



---

# TMA 03 – DRAFT PROJECT REPORT

---

Matthew Mason: C6122243



# TABLE OF CONTENTS

---

Developing a web application booking system for a three-venue boardgame café business .....	2
Problem Description .....	2
Aims & Objectives.....	3
Related Literature .....	3
Project Work.....	5
Research and The Design Context .....	5
System Users .....	5
Activity and Environment .....	5
Initial Design Sketches .....	6
Success Criteria.....	7
Interview Analysis .....	8
Domain Modelling.....	8
Business Rules.....	8
Business Processes.....	9
Use Case Model .....	15
Elaborated Use Cases .....	15
Prototype Design .....	17
Further Work.....	21
2 <sup>nd</sup> Stage Interview - Draft .....	21
Review .....	22

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

---

## Problem Description

The current reservation process at each venue has caused a number of problems such as miscommunication of details, double bookings, recording incorrect information, loss of customers details and incorrect booking cancellations, etc.

The process involves customers calling the venue direct where a staff member records the booking details and checks it against current 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 unreliability, inconsistent and untrustworthy services, lost business, revenue, trust and a bad reputation. The business has decided to rectify this by commissioning a booking system to counteract these problems.

One potential solution is developing a responsive web application for use with tablets and desktops to record customer bookings. This provides a shared centralised management solution accessible from anywhere by utilising web-based technologies, HTML, CSS and JavaScript, working across multiple devices. Combining ethernet, Wi-Fi and cellular data provides redundancy to overcome the necessity of requiring an always-online connection to access the system and staff can access the system using assigned credentials to manage user bookings, rearranging tables, dates, adding comments and fulfilling specific customer requirements.

A number of other solutions were considered:

- Installing dedicated software on PC's and tablets, however this would significantly increase development, maintenance and infrastructure costs.
- Developing a hybrid application, however this would notably place limits on functionality, usability and user experience goals through a reduced screen size affecting the apps utility and efficiency.

I will utilise personal experiences of managing bookings as a restaurant host, the shortcomings of using its paper-based reservation system, plus my knowledge and experience from previous modules to develop my skills in software engineering, database and API design and further explore and incorporate 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 projects 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 database records. This will improve the customer experience by alleviating the venues problems and impact the business by providing a number of benefits:

- Prevention of double bookings.
- Reducing lost bookings and cancelled bookings.

- Reducing recording of incorrect information.
- Bookings can be made and confirmed outside normal operating hours increasing business.
- Multi-user management and permission levels for accountability.
- Historical data of reservations if problems arise.

## Aims & Objectives

- To develop a responsive web application booking system providing reservation and email enquiry management, using HTML, CSS, JavaScript and other frameworks, that's accessible from a range of devices.
- Design and build a database.
- Develop a set of APIs to access and manipulate data.
- Extend and use knowledge of computing and IT subjects specific to the project.
- Successfully and independently manage a project from beginning to completion.
- Identify, gather, utilise, analyse and evaluate resources relevant to the project.

## Related Literature

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.

---

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.

---

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

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.

# PROJECT WORK

---

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

## Research and The Design Context

By identifying, researching and utilising a variety of similar boardgame café establishments and the systems they used to manage customer bookings, I was able to develop an initial solution by iterating over the categorisation of users and their abilities, activities and initial sketches in relation to the environment. This then allowed me to develop a set of success criteria and develop interviews with management at each venue.

## 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.

## Activity and Environment

Key activities that provide essential support of the system include:

1. Creating a booking by entering customer and booking details, confirming correct information and closing the process adding the booking to the database.
2. Modification of a bookings details and committing the changes.
3. Cancellation of an existing booking.
4. Navigation of time and dates displayed to search for past and future bookings.
5. Adding a new venue to the system.
6. Adding a new user and setting permissions and access credentials.
7. Reply to enquiries.
8. Updating a bookings status based on the amount of time remaining.
9. Check-in a guest for a booking.

The scope of each activity is constrained by requiring completion before another task can begin; users cannot create two bookings simultaneously. A logical progression relates tasks by requiring completion of one before another; a booking can't be cancelled if there is no booking, a booking can't be made if there is no venue or system administrator set up.

The foyer follows an identical layout at each venue: A solid-oak adjustable height desk deployed at the entrance is enclosed on three sides with a raised countertop and the booking system will be run off a laptop connected via Wi-Fi. There is adequate space for mobility impaired staff and adjustable ergonomic seating provides an accessible comfortable operational environment.

### Initial Design Sketches

Using the design context, I began to develop some sketches for initial consideration and feedback from potential users.

