system.

Postcondition: The booking has been removed from the system and the guest sent a confirmation email.

Main success scenario:

1. The guest request a cancellation of their booking.

2. The staff member requests the guests booking reference number, name, date and start time.
3. The guest provides the necessary details.
4. The staff member searches the system for the corresponding booking and confirms the provided details match those stored on the system.
5. The staff member selects the booking, confirms the cancellation with the guest and proceeds to remove the booking from the system.
6. The booking system provides an alert to confirm the removal of the booking.
7. The staff member confirms, and the system is updated to reflect the removal of the booking.

Extensions:

- 1.a** The guest requests their booking cancellation through email with all the provide booking information.
- 1.b** The guest requests their booking cancellation through email with no booking information.
- 1.c** The guest requests their booking cancellation through a web form.
- 4.a** The booking cannot be found, and the guest provides alternate details to search again.
- 4.b** The booking cannot be found and the guest does not continue with the process.
- 4.c** The guest provides all the correct details in the form and the system removes the booking from the system.

Functional Requirements

<i>Requirements Number</i>	<i>FR1</i>
<i>Event/Use Case</i>	UC1, Step 2
<i>Description</i>	The system shall obtain a venue name from the user
<i>Rational</i>	Associates guests booking to a specific venue
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid venue name shall be accepted
<i>Customers dis/satisfaction</i>	Neutral
<i>Priority</i>	Important
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023: FR Created

<i>Requirements Number</i>	<i>FR2</i>
<i>Event/Use Case</i>	UC1, Step 2
<i>Description</i>	The system shall obtain a date from the user
<i>Rational</i>	Associates guests booking to a specific date
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid date shall be accepted
<i>Customers dis/satisfaction</i>	Neutral
<i>Priority</i>	Important
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023: FR Created

<i>Requirements Number</i>	<i>FR3</i>
<i>Event/Use Case</i>	UC1: Step 2
<i>Description</i>	The system shall obtain a time from the user
<i>Rational</i>	Associates a guest booking to a specific time
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid time shall be accepted
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR4</i>
<i>Event/Use Case</i>	UC1: Step 2
<i>Description</i>	The system shall obtain a party size from the user
<i>Rational</i>	Associates a guest's party size to a specific booking
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid party size shall be accepted
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR5</i>
<i>Event/Use Case</i>	UC1: Step 3
<i>Description</i>	The system shall check the availability of tables for the given party size, time, date and venue. An offer shall be made.
<i>Rational</i>	Ensures double booking and overbooking prevention
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	Offers are consistent and accurate with availability and within request parameters
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR6</i>
<i>Event/Use Case</i>	UC1: Step 3.1.a
<i>Description</i>	<i>No Availability:</i> If a table is not available, the system shall accept an enquiry form, filled out by the guest.
<i>Rational</i>	System users can determine alternative availability based on enquiry parameters.
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid enquiry shall be accepted
<i>Customers dis/satisfaction</i>	3 (Neutral)
<i>Priority</i>	4 (Somewhat Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR7</i>
<i>Event/Use Case</i>	UC1: Step 4.a
<i>Description</i>	The system shall accept confirmation from the user
<i>Rational</i>	A guest may change their mind and decide to cancel their request
<i>Originator</i>	Business Process: Add a new Booking
<i>Fit Criterion</i>	A response indicating cancellation confirmation shall be accepted
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR8</i>
<i>Event/Use Case</i>	UC1: Step 5
<i>Description</i>	The system shall obtain a full name from the user
<i>Rational</i>	Allocates and identifies a specific booking by the guest's name
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid name shall be accepted
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR9</i>
<i>Event/Use Case</i>	UC1: Step 5
<i>Description</i>	The system shall obtain an email address from the user
<i>Rational</i>	Associates the guest's email address with a specific booking
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid email address shall be accepted
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR10</i>
<i>Event/Use Case</i>	UC1: Step 5
<i>Description</i>	The system shall obtain a phone number from the user
<i>Rational</i>	Associates the guest's phone number with a specific booking
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid phone number shall be accepted
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR11</i>
<i>Event/Use Case</i>	UC1: Step 5
<i>Description</i>	The system shall obtain which board games to reserve
<i>Rational</i>	Associates a number of board games to set aside with a specific booking.
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A valid selection of board games shall be accepted
<i>Customers dis/satisfaction</i>	3 (Neutral)
<i>Priority</i>	4 (Somewhat Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR12</i>
<i>Event/Use Case</i>	UC1: Step 5
<i>Description</i>	The system shall obtain any special requests
<i>Rational</i>	Associates special requests with a specific booking
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	Valid special requests shall be accepted
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR13</i>
<i>Event/Use Case</i>	Add a new section
<i>Description</i>	Adds a new section at the specified venue
<i>Rational</i>	Provides the ability to divide up the venue by area and delegate responsibilities to area managers
<i>Originator</i>	Developer
<i>Fit Criterion</i>	A valid section shall be accepted
<i>Customers dis/satisfaction</i>	3 (Neutral)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Structural Model
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR14</i>
<i>Event/Use Case</i>	UC1: Step 6
<i>Description</i>	The system shall create a booking from the provided parameters
<i>Rational</i>	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.
<i>Originator</i>	Interviewee
<i>Fit Criterion</i>	A reservation will be created in the system for a given period on the provided date.
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	15.07.2023 FR Created

<i>Requirements Number</i>	<i>FR15</i>
<i>Event/Use Case</i>	UC1: Step 6
<i>Description</i>	The system shall create a unique booking reference number
<i>Rational</i>	A unique number allocated to the booking provides identification
<i>Originator</i>	Use Case 1
<i>Fit Criterion</i>	A unique number shall be assigned to the booking
<i>Customers dis/satisfaction</i>	3 (Neutral)
<i>Priority</i>	3 (Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	17.05.2023 FR Created

<i>Requirements Number</i>	<i>FR16</i>
<i>Event/Use Case</i>	UC2: Step 7
<i>Description</i>	The system shall accept acknowledgement of the use of internal board games.
<i>Rational</i>	The venue can produce an accurate bill for the session
<i>Originator</i>	Developer
<i>Fit Criterion</i>	A valid acknowledgement is accepted by the system
<i>Customers dis/satisfaction</i>	3 (Neutral)
<i>Priority</i>	2 (Fairly Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i>	<i>FR17</i>
<i>Event/Use Case</i>	UC1: Step 8
<i>Description</i>	The system distributes a predefined confirmation of the bookings details to the guest.
<i>Rational</i>	The confirmation provides evidence the booking has been accepted by the system and its detailed contents can provide booking identification.
<i>Originator</i>	Interviewee/Developer
<i>Fit Criterion</i>	A confirmation shall be received by the guest
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	3 (Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 1
<i>History</i>	17.05.2023 FR Created

