TMA 02

Matthew Mason: C6122243

BSC (HONOURS) COMPUTING AND IT: TM-470

MAY 18, 2023

TABLE OF CONTENTS

Preparation and Planning

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

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 providing numerous 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.

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 as opposed to installing dedicated software on a limited range of devices, requiring multi-platform development and significantly increasing costs. 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.

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 TM352 and TM356 to develop my skills in database and API design and further explore and incorporate interaction design techniques. I will produce a demonstratable booking system comprising a range of functionality including adding, modifying and cancelling a reservation, setting up new venues, employees and exporting database records.

A number of ICT aspects need to be considered including the projects required tools such as VS Code, Chrome dev tools, GitHub, etc, acquiring resources through the OU library, Google scholar and Government websites to extend my learning and understand and implementing legal requirements of storage and transit of customer data including GDPR.

Aims & Objectives

- To develop a responsive web application booking system 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.

Additional resources identified: project management website Monday.com for dynamically scheduling project tasks, subtasks and workflows.

Tasks and Subtasks

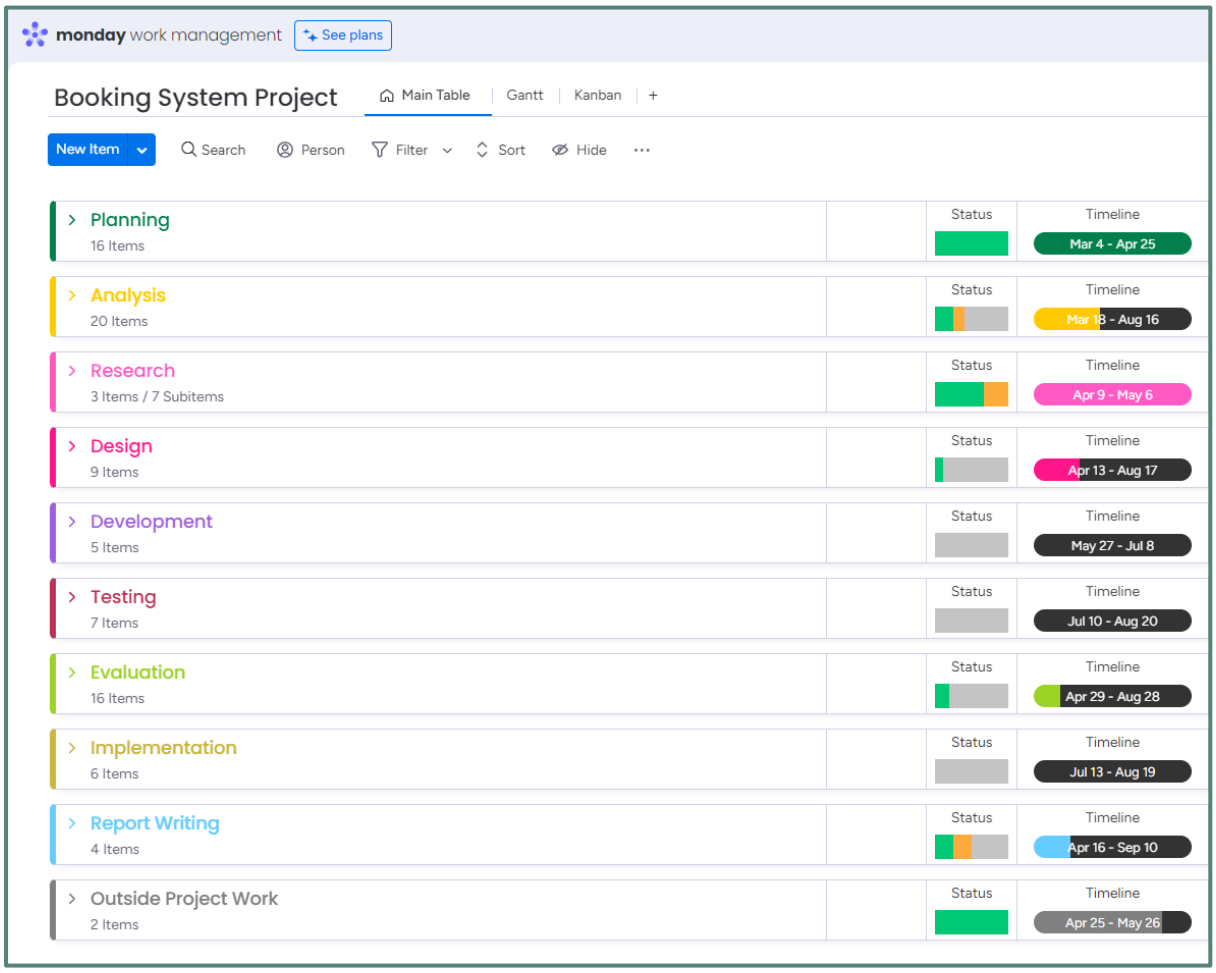

Figure 1: Using Monday.com to schedule tasks and subtasks, categorised, colour coded and displaying start and end dates.