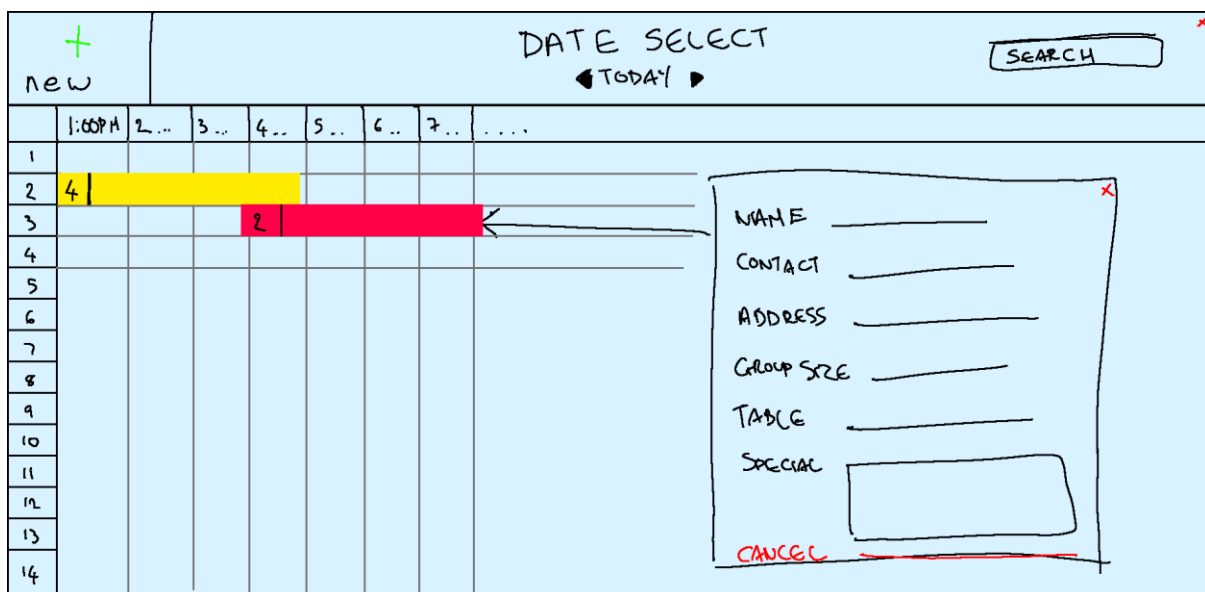


Figure 1: Horizontal grid design

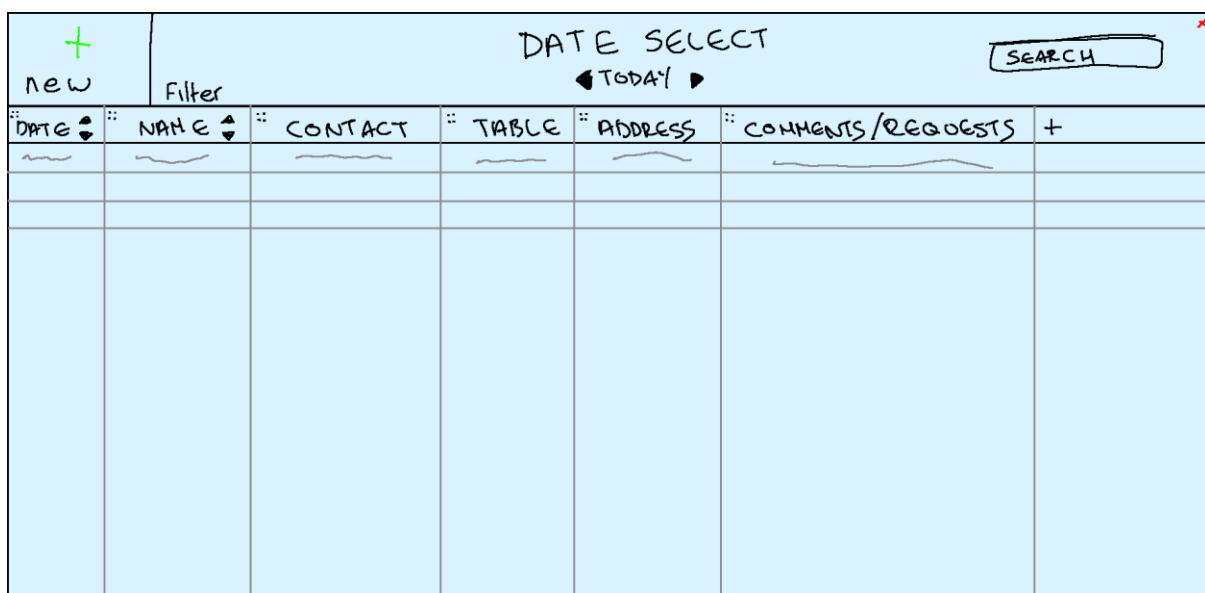


Figure 2: Spreadsheet design

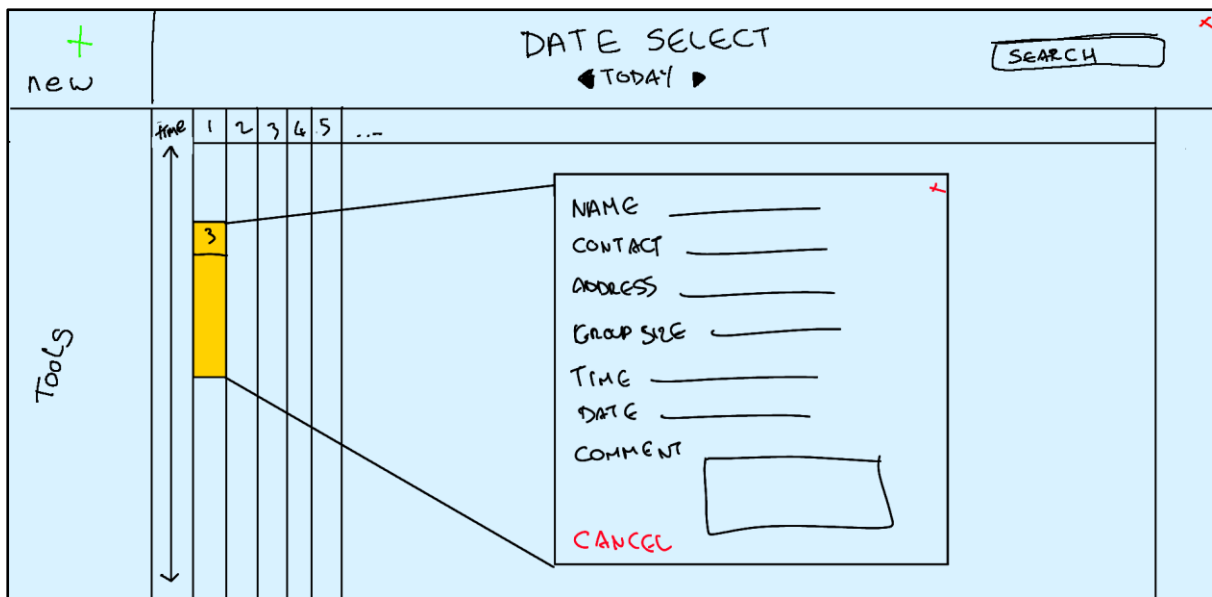


Figure 3: Vertical grid design

## Success Criteria

Using research of establishments, systems, design context and sketches I was able to develop the following success criteria to incorporate as part of further solution development.

**Effectiveness:** The system needs to allow users to carry out work quickly, access information they need, and manage availability.

**Q:** Is the system capable of offering a range of dates, tables and times that can be reserved and recorded into a database?

**Efficiency:** How does the system provide support in carrying out tasks in a minimal number of steps? The system will be accessed frequently and should allow users to return to other jobs as quickly as possible.

**Q:** Can the administrator add a booking in under two minute or under 10 steps.

**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.

**Q:** Is data validation provided to ensure correct dates, times, email addresses, phone numbers, and text entry fields can only accept specific data?

**Utility:** Does the booking system provide the right kind of functionality?

**Q:** Can administrators carry out the range of tasks that allow them to manage bookings, guest information and respond to enquiries in a way that is logical or in a preferred way?

**Learnability:** A top priority should allow swift onboarding and understand of the systems functionality. Is the layout 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)?

**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?



## 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.

## Interview Analysis

Semi-structured Interviews with management and staff at a selection of venues provided both qualitative and quantitative data, along with venue and participant observations details that were used to iterate over and refine initial expectations for what needed to be developed. It also revealed activities that were not considered within initial development activities. The following business rules, processes, activity diagrams and elaborated use cases were all derived from analysing interview data. A full list of interview question can be found in the appendix.