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

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

<i>Requirements Number</i>	<i>FR20</i>
<i>Event/Use Case</i>	UC3: Step 6
<i>Description</i>	The system will update the bookings tab view to display all bookings on the same date
<i>Rational</i>	Any modifications requests availability can be seen in a single view
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The bookings tab view will show all bookings on a specific date
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Use Case 3
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i>	<i>FR21</i>
<i>Event/Use Case</i>	Add a new venue
<i>Description</i>	A new venue will be added to the system
<i>Rational</i>	Venue tables and bookings can be managed by individual venues
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The system will accept a valid new venue
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Structural Model
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i>	<i>FR22</i>
<i>Event/Use Case</i>	Add a new table
<i>Description</i>	Adds a new table at a specific venue
<i>Rational</i>	Allows individual tables to be managed at each venue
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The system will accept a valid new table
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Structural Model
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i>	<i>FR23</i>
<i>Event/Use Case</i>	Remove a table
<i>Description</i>	Removes a table at a specific venue
<i>Rational</i>	Allows individual tables to be managed at each venue
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The system will remove the associated table
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	2 (Fairly Important)
<i>Conflicts</i>	N/A
<i>Supporting materials</i>	Structural Model
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i>	<i>FR24</i>
<i>Event/Use Case</i>	Mark table as out of service
<i>Description</i>	Marks a table as not being available at the specified venue
<i>Rational</i>	A table might be damaged and not safe for guest use so should be made unavailable.
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The table will not be available to book
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	3 (Important)
<i>Conflicts</i>	FR22: table needs to exist before being able to be marked as out of service
<i>Supporting materials</i>	FR22
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i>	<i>FR25</i>
<i>Event/Use Case</i>	UC2: Step 3
<i>Description</i>	The system shall accept a table from the user
<i>Rational</i>	Associates a table with a specific reservation
<i>Originator</i>	Developer
<i>Fit Criterion</i>	An available table shall be accepted by the system
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	FR22: table needs to exist and be available and in the corresponding section
<i>Supporting materials</i>	FR22 , FR13
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i>	<i>FR26</i>
<i>Event/Use Case</i>	UC4: Step 6
<i>Description</i>	The system will update the reminder status
<i>Rational</i>	Updates a reminder for a specific booking
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The status matches a specific time period of the booking
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	2(Fairly Important)
<i>Conflicts</i>	FR18: A booking must exist for it to be updated
<i>Supporting materials</i>	Use Case 4
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i> FR27	
<i>Event/Use Case</i>	UC4: Step 6
<i>Description</i>	The system will update the reminder colour
<i>Rational</i>	Updates a reminder for a specific booking
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The colour matches a specific time period of the booking
<i>Customers dis/satisfaction</i>	2 (Satisfied)
<i>Priority</i>	2 (Fairly Important)
<i>Conflicts</i>	FR18: A booking must exist for it to be updated
<i>Supporting materials</i>	Use Case 4
<i>History</i>	18.07.2023 FR Created

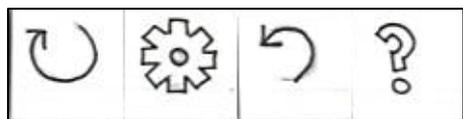
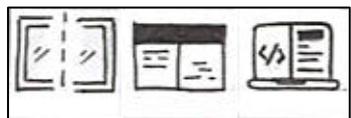
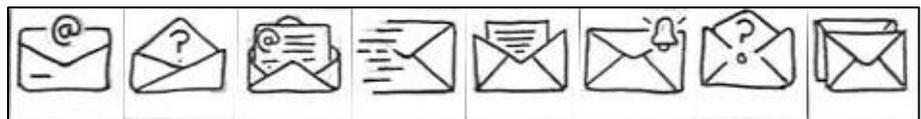
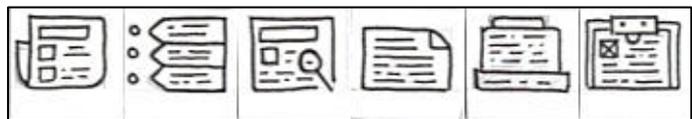
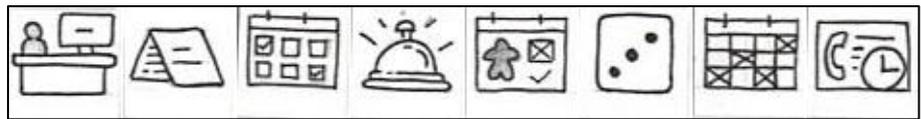
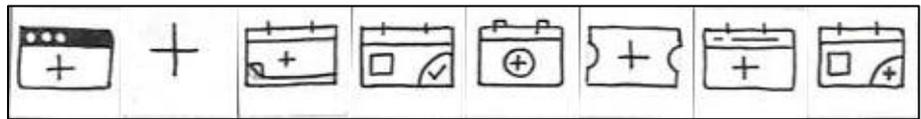
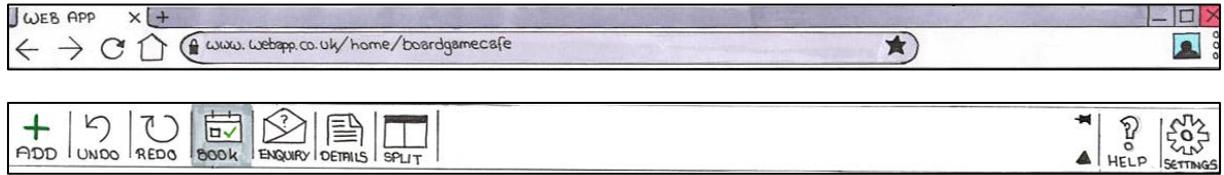
<i>Requirements Number</i> FR28	
<i>Event/Use Case</i>	UC4: Step 6
<i>Description</i>	The system shall update the bookings status
<i>Rational</i>	A bookings status and colour is modified to match a specific point in its lifetime.
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The reminder status and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

<i>Requirements Number</i> FR29	
<i>Event/Use Case</i>	Remove an existing venue
<i>Description</i>	An existing venue will be removed from the system
<i>Rational</i>	If a venue has to be shut down for any reason (refurbishment, demolition, out of business, etc...) it needs to be able to be removed from the system
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The system will no longer hold a record of the venue being removed
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	A venue must exist for it to be removed
<i>Supporting materials</i>	Structural Model
<i>History</i>	18.07.2023 FR Created

<i>Requirements Number</i>	<i>FR30</i>
<i>Event/Use Case</i>	Remove an existing section
<i>Description</i>	An existing section will be removed from the system
<i>Rational</i>	Sections may be combined into new sections and the system therefor requires the ability to remove them
<i>Originator</i>	Developer
<i>Fit Criterion</i>	The system will no longer hold a record of the section
<i>Customers dis/satisfaction</i>	1 (Very Satisfied)
<i>Priority</i>	1 (Extremely Important)
<i>Conflicts</i>	A section must exist for it to be removed
<i>Supporting materials</i>	Structural Model
<i>History</i>	18.07.2023 FR Created

Design

Prototypes and Component Parts



A. Hale	15	A. Hale	15
D. Bird	30	D. Bird	30
O. Webb	0	O. Webb	0
A. Wilkinson	0	A. Wilkinson	0
A. Wilkinson	15	A. Wilkinson	15
O. Webb	0	O. Webb	0
A. Wilkinson	15	A. Wilkinson	15
P. Melendez	-30	P. Melendez	-30

VENUE: SECTION ◀ SATURDAY 24TH JUNE 2023 ▶ LOCK

TAB	CAP	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
ENQ ▶																					
1	2																				
2	2																				
3	4																				
4	6																				
5	6																				
6	6																				
7	4																				
8	4																				
9	2																				
10	2																				
11	2																				
12	2																				
13	4																				
14	4																				
15	8																				
16	10																				
17	4																				
18	4																				
19	4																				
20	4																				
21	4																				
22	6																				
23	6																				
24	2																				
25	2																				
26	2																				
27	2																				

KEY AND RULES

Greater than 30 minutes until arrival

Arriving soon: 30 minutes or less until arrival

Check-in: Green after user checked-in. Reminder will appear when there is 30 minutes until a tables booking is finished

M. MASON 30

Warning: 15 minutes until payment taken

Payment: Payment has been taken and the table is waiting to be cleared

Cleared: The table has been cleared.