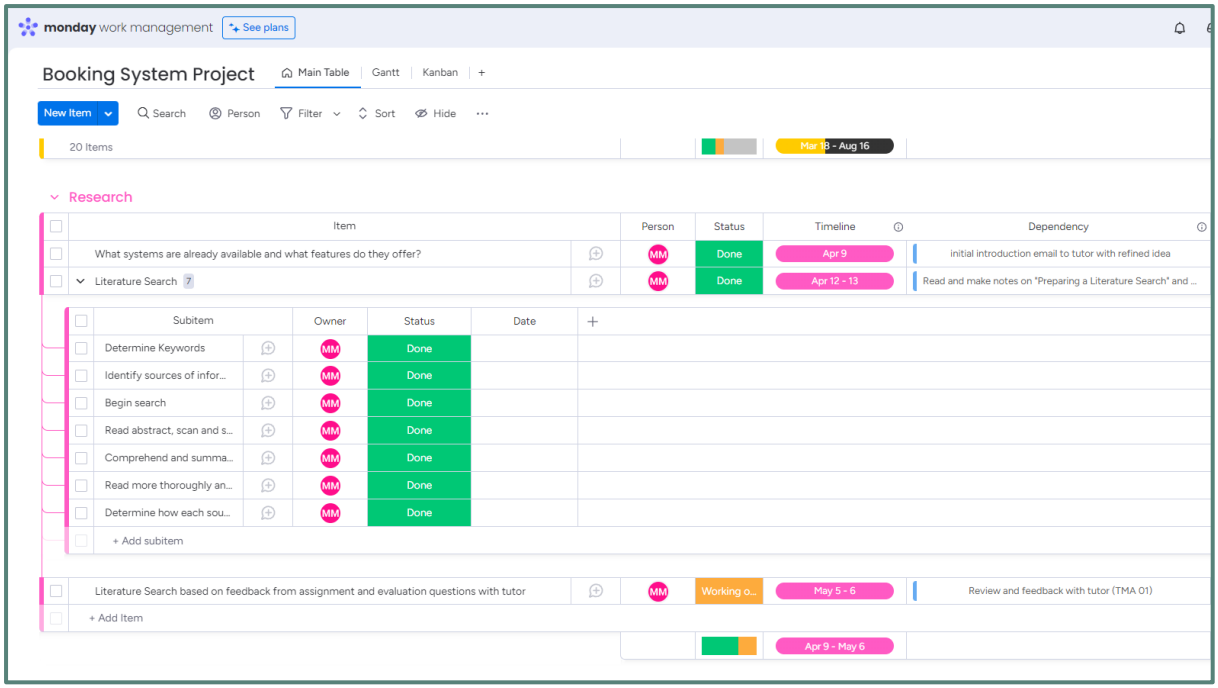

Figure 2: Research tasks being broken down into subtasks.

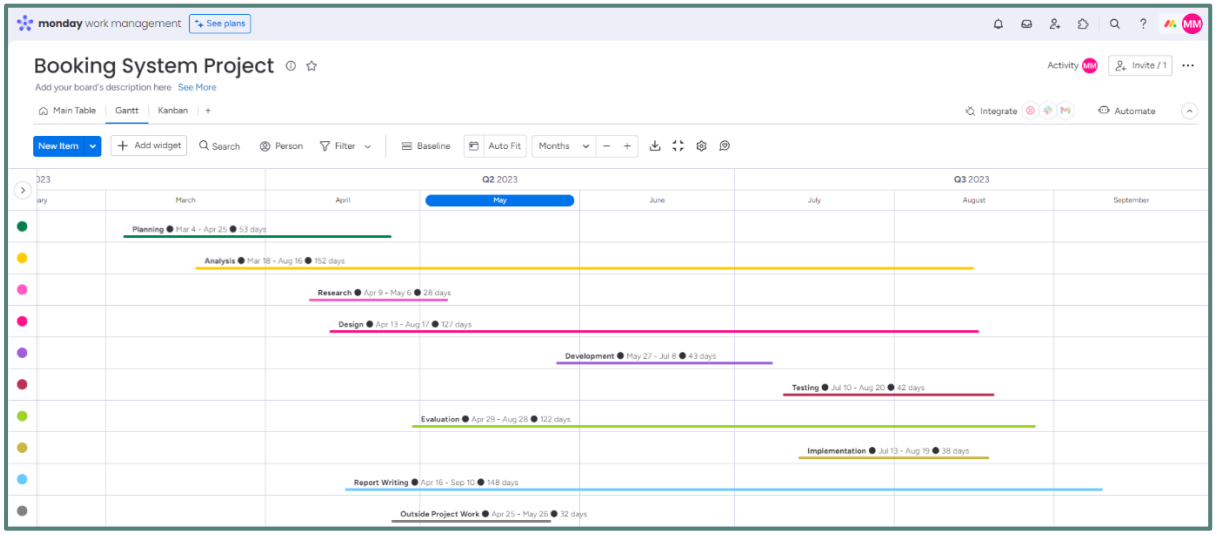

Figure 3: Gantt chart displaying full schedule timeline.