## Domain Modelling

### Business Rules

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

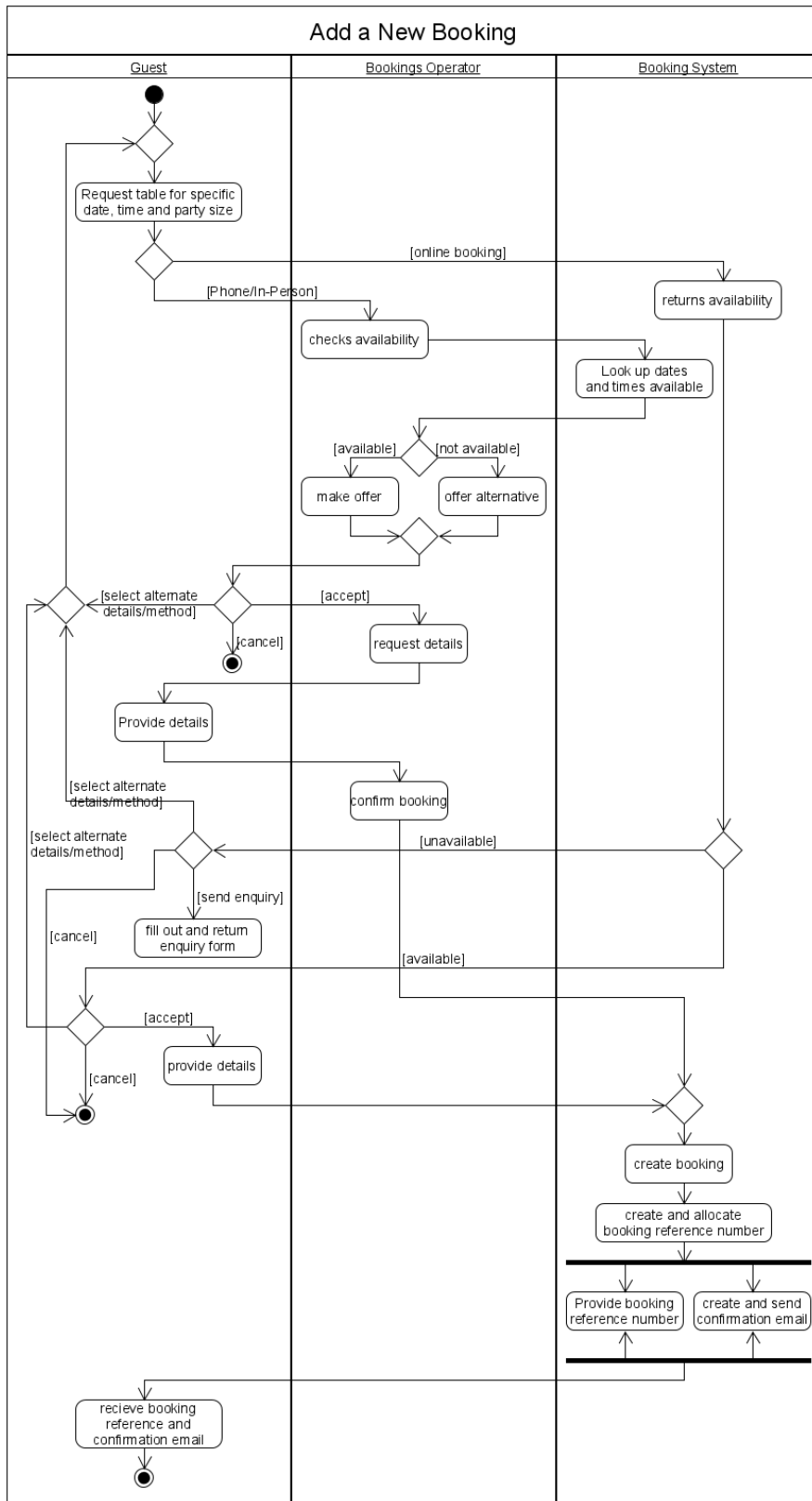


Figure 4: Add a new booking activity diagram.

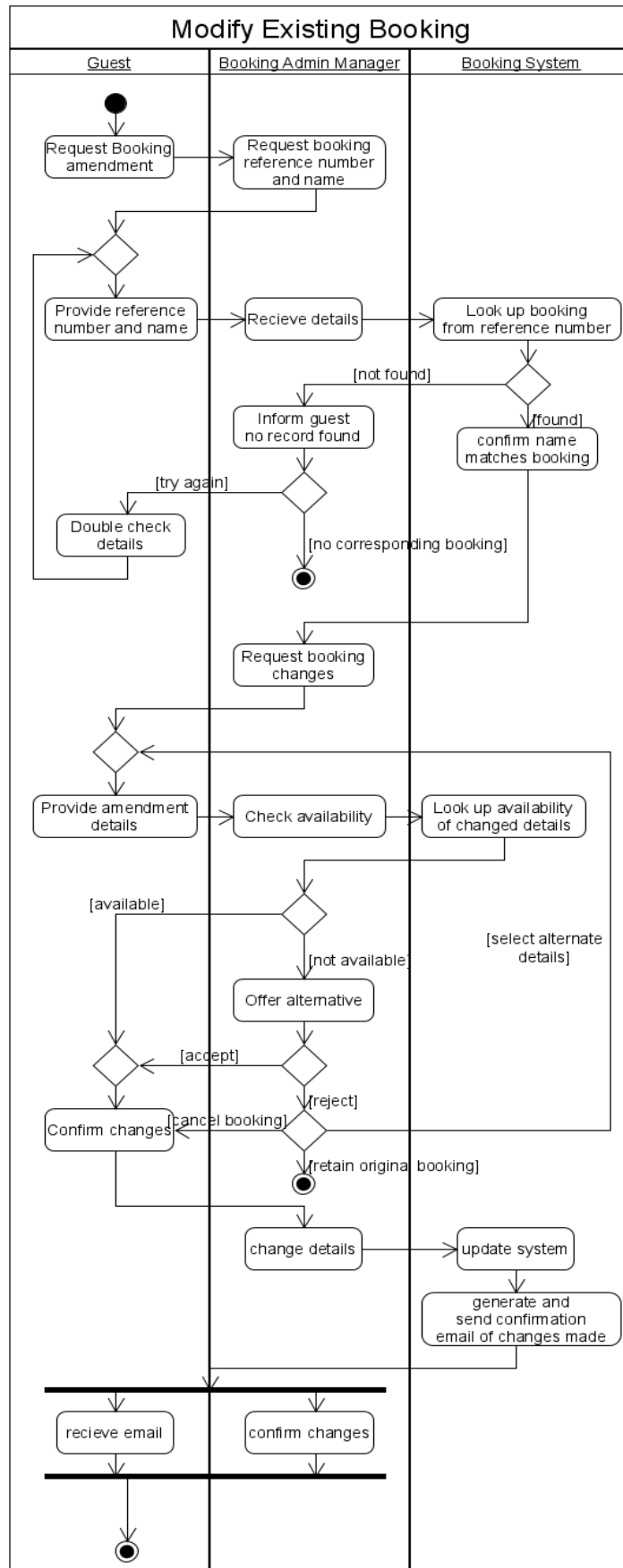


Figure 5: Modify existing booking activity diagram.