Reminders are always sorted by the least amount of time remaining at the top of the stack

ADD BOOKING

DATE
 SATURDAY 24TH JUNE ▼

TIME
 12:00 15:00

GUESTS 2 ▼

TABLE 2 ▼

NAME
 First name
 last name

PHONE
 +44

EMAIL

WALK-IN

NOTES

TYPE
 GENERAL ▼

GAMES
 T.I.M.E STORIES

COMPLETE

<input checked="" type="checkbox"/> INBOX	2
<input type="checkbox"/> DRAFTS	1
<input type="checkbox"/> SENT	2
<input type="checkbox"/> TRASH	
<input type="checkbox"/> 10:00	
^ UNREAD	
<input checked="" type="checkbox"/> V. Parker 10:00	
☆ Waterloo	
<input type="checkbox"/> C. Higgins 09:26	
☆ Waterloo	
^ READ	
<input type="checkbox"/> K. Schneider 15/8	
☆ Waterloo	
<input type="checkbox"/> L. Willis 10/8	
☆ Waterloo	
<input type="checkbox"/> L. Clarke 1/8	
☆ Shoreditch	
<input type="checkbox"/> C. Holland 30/7	
☆ Camden	
<input type="checkbox"/> F. Smart 26/7	
☆ Waterloo	
<input type="checkbox"/> D. Kinsman 20/7	
☆ Waterloo	
<input type="checkbox"/> C. Coleman 12/6	
☆ Waterloo	
<input type="checkbox"/> H. Harrett 10/6	
☆ Shoreditch	

<input checked="" type="checkbox"/> INBOX	3
<input type="checkbox"/> DRAFTS	1
<input type="checkbox"/> SENT	2
<input type="checkbox"/> TRASH	

Bookings

Bookings

ARRIVING SOON

CHECK-IN

REMINDERS

	A. Wilkinson	P. Garrett	F. Watkins
	T. Walker	T. Walker	P. Melendez
	F. Watkins	O. Webb	A. Wilkinson
	C. Huff	A. Carter	A. Hale
	A. Carter	A. Wilkinson	O. Webb
	A. Hale	C. Huff	D. Bird
D. Bird	P. Garrett		D. Sharpe
A. Wilkinson	P. Melendez		
D. Sharpe	O. Webb		

A. Hale	P. Garrett	D. Bird
F. Watkins	D. Sharpe	A. Carter
A. Wilkinson	C. Huff	
A. Wilkinson	P. Melendez	
T. Walker	O. Webb	

ALL | HOME

FROM: Lorem Ipsum
 EMAIL: Lorem@Ipsum.co.uk
 PHONE: +44 7464 592 359
 PARTY SIZE: 4
 DATE: 24.06.2023 TIME: 15:00 LENGTH: 3 hours

Re: Lorem Ipsum June, 10, 2023, 14:25PM
 Lorem Ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex commodo consequat. Duis aute irure dolor in reprehenderit in volupteta velit esse cillum dolore eu fugiat nulla pariatur.
Lorem Ipsum

▶ Boardgame@waterloo.co.uk June, 09, 2023, 16:30PM
 Neque porre quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora, incididunt ut labore et dolore magnam aliquam quareat voluptatem. Excepteur sint occaecat cupidatat non, sunt in culpa qui officia deserunt mollit anim id est laborum
 Lorem Ipsum

Special Requests and notes

Lorem Ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex commodo consequat. Duis aute irure dolor in reprehenderit in volupteta velit esse cillum dolore

REPLY

VENUE	Waterloo:001 ▼
DATE	24.06.2023 ▼
TIME	14:00 ▼
TABLE	7 COMBINE

TAB	SIZE	COMBINE	SECTION
1	2		FOH
2	2		FOH
3	4		FOH
4	6		FOH
5	6		FOH
6	6		FOH
7	4		FOH

MESSAGE	Standard reply ▼
ATTACH	0

▶ Boardgame@waterloo.co.uk

Lorem Ipsum

We're sorry we have no tables available at your requested time and date. The closest alternative booking we can currently offer is:

Venue: {VENUE}

Date: {DATE}

Time: {TIME}

Table: {TABLE}

We will hold the booking for 3 days from today. If you wish to accept or request alternatives please reply to this email.

Boardgame Team

Offer

Confirm

SEND

ALL
HOME

Waterloo:001

SEARCH

FROM: Vanessa Parker
EMAIL: VanessaParker1986@gmail.co.uk
PHONE: +44 7464 592 359
PARTY SIZE: 4
DATE: 24.06.2023 TIME: 15:00 LENGTH: 3 hours

Lorem Ipsum.

June, 10, 2023, 14:25PM

Dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

V. Parker

REPLY: Boardgame@waterloo.co.uk
MESSAGE: Standard reply @

Vanessa Parker

We're sorry we have no tables available at your requested time and date. The closest alternative booking we can currently offer is:

{VENUE}, {DATE}, {TIME}, {TABLE}

We will hold the booking for 3 days from today. If you wish to accept or request alternatives please reply to this email.
Boardgame Team

VENUE: Waterloo:001
DATE: 24.06.2023
TIME: 14:00
TABLE: 7 COMBINE

TAB	SIZE	COMBINE	SECTION
1	2		FOH
2	2		FOH
3	4		FOH
4	6		FOH
5	6		FOH
6	6		FOH

Offer Confirm

SEND

Name: _____ Position: Dev Experience: 1

T1. A:2, B:7, C:3, D:3, E:3

T2. A: New, B: Undo, C: Redo, D: Reservation, E: Enquiry
F: Details, G: Split H: Help I: Settings

T3. A , B

T4.