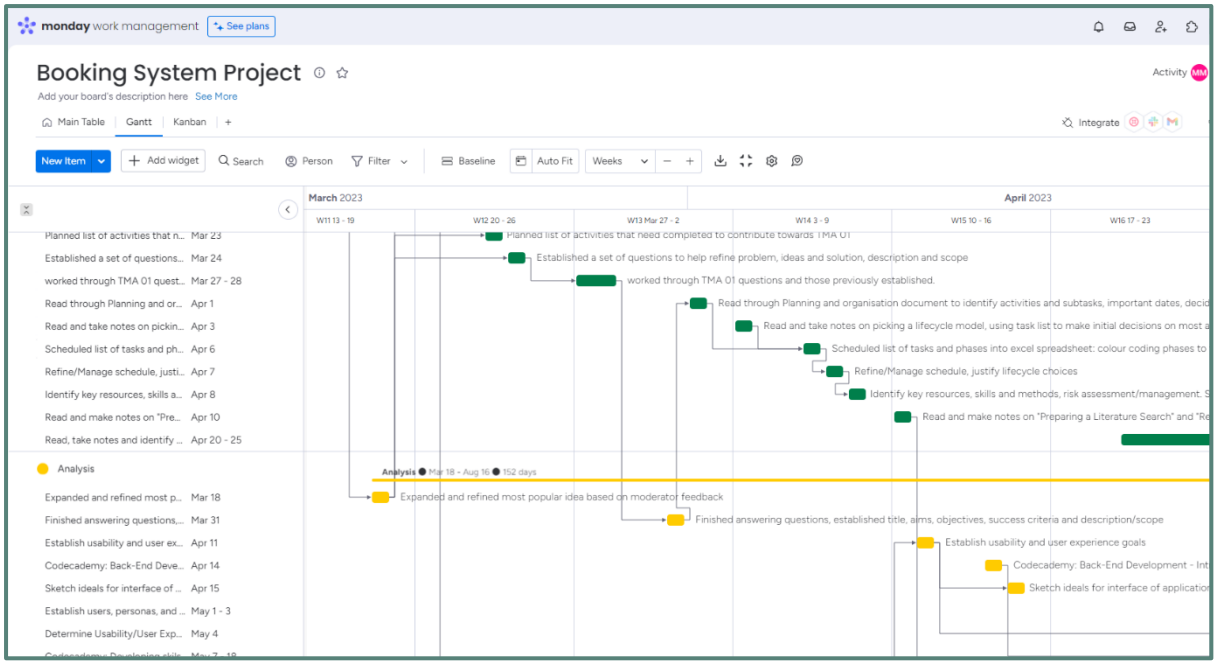

Figure 4: Gantt chart displaying dependencies of tasks.