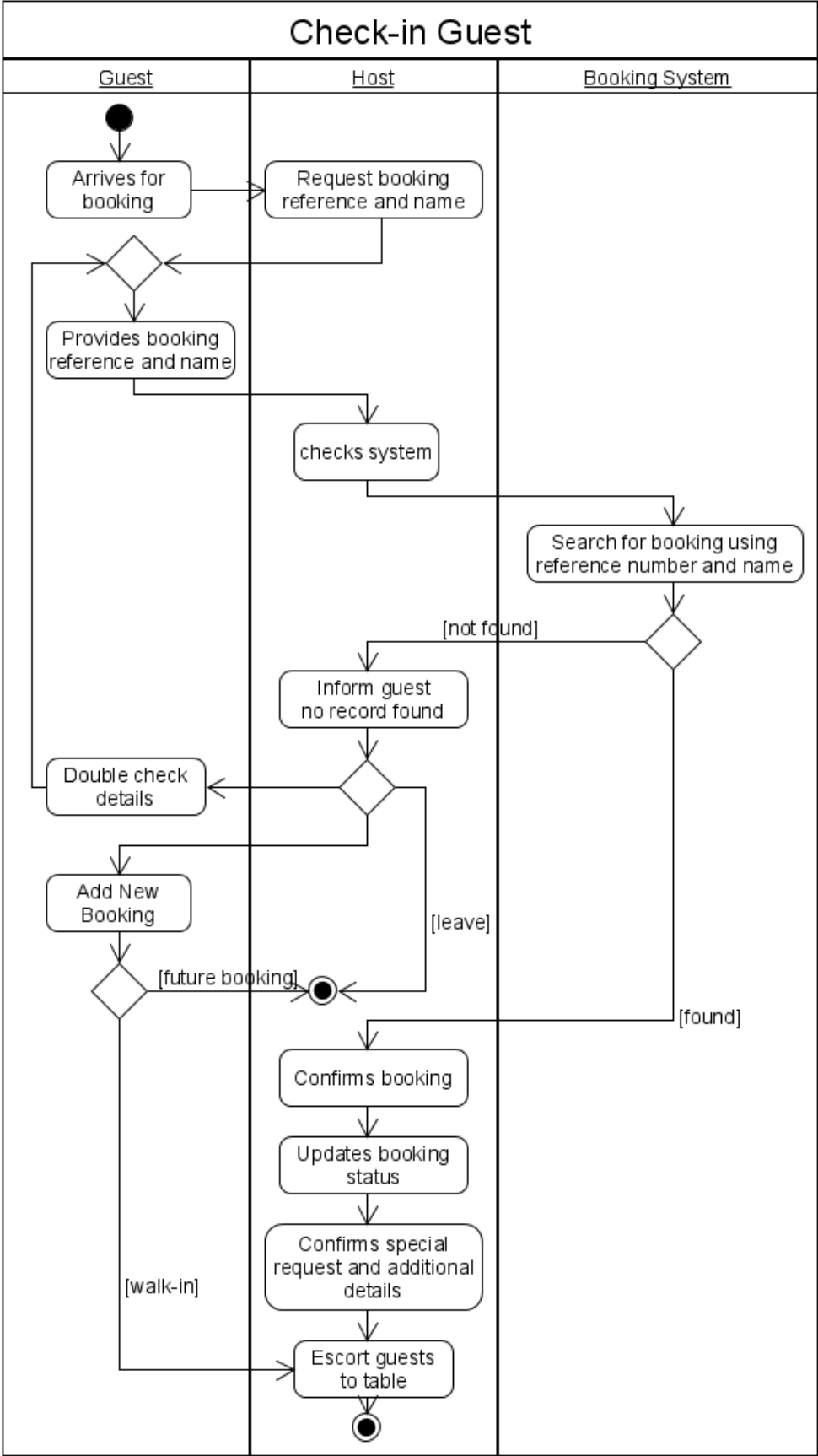


Figure 6: Check-in guest activity diagram.

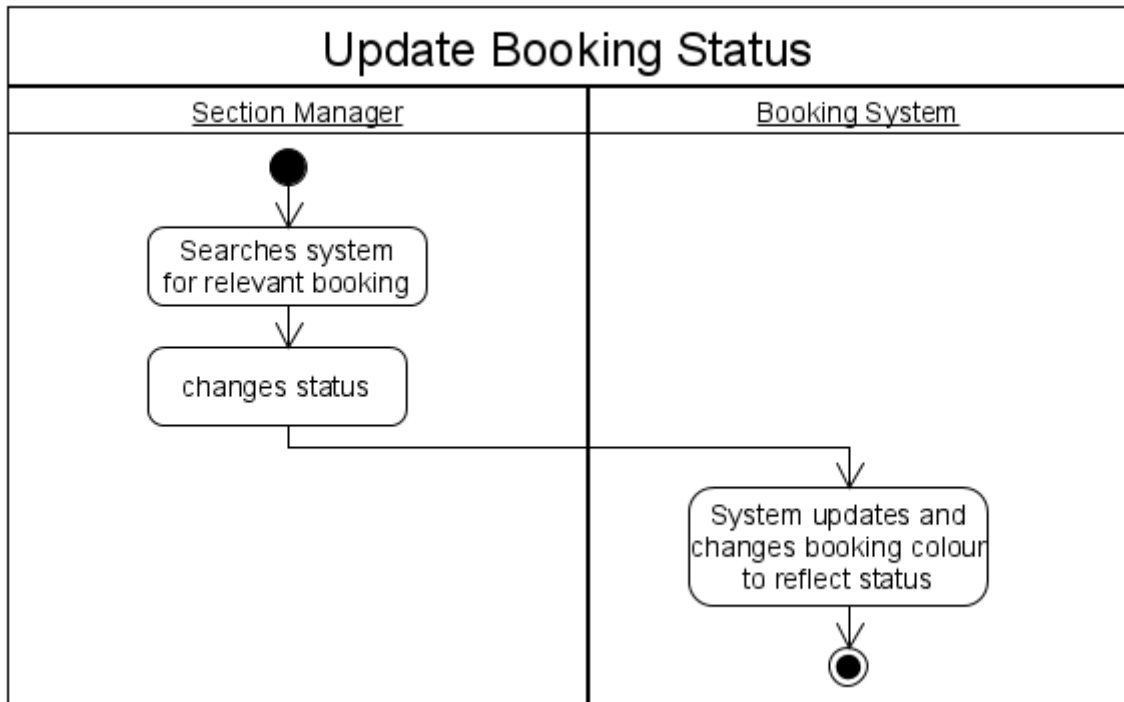


Figure 7: Update booking status activity diagram.