i. Easy to navigate and see all times and tables

ii. Too small or too busy/cramped

iii. Natural progression from left-to-right

iv. Controls for zooming in and out on the ribbon and grid

T5. 33 points (perfect score + 10)

i. individual movement of time and tables

ii. C and d v. Name, party size, icons representing extras

iii. a and c vi. N/A

iv. Both vii. N/A

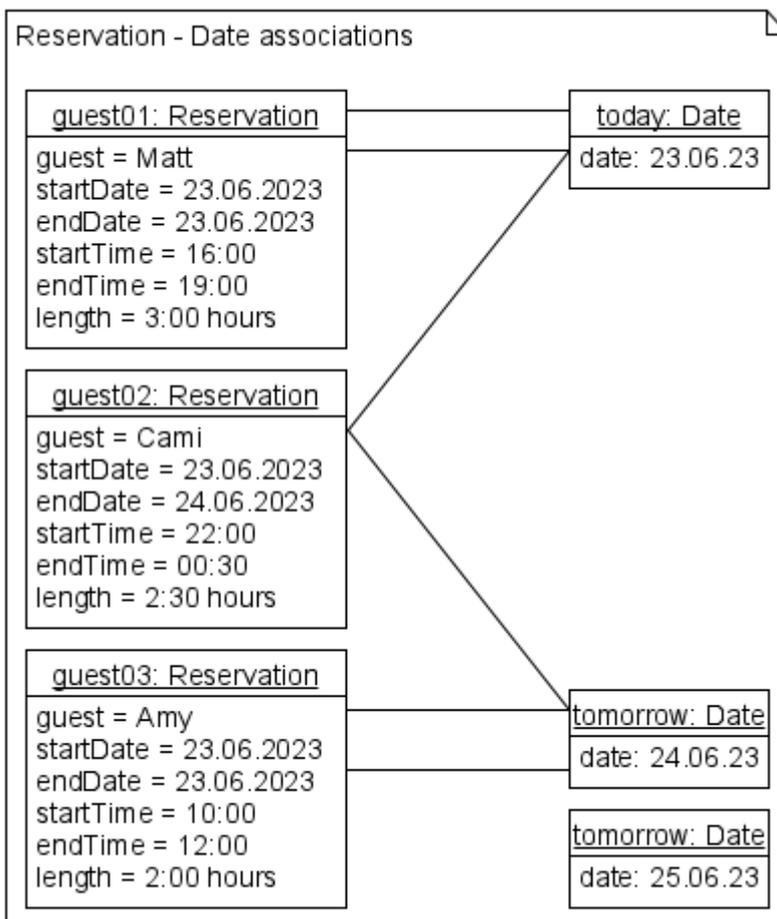
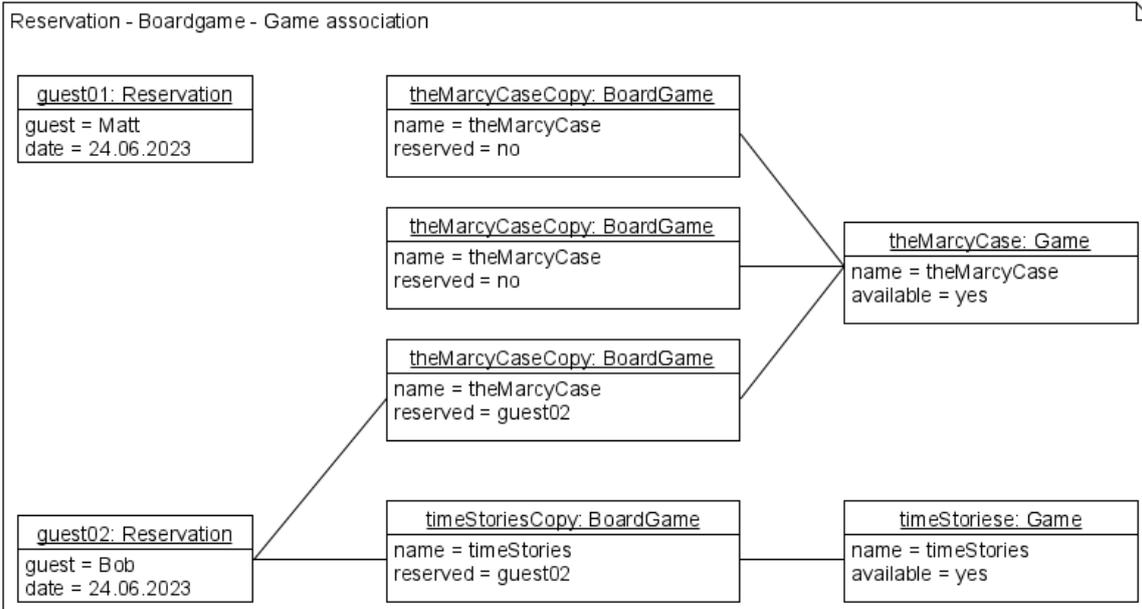
T6. 01:30:27

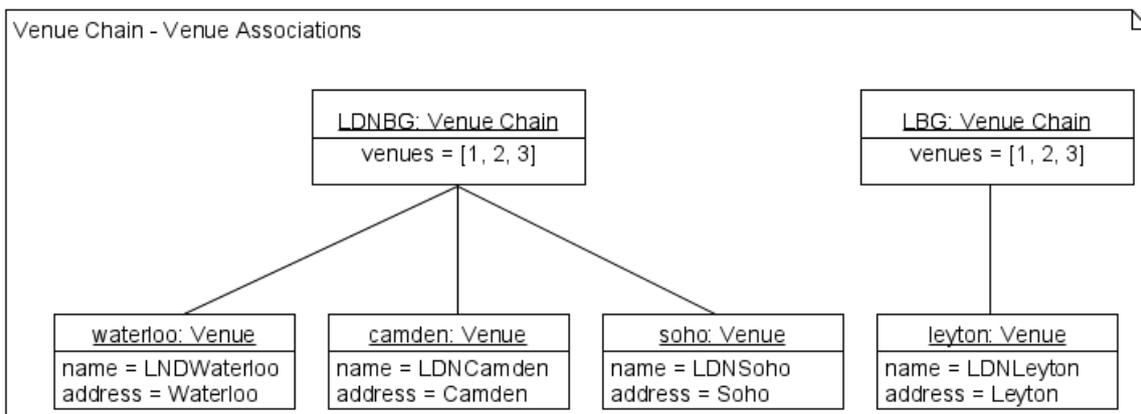
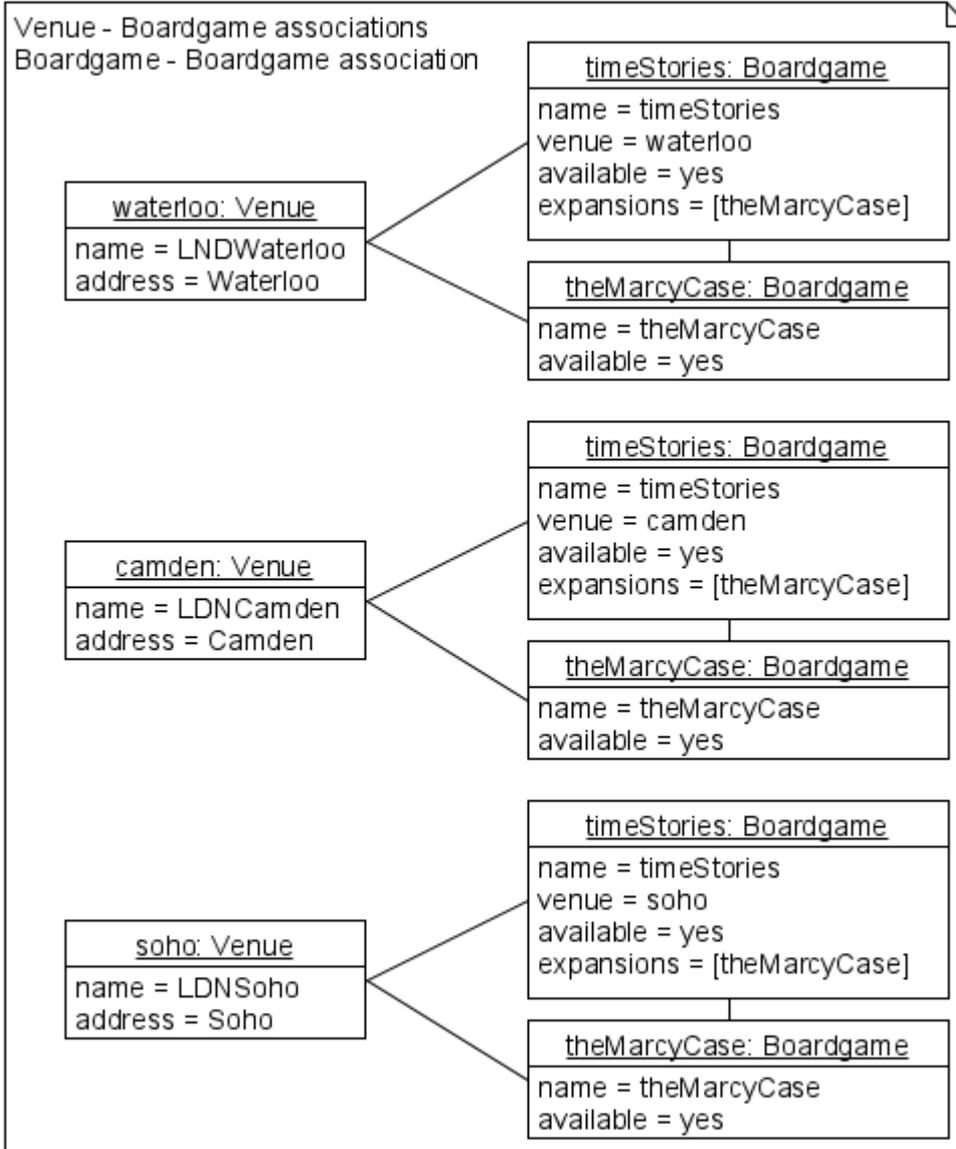
T7.