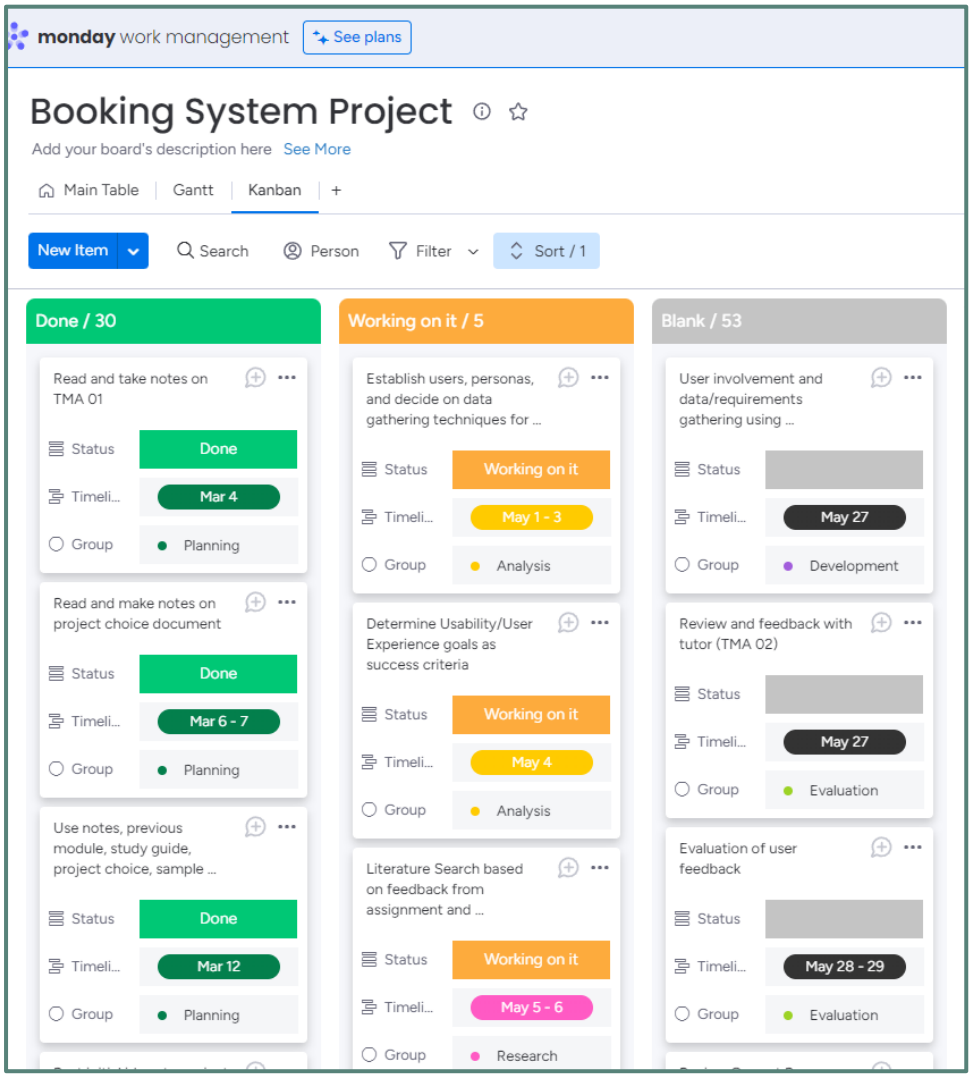

Figure 5: Kanban display of tasks arranged by completion status.

For a complete list of all tasks, please refer to appendix A/Booking System Gantt Chart PDF.

The project will follow a user-centred lifecycle approach with incremental prototyping: The system has a well-defined scope and boundaries; user involvement can feed back into the design at regular intervals using prototyping to clarify requirements, it provides an agile approach, taking advantage of its values/principles and the success criteria and essential functionality can be defined from the beginning through user involvement. The project is too large for a single developer and contains too much unfamiliarity within the domain to follow the classic or iterative waterfall method and the incremental scrum approach could have been applicable if not for the need to develop specific skills further; building APIs and databases, meaning it's not possible to quickly develop working prototypes.

Resources, Skills and Methods

Resource	Uses	Alternatives	Reason
Visual Studio Code	Writing and editing code	NetBeans: Overly complicated for project needs, prior experiences has been buggy/frustrating.	Familiarity, simplicity, source-code editing, debugging etc, GitBash terminal integration.
Microsoft Word	Writing notes and reports	None	Extensively used, provides required features for note and report writing.
Git	Version control	None	Simple to learn, industry standard, familiarity.
GitHub	Online repositories for backup and testing	None	Integrates with Git, collaboration and sharing, utilisation for testing purposes and backup.
StackEdit	Writing markdown documents	Notepad++: Plugins can be used to view side-by-side comparison, but features aren't as comprehensive or as intuitive as StackEdit.	Instant feedback on markdown display, incorporation of extensions, easy to use. Provides buttons for common styling options.
Codecademy	Database/API development	Udemy: Subscription not available and would have to purchase individual courses. Quality is based on user feedback not curation.	Provides ample educational materials, exercises, courses and environments. Already have subscription. Provides professional certification.
MDN Web Docs	HTML, CSS and JavaScript reference	W3Schools: Offers basic referencing of beginner concepts.	UpToDate comprehensive reference for all languages.
GOV website	GDPR, legal and data research	None	Legal requirements of UK data storage and transmission.
Material Design	Design using pattern languages	Apple design Resources: requires an all-Apple environment.	Design cohesive and common layouts across all devices.

Skill	Reason
Project & Time management	Ensure timely application of resources, set targets, expectations, goals, consistent and productive workflow, define responsibilities and achieve aims and objectives.
HTML	Define application content.
CSS	Define application style utilising media queries for a range of devices.
JavaScript	Interactivity between users, the system and the APIs/Database.
SQL	Managing requests to the database.
Local and remote repositories	Version control, backups and branching workflows.
Chrome Dev Tools	Testing responsiveness of different devices in browser.
Markdown	Providing project information for GitHub repository.
Interaction Design	Techniques and methods to develop and design a product and interface adhering to progressive user involvement and feedback.
Web Development	Application of knowledge and languages to produce a web application.
Literature Search & Review	Finding and evaluating resources relevant to my project that can be incorporated and extended upon.
API Design/Development	Implementation of interactivity between the front and back-end systems.
Database Design/Development	Defining appropriate tables and access to stored data.