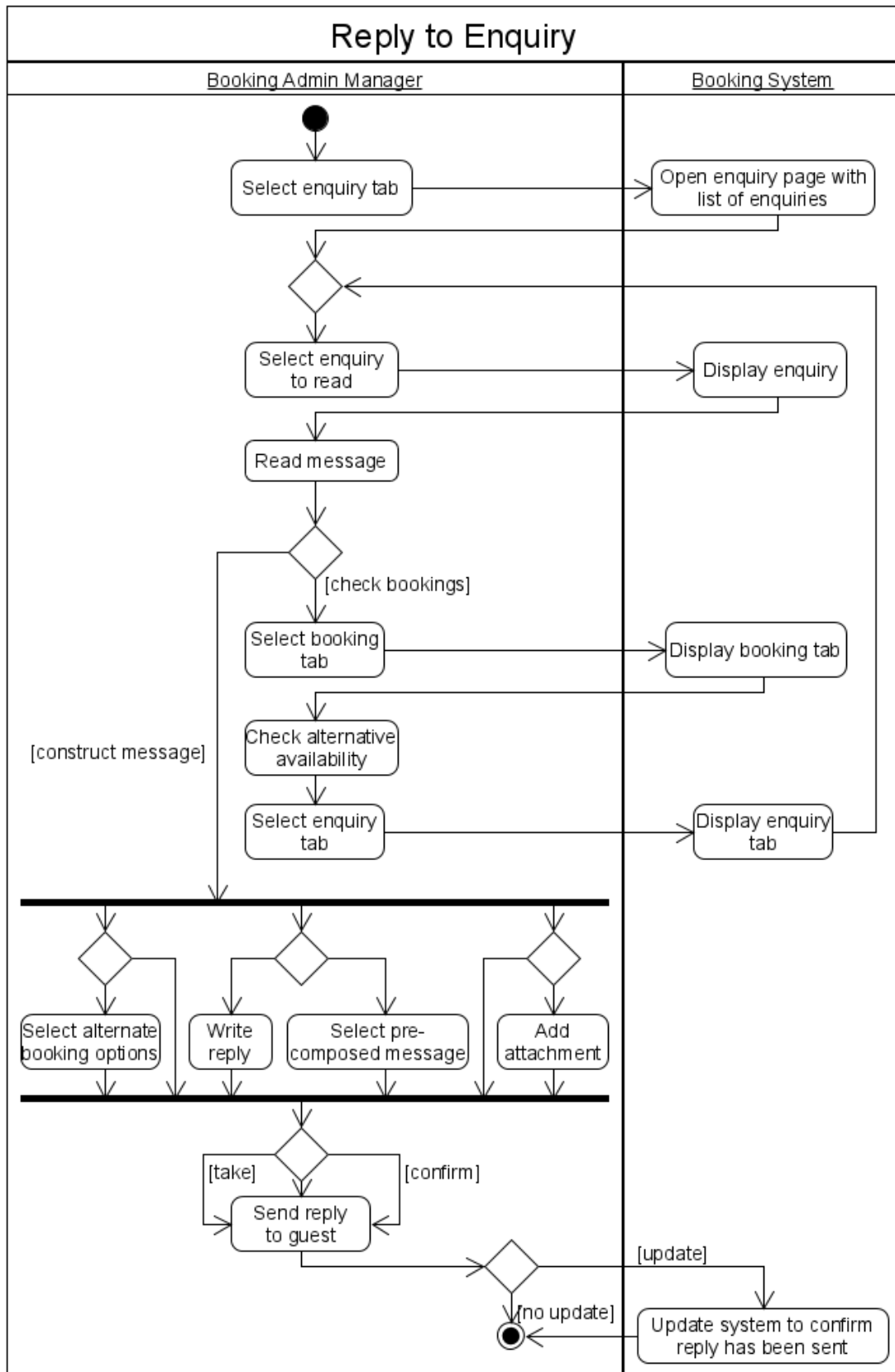


Figure 8: Reply to enquiry activity diagram.

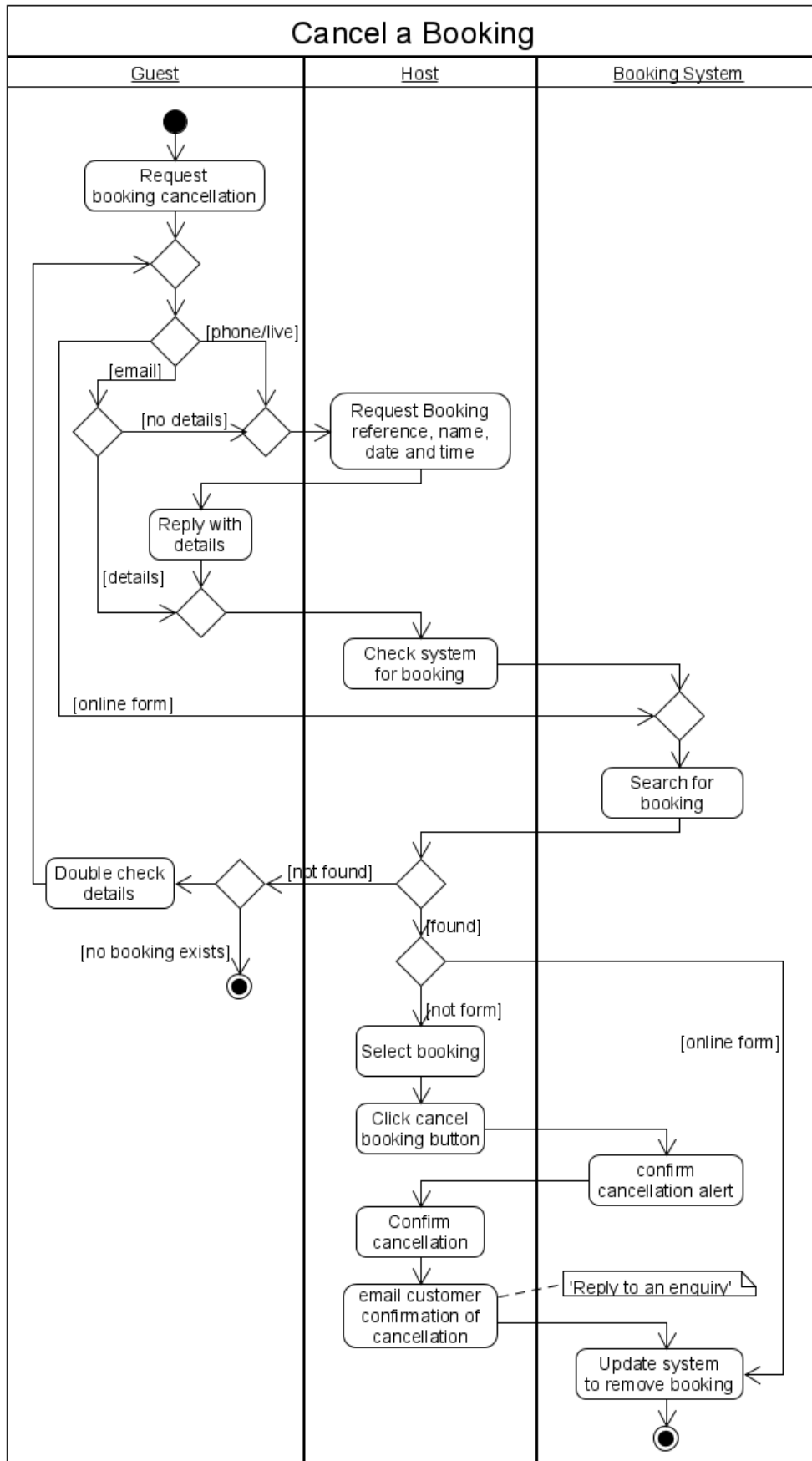


Figure 9: Cancel a booking activity diagram.