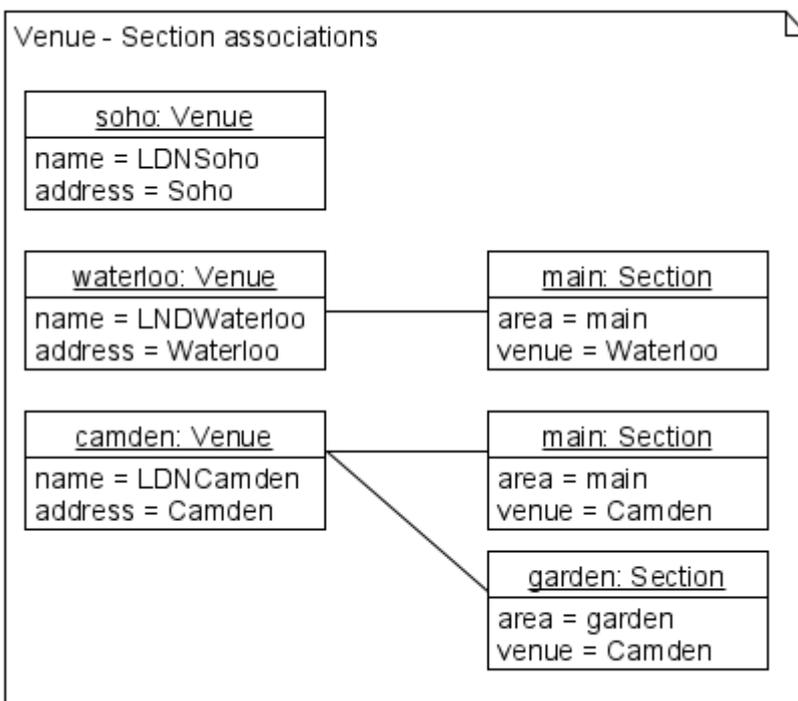
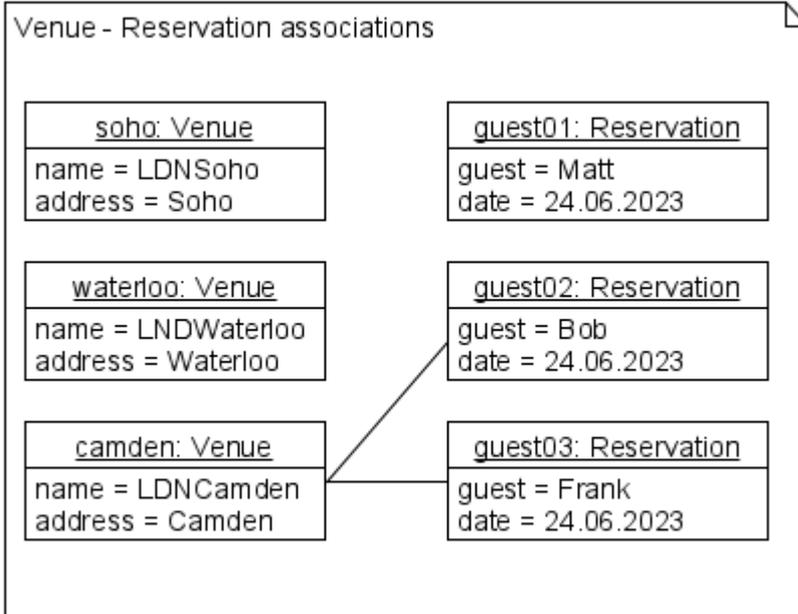
i. Easy to see everything all in one place. Is the search bar used to search through all emails or just the selected one - confusing.

T8.

i. Mismatch colours, slightly overwhelming at first glance.







Behaviour Models

Data Gathering

Identified Venues and Systems

Name and locations of businesses have been omitted to protect staff and interviewee identities.

- 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.

System Users

For operation of the system a typical user might possess the following physical, sensory and cognitive abilities:

- Kinesthetics or cutaneous recognition for input interaction.
- Visual or auditory senses to see and carry out operations through touch or the use of assisted technology such as screen readers.
- Cognitive abilities including:
 - Attention: Concentrating on select tasks at specific points in time.
 - Perception: Affordance of icons, buttons, distinguishable sounds, speech output, etc.
 - Memory: Recollection of knowledge within the environmental context of the system.
 - Learning: Process steps required to carry out tasks.
 - Reading, Speaking or Listening: to comprehend and convey understanding of tasks and take customer details.
 - Problem-solving/Decision-Making: Organising and managing groups of customers and their booking.
- A limited experience of interaction with digital devices and understanding the concepts of digital buttons and menus etc.

Stage 1: Semi-Structured Interview Questions

1. Begin with a quick introduction and purpose of the interview and observation:
 - a. Who I am.
 - b. What I do.
 - c. Why I'm here.
 - d. What are my expectations.
2. Determine if participant is happy to take part in the project and present them with the consent form. Explain why the consent form is needed, answer any questions and how the data's used.
 - a. Are they willing to participate in current data gatherings?