Risk Management and Analysis

Chance & Impact:

- Low: Considered surprising if occurred/Has negligible effect, a delay of a few days
- Medium: Disappointing but not surprising if occurred/Considerable effect, still likely to succeed.
- High: Will likely occur/The project's success is threatened.

Methods addressing risks:

- Avoidance: Avoid the risk by eliminating it.
- Mitigation: Form an appropriate strategy to manage the problem.
- Acceptance: Accept the possible negative consequences of the risk.

Activity	Risk	Chance	Impact	Method	Plan	Contingency
Requirements Gathering	Incorrect and inappropriate techniques used to elicit information.	Low	High	Mitigation	Use resources to identify appropriate techniques and apply them in constructive ways.	Evaluate quality of data to determine its usability, re-evaluate techniques for future cycles, use personas and own judgement to elicit requirements from 'typical' tasks and tutor advice.
Determining Users	Inappropriate user selection.	Low	Medium	Mitigation	Use resources to establish user base from an administrative management perspective and contact various boardgame venues to request access and participation of specific user base.	Create personas built from research around reservation systems and use friends and family, ask for help using OU forums, and myself.
Prototyping	Misunderstanding, not listening & incorporating user feedback.	Low	Medium	Mitigation	Utilise and evaluate feedback from testing, identifying significant functionality and incorporating it in the next prototype.	Re-evaluate feedback, request tutor advice, focus and fully comprehend user feedback, contact user with follow up questions where appropriate.
Availability of Resources	Unavailability and retraction of user participation. Resources removed/put behind paywall.	Medium	Medium	Accept	Incorporate appropriate but substantial users from various sources and perspectives.	Utilising a range of resources from different locations ensures work can continue without too much interruption.
Technical Limitations	Equipment or knowledge not available.	Low	Medium	Mitigation	Use a range of resources to redirect project feasibility using alternative methods and equipment.	Have a range of alternative resources available and plan for alternative solutions, one where the project can't be completed as initially envisioned.
Quality Control	No testing, evaluation and analysis conducted.	Low	Medium	Mitigation	Plan appropriate phases into schedule.	Reschedule phases to incorporate into project.
Online Work	Weather or other factors disrupting connection.	Low	Medium	Accept	Continue to work as scheduled and back up work regularly.	Use local resources such as Git to backup and push to repositories when they become available. Use mobile data.

Project Planning	Over ambitious, scope too wide, lost quality.	High	Low	Mitigation	Collaborate with tutors and prep forums to iterate ideas	Iterate over multiple ideas and continued collaborate to refine ideas to appropriate level.
Project Management	Overestimated expectations, no schedule, poor balance between quality, productivity and time.	Medium	Low	Mitigation	Schedule activities, regular evaluation, define scope, aims, objectives, success criteria.	Evaluate realistic availability of time, resources and expectations.
Development	Poor quality code, inadequate documentation, product doesn't work.	Medium	Low	Accept	Use knowledge and skills to the best of ability.	If success criteria not met, evaluate why, what happened and what could have been done better.
Research	Irrelevant material included.	Medium	Low	Mitigate	Use literature search plan using critical reading techniques, PROMPT, and active reading.	Re-evaluate published research relevance, its connection to my project's techniques and concepts of interaction design. Differentiate between technical and literature sources.

Project revisions include considerable consolidation of ideas, clarification of the organisations size, focusing more on user interactivity and functionality of the system, disregarding the design of the website from a customer POV. The APIs called will be the same for the system and website so a simple site with input forms can be utilised for testing purposes. Scheduling software using Monday.com has replaced excel to provide dynamic adjustment of dates and various view such as Gantt and Kanban to keep track of progress. Alternatives to tools have been considered and reasons for choices included. Finally, contingency plans have been considered in risk analysis if plans to mitigate problems fail. This will ensure high probability of success despite any shortcomings and provides a thorough analysis of risks involved.

PROJECT WORK

Success Criteria

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 process of making a booking in the way they want?

Learnability: A top priority and 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?

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: 1 hour?

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.

The Design Context

System Users

For operation of the system user 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 at a basic level 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.

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 environment for the system will vary with each venue and will influence how functionality will need to be incorporated, for example, Is the system used at the entrance to check-in guests or only used in back of house? Are they using a fixed desktop, laptop or tablet? Is the system easily accessible by users? Is there a waiting area and queue? Does customer traffic cause distractions? Does the volume make it difficult to carry out tasks? How is the operation of tasks different during peak hours?

To help answer these questions and identify, understand and support potential users, stakeholders and gather requirements, it is essential to understand how task are caried out in the domain in which they will be employed. A number of local venues have been identified with the intention to visit each barring permission and employ the experience of current system users to extract essential functionality and best practices.