## Use Case Model

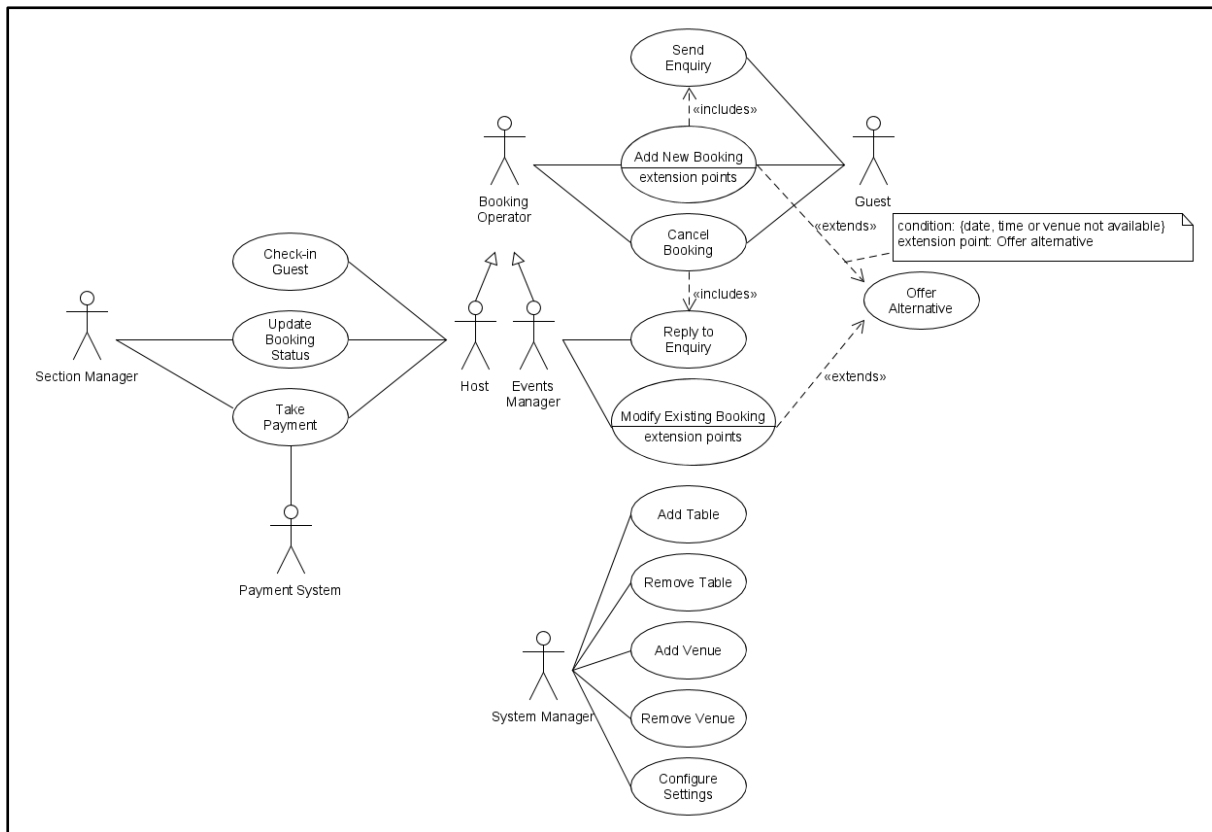


Figure 10: Use case diagram of initial business processes

## 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 agrees to proceed with the offer.
5. The guest provides their full name, contact details, games to reserve, 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.
- 

**Identifier and name:** UC3 *modify existing booking*

**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.

**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.

---

For a full list of initial use cases 1 - 7, please see the appendix.

## Prototype Design

After analysing, evaluating and understanding the domain, design context, business rules, processes and elaborating the use cases, I continued by designing a variety of interactive prototypes based on all my previous research and evaluated them against the success criteria. These prototypes consist of card components and alternate designs so participants can construct their preferred layout and include their reasoning, design refinement using iteration and detailed functional and non-functional requirement engineering can be revealed.

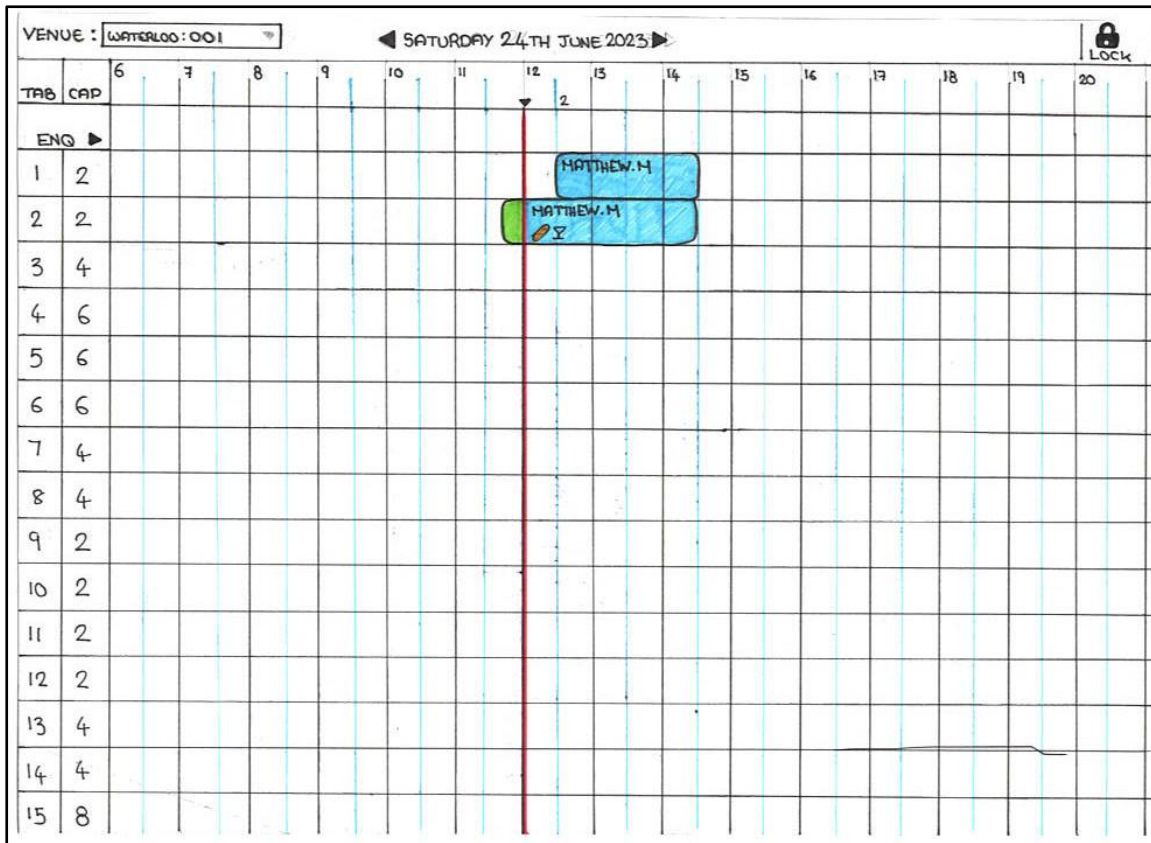


Figure 11: Grid design A.