- b. Are they willing to participate in future data gatherings?
3. Being Interview: Ask general questions, name, where they're from, favourite board game, etc.
 - a. First off, what is your role and how long have you worked here?
 - b. What responsibilities do you have? (Do you float between roles or stay in your assigned section for the shift? – Helps to determine user abilities: attention, problem-solving, etc)
 - c. How much experience do you have with the venues booking system?
 - d. Do you have specific credentials to access the system?
 - e. What are the main activities you use the system for? (Prompt with examples if needed: check-in guests? Modify bookings? etc).
 - f. What is the most common task you have to perform?
 - g. What is the most infrequent task?
 - h. Can you show me the steps involved? (Ask for them to explain what they do as they do it and encourage them to show their personal preference).
 - i. Do you know of any other ways to carry out the same task?
 - j. Which method do you prefer and why? (If applicable).
 - k. Would you be able to show me (if possible) how to:
 - i. Make a booking.
 - ii. Modify a booking.
 - iii. Cancel a booking.
 - l. What is the most complicated task you have to carry out? (What does it involve?).
 - m. What is the simplest task you have to carry out?
 - n. Do you remember what your first impressions of the system were when your first had to learn how to use it? (And can you expand on that? – How did you feel?).
 - o. Is there anything you can tell me about the system that you find confusing? (Like you can't do something that should obviously work or makes sense logically).
 - p. What feature would you want added to make any general/specific task easier?
 - q. What is the most frustrating thing about the system and why?
 - r. What do you like most about the system (Colours, layout, ease of use, etc...).
 - s. Is it obvious to you what each button or element on the screen does?
 - t. In three words, describe how you feel about the system.
4. Bring the interview to a close: Thank them for participating, their answers will be of great use and support for me and ask them to get the next person.

Questions for manager if applicable:

1. Are the systems at both venues interlinked?
2. What made you choose this specific system?
3. What type of system is it? Installed? Web app?
4. What is the system running on?
5. Who is responsible for adding new users?
6. Who is responsible for adding new venues?

Venue Observation Details

- Space: What is the physical space like? Take picture of the environment where appropriate.

- Objects: What physical hinderances are present? (Desk, tables, chairs, decorations, people, etc).
- Equipment: What is the system running on? Desktop? Tablet?
- Software: What type of software is the system? Web application? Installed software?
- Issues and suggestions for further investigation should be clearly marked.

Participant Observation Details

- Actors: What are the names and relevant details of the participants?
- Activities: What are the participants doing and why?
- Acts: What are specific individual actions?
- Time: What is the sequence of events?
- Goals: What are the participants trying to accomplish?
- Feelings: What is the mood of the participants?

2nd Stage Interview

The purpose of these interviews is to:

- 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 gamifications techniques.
- Make sure I am building the correct product.
- Redesigns adapted to stories and evaluations.
- Building additional models (conceptual, analysis, sequence, etc).

There are two methods of play in the gamification techniques, these are:

1. Cooperative mode: Everyone working together to achieve a common goal - Ensures that everyone is able to use the software to an acceptable level such as 'everyone must do a specific task without making more than 3 mistakes', or 'everyone must be able to complete a set number of tasks in multiple ways/a set time'.
2. Competitive mode: Everyone working against each other to try and be the winner - Ensures users are engaging in providing higher volume and better quality feedback.

The interview runs as follows:

Part 1: Beginning the interview:

a) Provide participants with the information sheet and consent form. Explain the purpose of the study:

- Who I am: Matthew Mason
- What I do: I'm currently in my last year studying computer science at The Open University, and as part of my final project I am planning and building an online booking system.
- Why I'm here: I'm here to gather feedback from users that will influence the final design.
- What are the expectations: I'll be getting you to help decide how the system will look by building the interface from different components.
- Gamification: To try and make things a little bit more interesting and fun, I've turned it into a game where you'll get points for completing specific tasks, answering questions correctly or getting the correct majority answer. There will be a cooperative element where everyone's score will contribute to reaching a specific target and a competitive element where you'll be trying to get the highest score against everyone who participates (Winner might get a prize: TBD. Score will be determined later after the project has finished).
- Why the consent form: Agrees to participate.
- Answer any questions.
- Explain how the data is used.

b) Check consent form for what user has agreed to and begin interview

c) Ask for name, position and experience level with the existing 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: Present users with categories of icons and get each to describe the set in one word. The majority will receive the highest points.

Set A: Add Booking.

Set B: Booking View.

Set C: Details.

Set D: Enquiries.

Set E: Split Screen.

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

- a. Plus symbol: Add, New, Book, Submit.
- b. Undo symbol: Undo, Return, Before.
- c. Redo symbol: Redo, Forward, After.
- d. Booking symbol: Appointment, Booking, Reservation.
- e. Enquiry symbol: Enquiry, Mail, Request.
- f. Details symbol: Details, Information, Specifics.
- g. Split Screen symbol: Split, Divide, Separate.
- h. Help symbol: Help, Assist, Support.
- i. Settings symbol: Settings, configure, Adjust.

Success Criteria: Learnability: Tests user 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.

Mode: Cooperative

Part 3: The Grid View:

Test 3:

A. Present users with elements for the booking layout and get each to assemble them in their preferred orientation (3 grid elements, 1 booking entry component, 1 web bar component and 1 ribbon component).

B. Give each participant the reminders components, explain how they work, that it can be pinned in place, minimised, etc, ask them to place one of them where they would like it.

Test 4: Now that all components have been selected and are in place, which icon/word best represents each view. The participant can change their mind from earlier selections.

- I. Why did you choose this configuration and what do you like about it?
- II. What didn't you like about the other views?
- III. Why did you put the reminder column on this side?
- IV. What about the design could be improved or added to make it better?

Test 5: Do you understand each element in the view, and can you tell me what they are?

- Ribbon (1 point)
- Each button (1 point per button: add, undo, redo, book, enquiry, details, split, help, settings)
- Pin ribbon button (2 point)
- Expand ribbon button (2 point)
- Reminder's component (1 point)
- Minimise Reminder component button (2 points)
- Reminder filter (1 point)
- Booking grid view (1 point)
- Venue select (1 point)
- Date select (1 point)
- Lock grid button (1 point)
- Table column (1 point)
- Table capacity column (1 point)
- Timeline (2 points)
- Timeline lock button (2 points)

+10 points for perfect score

+8 points for greater than 18

+ 6 points for greater than 15

Questions regarding the grid view:

- I. When zooming in and out, how would you expect the grid to change?
- II. Which of the following methods would you use to zoom in and out:
 - a. Buttons configured to select specific views (open-close, 6/8/12/24 hours).
 - b. Buttons to adjust time and tables simultaneously.
 - c. Sliders to adjust individual zoom levels (time/tables).
 - d. Control with mouse wheel
- III. Which of the following would you use to navigate the grid view?
 - a. Click and drag (Horizontal/Vertical)
 - b. Horizontal/Vertical scroll bars.
 - c. Mouse wheel and buttons (right mouse click)
 - d. Keyboard (page up & down).
 - e. Touch.
- IV. In regard to the timeline (the red vertical line), should it move from left to right while the grid stays in a static position, the timeline stays static and the whole grid moves from right to left, or both?
- V. What is the minimum amount of detail required on a booking component?

- VI. What additional information would it be nice to have available at a quick glance?
- VII. How is that represented in the existing system?