Initial data gathering will be conducted by onsite premises visits to interview users individually and answer a series of questions and demonstrate their preferred sequence of steps in a variety of tasks. The purpose is to identify the ways in which users interact with the system in each environment, evaluate and analyse the results and decipher how these can be applied and adapted for use in my project. Only handwritten notes and sketches will be used to record answers and task steps as opposed to video or photos to ensure personal data is not compromised, however photos of the environment will be used where appropriate. A list of all venues identified, and interview questions can be found in Appendix B. The participant information sheet and consent form can also be found in the appendix folder.

Initial Design Sketches

A variety of sketched design ideas for initial consideration and feedback from potential users.

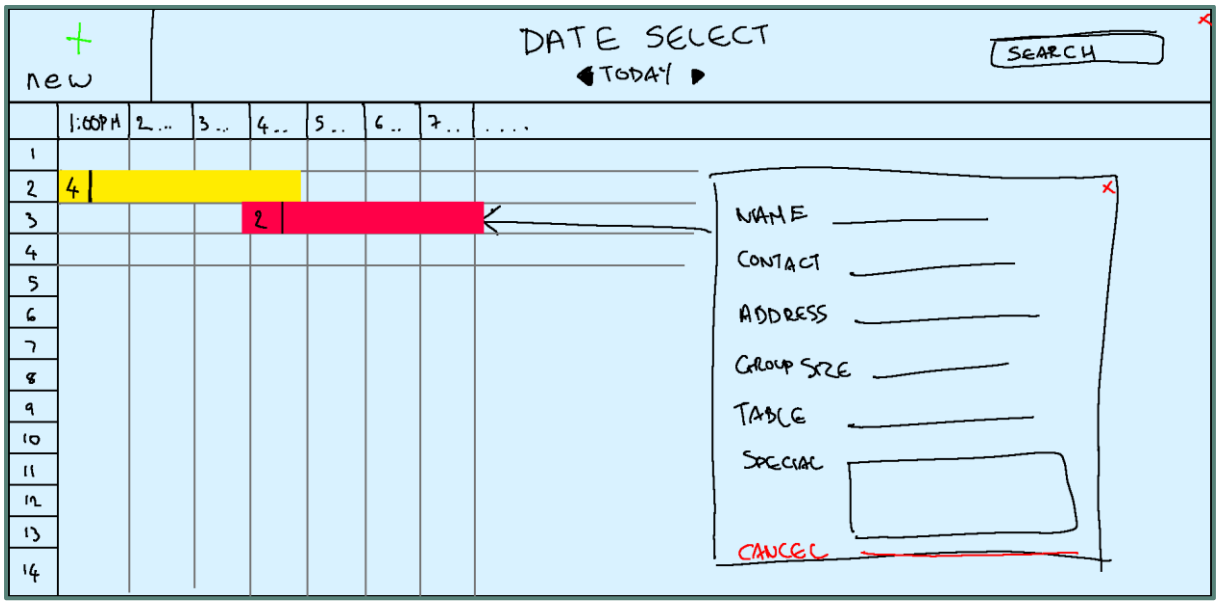

Figure 6: Horizontal design of booking system admin management console.

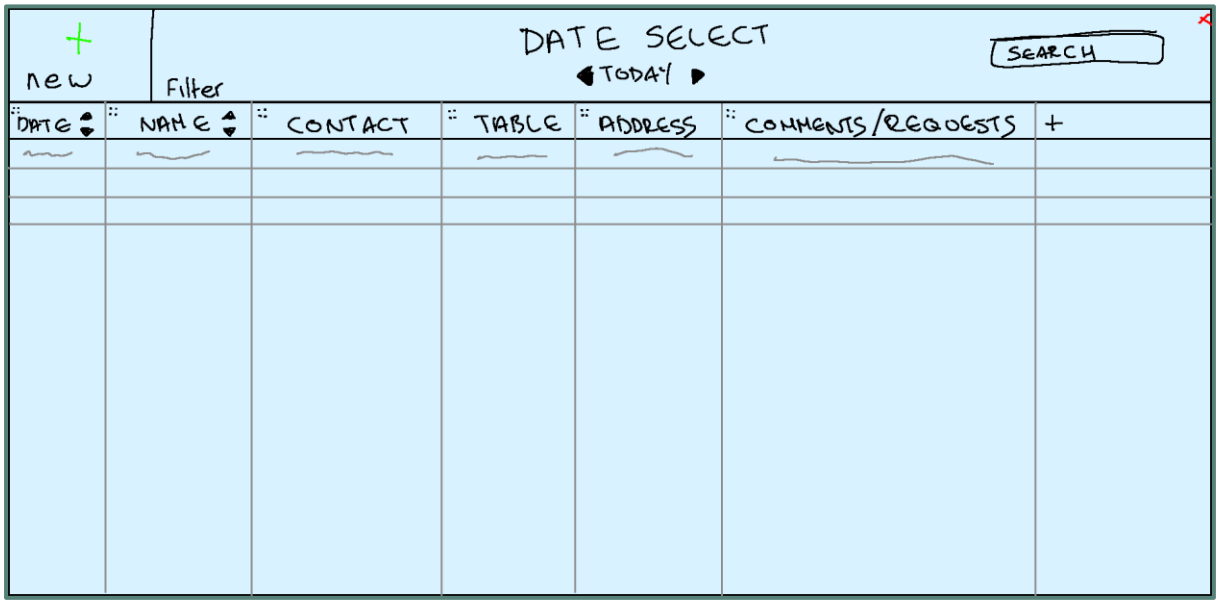

Figure 7: Table design of booking system admin management console.