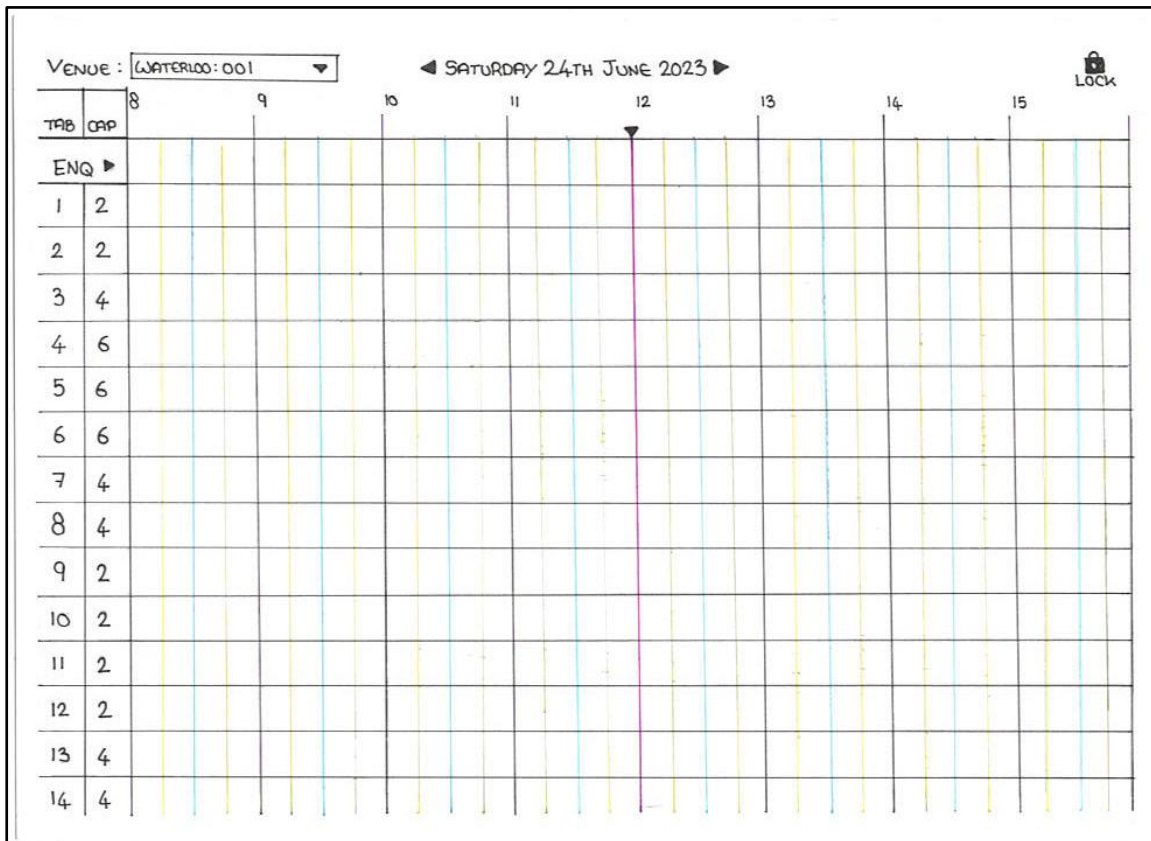


Figure 12: Grid design B.

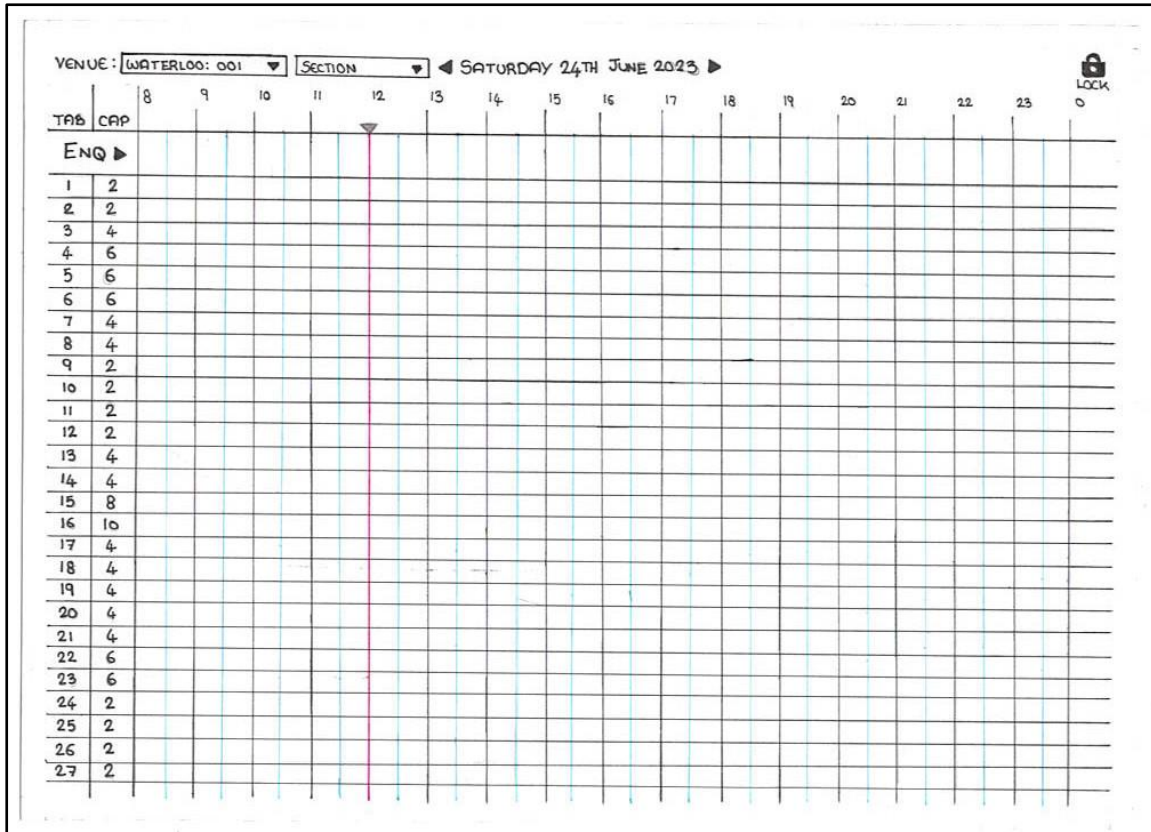


Figure 13: Grid design C.