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

Mode: Competitive and cooperative

Part 4: Testing Effectiveness and Efficiency:

Test 6: Present the participant with the necessary components for this test (marked test components: Compact, Regular or Large). This test requires the participant to follow a given set of instructions, placing booking components on the grid, the reminders panel, rearranging them, undoing and redoing parts, changing bookings, updating bookings, managing conflicts, etc. This test is timed and should finish with a specific configuration. Points will be awarded for fastest time and for each component in the correct place at the end of the test.

Test booking pieces will only have the full name on them.

- i. Tori Walker: 13:30 | Tab: 12 | Party Size: 2 | Length: 3 hours
- ii. Daniel Bird: 12:30 | Tab: 1 | Party Size: 2 | Length: 2 hours
- iii. Olivier Webb: 12:00 | Tab: 4 | Party Size: 6 | Length: 3:30 hours | Checked-in
- iv. Ann Wilkinson: 10:00 | Tab: 5 | Party Size 6 | Length: 2:30 hours | Not Updated
- v. Callie Huff: 13:00 | Tab: 6 | Party Size: 4 | Length: 3 hours
- vi. Fiona Watkins: 15:00 | Tab: 3 | Party Size: 4 | Length: 3 hours
- vii. Denis Sharpe: 09:00 | Tab: 9 | Party Size: 2 | Length: 2 hours
- viii. One of the staff members has updated Ann Wilkinson's booking. Make the changes that reflect this update.
- ix. Alana Carter: 13:00 | Tab 5 | Party Size: 6 | Length: 3 hours
- x. We have an enquiry: Paige Garrett wants to book a table at 15:00 for 3 hours for a party of 6 but there aren't any tables available. Place this booking in the enquiries row at the top at the correct time.
- xi. Arjan Hale: 12:15 | Tab: 2 | Party Size: 2 | Length: 3 hours
- xii. Pippa Melendez: 08:30 | Tab: 13 | Party Size: 4 | Length: 3 hours | Not Updated

1st Place gets 20 points then 1 less for each position after.

Take photo to record layout of components and placement of booking components.

Part 5: Enquiries and split view initial impressions:

Test 7: For those users who have experience with replying to enquiries, provide the user with the corresponding components and ask them to arrange them in the configuration they prefer. Explain what each of the components does and answer any questions they may have.

- I. What are their initial impressions of this layout and what changes would they make?

Test 8: For more advanced users, provide them with the split screen components and ask them to arrange them in their preferred configuration. Explain any necessary components.

- II. What are their initial impressions of this layout and what changes would they make?

Part 6: Reset:

Thank participants for their assistance and reset the components ready for the next user.

Literature Search

Relational Database Design and Implementation (Harrington, 2016)

An initial first scan indicates 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.

After a more thorough read, I can see a number of sections that will prove relevant to my project:

- 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 key 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 the Codecademy course will provide and extend my knowledge providing a comprehensive, practical and theoretical framework to base my own database on.

Learn PostgreSQL (Ferrari & Pirozzi, 2020)

The interest in this book stems from knowing that part of the Codecademy course which I will be following has sections covering PostgreSQL so this would be an additional resource to compliment learning and cross-reference/confirm knowledge. On first glance, this appears to be aimed at those familiar with databases already and provides an introduction in the first few chapters that explain:

- What PostgreSQL is, its history, versions, dependencies, cluster anatomy, how configuration files are used, command line utilities.
- Management of users and connections, the concept of a “role”, account and group creation.
- Basic and Advance statements.

From chapter 6 and onwards the book delves into seemingly more advanced topics that I will require such as server-side programming, extending the database and physical and logical replication. It also expects prior knowledge of the Linux OS. This resource may be handy later on as a reference and to consolidate knowledge and understanding but may be at a level too high for an introductory reading.

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 starting 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 patterns to abstract a recognisable quality and apply it to other designs. However, pattern languages can be more powerful, albeit less common, as they incorporate a network of patterns that reference 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.

Understanding APIs and RESTful APIs Crash Course (Taulien, 2020)

A very short 40-minute video introduction to understanding what APIs are, how they work and why they're important. While it doesn't provide any guidance on how to build your own API, as a very basic introduction to APIs and request types, with provided examples, it will provide a good starting point and reference for future development of my project.

RESTful Web API Design with Node.js (Hamadeh, 2016)

A 2-hour video course providing an overview of three RESTful APIs, the steps to build them and their similarities and differences. The three APIs explored are Twitter, Facebook and GitHub. It also looks at best practices to keep APIs secure, maintainable and ensure they perform well. Similar to the previous API Crash course, this feels like more of an introduction to APIs rather than a comprehensive guide or reference on how to build them. With that being said, the Codecademy course uses and implements back-end scripting using Node.js and my project will be developed using HTML, CSS and JavaScript so I feel like this would be a comparatively worthwhile resource to incorporate into my literature.

Node.js – The Complete Guide (Schwarz Müller, 2019)

A longer more comprehensive video course that guides readers from installing Node.js, working with Express.js, sessions and cookies, authentication, sending emails (may be particularly useful if I am to develop the application's ability to automatically send confirmation and reminder emails), validation, Async requests, REST APIs, WebSockets, deploying applications, testing plus a range of other topics. It provides a substantial amount of practical content alongside the theory and will provide a good reference for my project due to the large amount of content available.

REST API Development with Node.js: Manage and Understand the Full Capabilities of Successful REST Development 2nd edition (Doglio, 2018)

Building on the previous three video courses this textbook is a full course beginning with the history of REST, theoretical development, practical API development to the use of Node.js modules to create a RESTful API. It covers requirements gathering to tools section and troubleshooting.

The contents appear to be clearly laid out and structured in a logical way for learning how to develop a REST API starting with an introduction to REST, then following up with API Design best practices, Node.js, Architecting a REST API, working with Node.js modules, REST API planning and development, testing, deploying and finally troubleshooting.

The idea will be to use the video courses and this textbook in conjunction with each other to try and fully understand all the necessary elements required for my web application. While they all contain more advanced materials, they also provide a good reference point for the Codecademy material and alternative points of view and perspectives on how to develop APIs.

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 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 including points,

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 examples provided utilize user stories and gamify 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.

The core participants group for my project, 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.