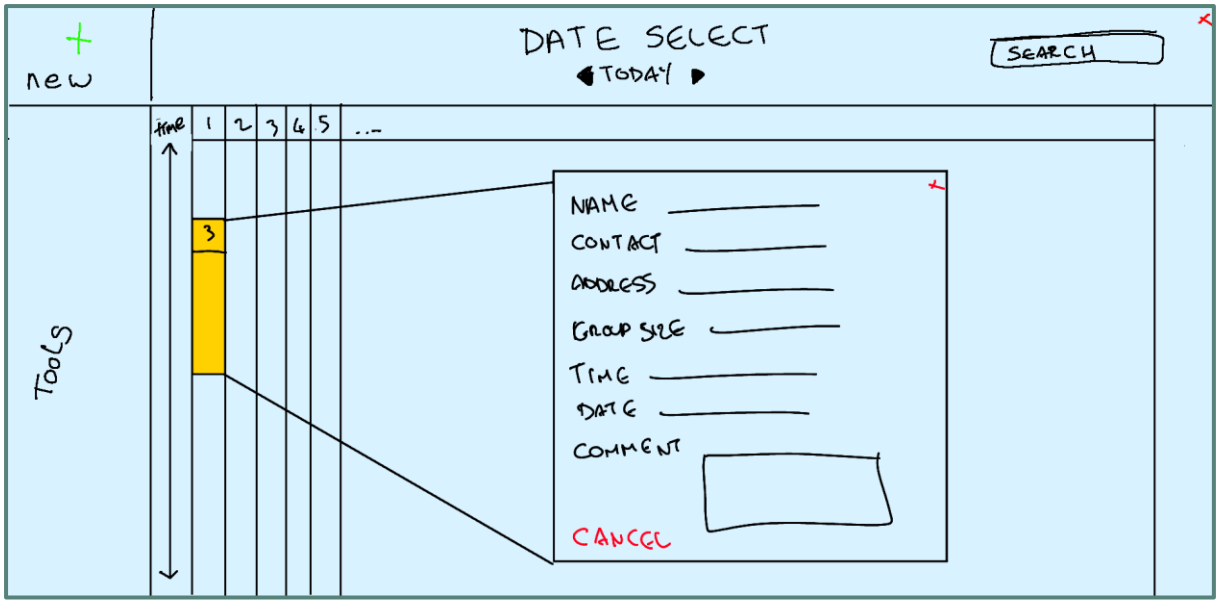

Figure 8: Vertical design of booking system admin management console.

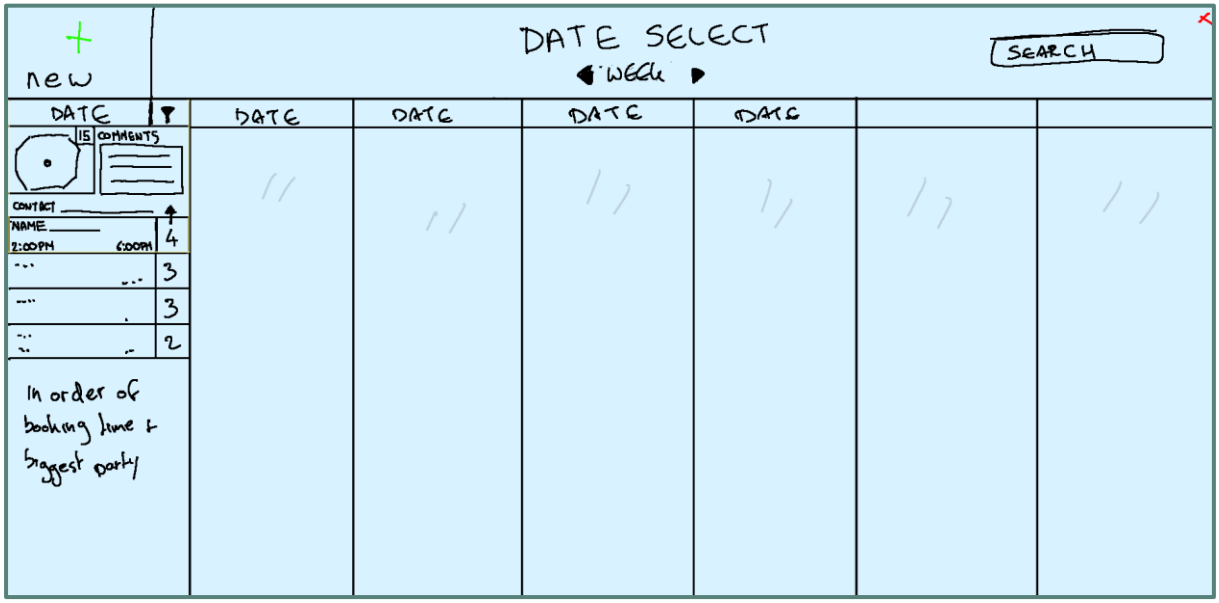

Figure 9: Card design of booking system admin management console.