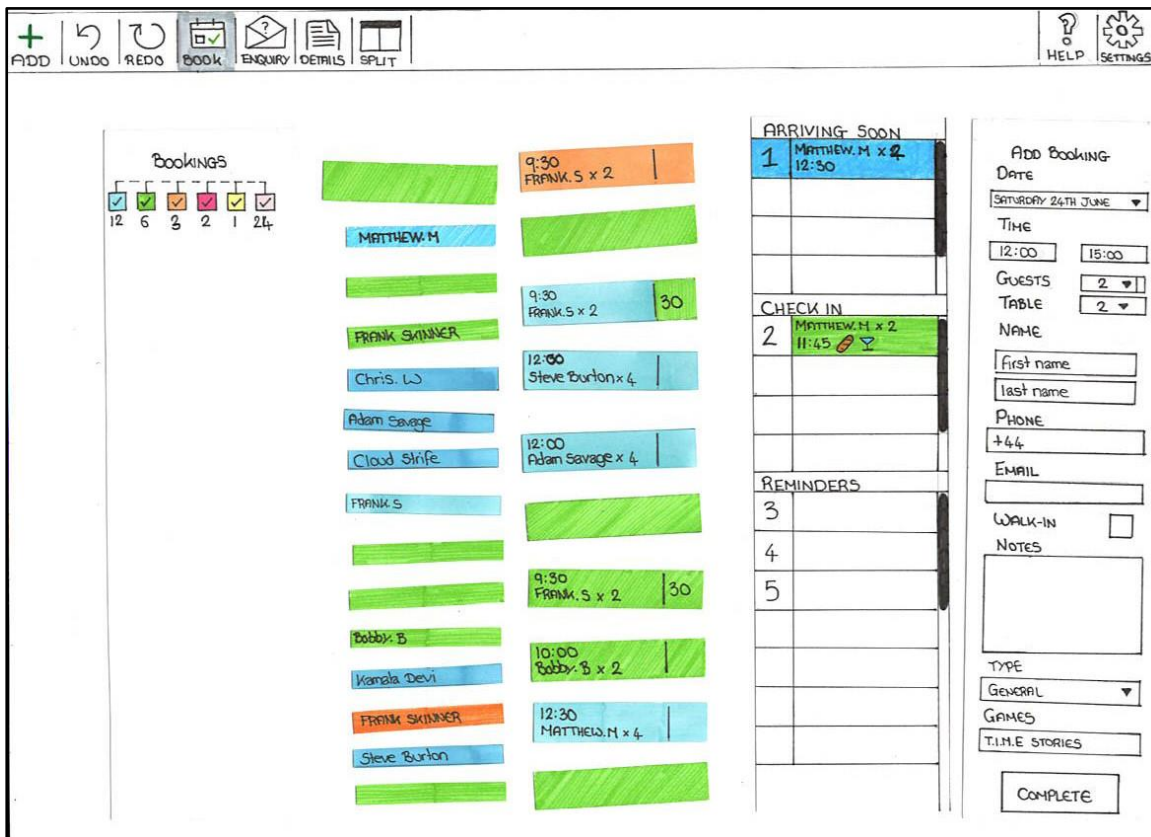


Figure 14: Navigation bar design, toolbar, reminders and components.



## Further Work

I am currently waiting to return to the data gathering and user participation phase of my work to expand and evaluate the prototypes developed and confirm functional and non-functional requirements. This will allow me to include any additional elements in the redesign and coding can commence.

2<sup>nd</sup> stage interviews will consist of participants interacting with the components to determine each user's preferred application layout, their expectations and how they anticipate specific features will work, a small number of tests to help evaluate success criteria further and their feedback on any missing or confusing features. An initial draft of the interview plan is as follows:

### 2<sup>nd</sup> Stage Interview – Draft

1. Ask each user to take the components and assemble them in their preferred arrangement.
2. Why did you choose these elements?
3. Why did you choose this configuration?
4. Why did you put the reminders column on this particular side?
5. Do you understand each element of the view, and can you tell me what they are?
6. Place booking component elements on the grid to simulate an example of a live view.
7. What information do you expect or need on each booking element on the grid?
8. In regard to the reminders: Which colour scheme do you prefer?
  - a. Colour changes to indicate next phase?
  - b. Colour stays the same with progress colour at the end with the timer.
  - c. Colour is the same as displayed on the grid view.
9. How do you expect grid navigation and zooming to work?
  - a. Show more/less time.
  - b. Show more/less tables.
  - c. Both.
10. How would you prefer the grid to progress throughout the day?
  - a. The timeline is locked in the centre and the entire grid moves.
  - b. The grid is locked in place and the timeline moves from left to right across the page.
11. How much time do you expect to see on the grid?
  - a. Open to close.
  - b. 24 hours
  - c. 8 hours
  - d. 6 hours
  - e. Custom
12. Tests:
  - a. Place these booking elements on the grid as fast as possible.
  - b. Find these bookings (Give a set of facedown cards, flip one over and being removing elements, only flip the next one over once you have completed the previous).

## REVIEW

---

Small setbacks have slowed progress slightly, but I am confident the development is moving as expected. An unwelcoming injury forced me to take two weeks rest and I had to delay developing the 2<sup>nd</sup> stage user participation while I recovered, however my circumstances will allow me more than enough time to catch up on additional work that I was unable to include this time. I don't feel like there are any tasks that have gone badly but if I were to start again, I would try to start the project work and data gathering tasks a lot sooner. At this point in time, I still feel like there is a lot of work left to do and I am excited but also very nervous about starting the coding elements of the project. It feels like the direction and skills for completing the project have changed in ways that I didn't anticipate at the start but not enough to make me uncertain of where it is going. I'm having to incorporate more software engineering techniques than I planned which has added to the work, but it's also provided me with some clarity on the project's end goal and reminded me to constantly evaluate my work against the success criteria and throughout the project as a whole.

The additional work was a risk I didn't anticipate and has forced me to reevaluate whether a demonstrable prototype including the use of APIs and a database will be feasible. Having little to no experience with these elements, I feel like the risk may have clouded my expectations of the scope I would be able to achieve when starting, however, with that being said, there is still a little over 2 months left and I have the determination to push myself to produce a solution to the best of my ability. While this is now a major risk, it is the design of the web applications interface and usability and user experience goals which is the heart of my project, so I don't feel like their absence will be a major hindrance and these aspects may have to be included in future work outside of the module.

In my personal development I feel more confident in managing a project and have a much better understanding of what is achievable in a given amount of time. I still feel like I have a lot to learn and will continue to develop my skills in software engineering and interaction design techniques, programming languages like React and SQL as well as API development. The main difficulty I've had is finding relevant academic literature and incorporating it into the work as it feels like most things are outside the scope of my project or have no relevance and so this is a skill I am constantly striving to improve.

### Feedback Request:

I've not had much experience with writing reports so any feedback about style, structure, content of this report would be immensely helpful going into the final report.

Do you feel that the amount of work and progress of my project is sufficient at this point or was there an expectation that a lot more should have been completed by now?

As my project is more geared towards designing a web application interface, if I code the design but it doesn't have interactable elements will this detract from my final grade?

## 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.

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

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

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.