Glossary

Term	Category	Definition
Venue Chain	Concept	The company who manages the venues
Venue	Concept	One of the venues in the chain
Boardgame	Concept	A physical copy of a <i>Game</i> that is available for guests to play
Game	Concept	A record of a specific game that is available at the venue.
Section	Concept	An area within a venue that occupies a physical space and contains any number of tables and table types
Table	Concept	Furniture upon which guests can play games
Table Type	Concept	The properties of the table such as topper, if a drinks holder is available, etc.
Reservation	Concept	A record of the fact a particular table has been held for a given period by a guest
Date	Concept	A day on which reservations can be associated
Guest	Concept	A person who reserves a table
Reminder	Concept	A notification delivered by the system indicating how long until a reservations status change
Booking status	Domain Vocabulary	The current phase of a reservation's life: Arriving later, Arriving soon, Check-in, Warning, Payment, Cleared.
Topper	Domain Vocabulary	The material that sits on top of a table such as felt, a plastic mat, or removable wooden panels for recessed tables.
DM	Domain Vocabulary	Stands for Dungeon Master: The storyteller of roleplaying games.
GM	Domain Vocabulary	Synonym for DM, stands for Games Master
Role Playing	Domain Vocabulary	Guests take on the personal of a character for the duration of a story
Take	Domain Vocabulary	Offering and holding alternative booking options for a guest before being committed to the system
VenueChain-Venue	Association	A <i>Venue</i> belongs to the <i>VenueChain</i> (There must be at least two <i>Venues</i> for it to be considered a chain)
Venue-Boardgame	Association	A <i>Venue</i> stocks copies of a range of different <i>boardgames</i>
Venue-Reservation	Association	A <i>Venue</i> has a number of <i>Reservations</i> both past and present
Venue-Section	Association	A <i>Venue</i> can have a number of different <i>sections</i> dividing the space of the venue
Venue-Table	Association	A <i>Table</i> belongs to a <i>Venue</i>
Venue-TableType	Association	A <i>Venue</i> contains a number of <i>TableTypes</i>
Table-TableType	Association	A <i>Table</i> is of a specific <i>TableType</i>
Table-Section	Association	A <i>Section</i> contains a number of <i>Tables</i> . (A table may exist outside a section if broken, removed or in a non-designated section).
Table-Guest	Association	A <i>Guest</i> can occupy a <i>Table</i>
Reservation-Date	Association	A <i>Reservation</i> is for a specific <i>Date</i> (A reservation may start on one day and finish on another and therefor be connected to two <i>Dates</i>)
Reservation-Reminder	Association	A <i>Reminder</i> belongs to a <i>Reservation</i>
Reservation-Boardgame	Association	A <i>Boardgame</i> can belong to a <i>Reservation</i>
Reservation-Guest	Association	A <i>Guest</i> can have a number of <i>Reservations</i>
Reservation-Table	Association	A <i>Table</i> is allocated to a <i>Reservation</i>
Reminder-Date	Association	A <i>Reminder</i> updates based on <i>Date</i> and time

Game-Game	Association	A <i>Game</i> can be an expansion for another <i>Game</i> . It can also be an expansion that can be played without the base <i>Game</i> .
Game-Boardgame	Association	A <i>Game</i> has multiple <i>Boardgame</i> copies
venue	Attribute of <i>VenueChain</i>	The venues that belong to the <i>VenueChain</i>
chain	Attribute of <i>Venue</i>	The <i>VenueChain</i> the <i>Venue</i> belongs to
Tables	Attribute of <i>Venue</i>	The <i>Tables</i> that belong to the venue
reservations	Attribute of <i>Venue</i>	The <i>Reservations</i> at the venue
Boardgames	Attribute of <i>Venue</i>	The <i>Boardgames</i> that are available at a <i>Venue</i>
Section	Attribute of <i>Venue</i>	The <i>Sections</i> of a <i>Venue</i> (for example main, garden, roof...)
copyId	Attribute of <i>Boardgame</i>	The unique identifier of a boardgame
game	Attribute of <i>Boardgame</i>	The <i>Game</i> of which the <i>Boardgame</i> is a copy of
available	Attribute of <i>Boardgame</i>	A signifier of whether the <i>Boardgame</i> is available to use
venue	Attribute of <i>Boardgame</i>	The <i>Venue</i> that particular <i>Boardgame</i> is stocked at
name	Attribute of <i>Game</i>	The name of the <i>Game</i>
expansions	Attribute of <i>Game</i>	The expansions that belong to a <i>Game</i>
baseGame	Attribute of <i>Game</i>	The <i>Game</i> the expansions belongs to
copies	Attribute of <i>Game</i>	The number of copies of the <i>Game</i>
style	Attribute of <i>TableType</i>	The classification of <i>Table</i> (for example tabletop, recessed, interactive...)
topper	Attribute of <i>TableType</i>	The cover of the <i>Table</i> (for example, felt, removable panels, plastic, tile...)
material	Attribute of <i>TableType</i>	The material the <i>Table</i> is made from (for example oak, pine, beech, metal...)
features	Attribute of <i>TableType</i>	The characteristics of the <i>Table</i> (for example drinks holders, draws, shelves, USB, led, GM screens...)
tableNo	Attribute of <i>Table</i>	The number allocated to identify the <i>Table</i>
capacity	Attribute of <i>Table</i>	The number of people the <i>Table</i> can accommodate
venue	Attribute of <i>Section</i>	The <i>Venue</i> the <i>Section</i> belongs to
areaName	Attribute of <i>Section</i>	The name of the <i>Section</i>
firstName	Attribute of <i>Guest</i>	The first name of a <i>Guest</i>
lastName	Attribute of <i>Guest</i>	The last name of a <i>Guest</i>
phone	Attribute of <i>Guest</i>	The phone number of the <i>Guest</i>

<i>email</i>	Attribute of <i>Guest</i>	The email address of the <i>Guest</i>
<i>date</i>	Attribute of <i>Date</i>	The <i>Date</i> (Can be current, past or future)
<i>time</i>	Attribute of <i>Date</i>	The <i>Time</i> of the current date
<i>status</i>	Attribute of <i>Reminder</i>	The current <i>Reminder</i> period of the <i>Reservation</i> (for example, Arriving later, Arriving soon...)
<i>colour</i>	Attribute of <i>Reminder</i>	The colour of the current <i>Reminder</i>
<i>venue</i>	Attribute of <i>Reservation</i>	The <i>Venue</i> the <i>Reservation</i> is for
<i>guest</i>	Attribute of <i>Reservation</i>	The <i>Guest</i> that the <i>Reservation</i> is for
<i>date</i>	Attribute of <i>Reservation</i>	The <i>Date</i> on which the <i>Reservation</i> starts and ends
<i>startTime</i>	Attribute of <i>Reservation</i>	The time at which the <i>Reservation</i> begins
<i>endTime</i>	Attribute of <i>Reservation</i>	The time at which the <i>Reservation</i> finishes
<i>boardgames</i>	Attribute of <i>Reservation</i>	The <i>Boardgames</i> that have been made available for the <i>Reservation</i>
<i>reminder</i>	Attribute of <i>Reservation</i>	The <i>Reminder</i> that manages status changes and updates for the <i>Reservation</i>
<i>table</i>	Attribute of <i>Reservation</i>	The <i>Table</i> that has been allocated to the <i>Reservation</i>
<i>referenceNumber</i>	Attribute of <i>Reservation</i>	The unique number identifying the <i>Reservation</i>

REFERENCES

Doglio, F., 2018. *REST API Development with Node.js : Manage and Understand the Full Capabilities of Successful REST Development*. 2 ed. s.l.:Apress.

Ferrari, L. & Pirozzi, E., 2020. *Learn PostgreSQL*. s.l.:Packt Publishing.

Hamadeh, S., 2016. *REST API Development with Node.js*. s.l.:Packt Publishing.

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.

Schwarz Müller, M., 2019. *Node.js - The Complete Guide*, s.l.: Packt Publishing.

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

Taulien, K., 2020. *Understanding APIs and RESTful APIS Crash Course*, s.l.: Packt Publishing.

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: Prioritizing Requirements*. pp. 1 - 6.