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.

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

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

Christopher Alexander first proposed the idea of patters to abstract a recognisable quality and apply it other designs. However, pattern languages can be more powerful, albeit less common, as they incorporate a network of patterns that references each other to create a complete structure. (Preece, et al., 2015)A good example of this is Material Design: an adaptable system of guidelines, components and tools to support best practices of user interface design.

REST API Development with Node.js: Manage and Understand the Full Capabilities of Successful REST Development 2nd 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 to be 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.

A full list of researched articles, journals and conferences technical documents including textbooks and video courses can be found in Appendix C.

Review and Reflection

The time between subsequent assignment hasn't gone entirely to my schedule, however I have made adjustments to ensure that each task can still be completed within a reasonable amount of time. I spent more time trying to consolidate and refine the scope of the project to try and understand its potential users, goals, outcomes and any issues that could occur with contingency plans, than conducting more practical work in this initial period. However, I do feel with the feedback from the previous TMA, and taking time to refine and understand the scope and boundaries of the project I am in a better position going forward with tackling practical components and gathering requirements so I can iterate and start analysis and evaluation phases before beginning implementing functionality.

I am still continuing to develop my skills and have attempted a very small amount of coding to implement the effect of media queries and responsive design using GitHub pages. I still need to do a second literature search exploring topics such experiential and reflective cognition, emotional aspects of technology interaction, implementation of remote data gathering techniques and analysis/evaluation techniques for interactive products. I am feeling more confident in the direction the project is going, its scope and what I need to do to achieve the results I am aiming for.

LSEPIs and EDI

There are a number of people who will be affected by the development starting with:

1. Customers: Legal issues would revolve around the storage and secure transmission of customer details, GDPR, etc.
2. Administrative personnel: They will need to be trained on GDPR so they can comply with legislation. Harmful effects as a consequence of the system is the ability to check logs for responsibility of mistakes.
3. Staff of the venue – The success of the software directly affects front of house staff by lessening the frequency of dealing with customer complaints from the mismanagement of reservations.
4. Managers: The responsibility of who was at fault is easier to narrow down in the event of problems as logs can be checked (time/date booking was made, any modifications made to the booking and by who at what time/date, etc) and situations can be managed accordingly and more appropriately.
5. Business owner: legal issues could be applicable such as the negotiation of contacted outputs and deliverables, cost of potential lawyers, legal ramifications of failing to deliver, etc.
6. Web developers: Unless contracted under the same company, web and software developers would need to collaborate together to modify the website and integrate the web application into the structure of the current website. This benefits the web developers by providing them with work and revenue but could provide harm if agreements and decisions cause disparity between designers and fails to provide integration and software that meets the users and stakeholders needs.
7. Software designers/developers: Similar to above, provides work, revenue, and opportunity to gain and maintain a potential new client for further development in the future. Harm could arise from exorbitant client expectations causing a conflict and damaging reputations.
8. IT and telecoms contractors/staff: Would be responsible for working with the software developers and website developers to agree on the best integration approach based on

current infrastructure (on-site or cloud services, new equipment, responsibilities, who has access to what services and data? etc).

9. Competitors: other booking systems wouldn't see much if any impact from a bespoke system being developed unless it expanded into a commercial product later on.

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

While my project shouldn't have any bias towards any specific group, I think just simply being aware that everyone is different and doing my best to understand those differences, how they affect people in different ways and always being aware of them will help develop competency in equality diversity and inclusion. Even systems that use machine learning can evolve to replicate social bias evidencing attention to equality, diversity and testing for inadvertent bias needs to be built into the whole process.

During involvement of participants I will comply with the ethical principles set out in the code of professional conduct for usability practitioners (Association, 2023), the CAN Code of ethics and professional conduct (Machinery, 2023) to not discriminate against anyone due to any differences.

REFERENCES

Association, U. E. P., 2023. *UXPA Code of Professional Conduct*. [Online]
Available at: https://uxpa.org/uxpa-code-of-professional-conduct/
[Accessed 21 05 2023].

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.

Machinery, A. o. C., 2023. *Code of Ethics*. [Online]
Available at: https://www.acm.org/code-of-ethics
[Accessed 21 May 2023].

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.