



An Improved UoP Sports App

By

Theodore Uchechukwu Ikem

Project unit: PJE40

Supervisor: Matthew Poole

May 2023

Abstract

The project aims to evaluate, redesign, and make improvements to the user interface design and experience of the current UoP Sports application. It would document the process of attempting to develop an improved application focusing mainly on Android devices. An analysis would be conducted to ensure that the project is steered in the right direction. Several methodologies were researched and critiqued, and the project would use the iterative methodology. The project artefact is developed and evaluated against the requirements, and then concluded by summarising findings, reflecting on the project, and discussing future work.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Matthew Poole, for providing valuable guidance and feedback and allocating time for support throughout this project.

I would also like to thank my lecturers at the university for their constant support throughout the years to help equip me with the necessary knowledge and tools to take on this project.

Finally, I want to thank my family and friends for their support and encouragement during this project.

Table of contents

Abstract.....	1
Acknowledgments.....	3
1 Introduction.....	5
1.1 Background.....	5
1.2 Aim.....	6
1.3 Document Overview.....	6
2 Literature Review.....	8
2.1 Booking systems and design.....	8
2.2 User Experience Design Principles.....	9
2.2 Interface Design.....	10
2.4 Similar applications.....	11
2.5 Summary.....	13
3 Methodology.....	14
3.1 Waterfall.....	14
3.2 “XP” Extreme Programming.....	14
3.3 Iterative development.....	15
3.4 Chosen Methodology.....	15
4 Requirements.....	16
4.1 Requirements Elicitation.....	16
4.2 Functional Requirements.....	19
4.3 Non-Functional Requirements.....	21
5 Design.....	22
5.1 Use Case.....	22
5.2 Architecture.....	24
5.3 Database.....	26
5.4 UI Design.....	26
6 Implementation and Testing.....	30
6.1 Iteration 1.....	30
6.2 Iteration 2.....	35
6.3 Iteration 3.....	41
7 Evaluation.....	47
8 Conclusion.....	49
8.1 Future Work.....	49
8.2 Reflection.....	50
References.....	52
Appendices.....	56
Appendix A: Project Initiation Document.....	56
Appendix B: Gantt chart.....	61
Appendix C: Ethics Form.....	62
Appendix D: Use Case Tables.....	63

1 Introduction

This project (the Sports App) is an evaluation, redesign, and improvement of the user interface design and experience of the existing UoP Sports application. The aim of the document is to communicate the journey and details of the development of the mobile application.

This chapter is to provide an understanding of and expand on the idea behind the project. We begin in Section 1.1 by covering the effectiveness of good UI/UX design in the success of mobile application development and also an evaluation of the existing application that triggered the idea for this project. Section 1.2 describes what the project hopes to achieve at the end and the steps that would be taken to achieve the desired outcome, and Section 1.2.1 would look into the risks and constraints of this project, taking into consideration the timescale and the proficiency in mobile development. Lastly, Section 1.3 covers a brief summary of the chapters and their organisation in the report document.

1.1 Background

The UoP sports application can be easily said to be a day-to-day application as several users use the sports centre on a daily basis or multiple times a day. For these reasons, the application is expected to have features and a great design that grab the attention of the users and, overall, provide a great experience while interacting with the application. This enables users to take advantage of the functionality and services it offers. There needs to be a balance between functionality, user experience, and interface design for the application to be successful among the users.

In connection with the sports centre, it is quite important to note that a great application that appeals to the users can be a huge support to the business, and this is dependent on the number of users that actually download and use the application on a daily basis. It is important that some investments are made to ensure that the application has a great interface design and is in a very usable condition, as this tends to increase customer satisfaction and retention.

The existing application does not have an entirely unattractive user interface design or is unusable, but there are some aspects of the application that may seem uneasy, redundant, and difficult to understand. An evaluation of the application would be detailed in the **requirements** chapter.

1.2 Aim

This project attempts to produce a mobile application (for Android) that implements an effective and easy user interface design and experience and can be adopted and built upon by the UoP Sports tech department. It focuses on, but is not limited to, the structure, design, and experience, as well as some of the functionalities like booking services and account management. Research would be done on the importance and principles of UI/UX design in mobile application development to ensure that the implementations in this project should be evaluated against the existing application at the commencement of the project, and a conclusion would be drawn to discuss the implemented features and any potential evolution for future works.

There were some risks and constraints considered in this project. Some features and requirements may be met partially or not at all as a result of the amount of time allocated for this project.

The proficiency of the programming language that will be used to code this project is not very high as the journey was a “learn as you code” experience; however, it was a speedy learning process and the foundations to build the application were sorted. All the various sections of the software development were completed by myself, including the front-end and back-end development, as well as the model and data management (database).

1.3 Document Overview

Chapter 2 includes a narrative literature review on the topic relevant to booking systems and the importance of User Interface and User experience in mobile application development.

Chapter 3 describes the methodology used for the development and management of the project. It covers the various methodologies, justifies the chosen one, and describes the implementation of the methodology used.

Chapter 4 outlines the requirements for the application, which include functional and non-functional requirements. It also discusses the process of requirement elicitation and the techniques used to gather requirements.

Chapter 5 provides an understanding of the system's construction through the architecture design. It presents the design of the mobile application and describes the design decisions and rationale behind them.

Chapter 6 discusses the implementation of the mobile application, including the development tools and technologies used. It covers the decisions made and problems faced during the implementation process, as well as the testing approaches.

Chapter 7 evaluates the mobile application requirements and discusses to what level they were met. It also identifies areas for improvement and makes recommendations for future work.

Chapter 8 concludes the entire report, summarising the findings and contributions of the project. It discusses the extent to which the aims and objectives were met, future work, and potential areas of research and personal development.

2 Literature Review

This literature review scrutinises the research that has been done concerning booking systems, user interface design, and user experience in mobile application development. Section 2.1 looks into the booking system, the design, and the features of good booking systems. Section 2.2 discusses the user experience, its design principles, good examples, challenges, and importance. Section 2.3 discusses interface design, its importance, examples, and the challenges faced. Section 2.4 elaborates on similar applications, examining their user interfaces and the features implemented in them that make them standout. Section 2.5 summarises the research. At the end of this chapter, there should be an understanding of some key design principles and their importance in mobile applications.

2.1 Booking systems and design

In today's society, people are on the go, and appointments are scheduled, modified, confirmed, and even cancelled at a quick rate; some require patience and a cost (Akshay et al., 2019). Due to the increasing number of people using a service and everything being online, the need for an online booking system arises in order to control and limit access to that particular service and also allow users to make appointments with ease (Pişirgen & Peker, 2021).

Good booking systems are required to have certain features to ensure that they work well and that their purpose is achieved successfully. The most important is a smooth and clear user interface; the system should be pleasant to the user and as painless as possible to interact with (Wigley, n.d.). This can be done through the use of calendars to allow users to easily check availability (Muniz, 2017). The date selection should be easy in order to allow users to easily select their preferred date and time, as the difficulty of this may jeopardise the service. If included in the system, the payment system should be very secure, simple, and straightforward. It can integrate popular payment systems like PayPal, Apple Pay, and Google Pay (Muniz, 2017). For some, it should include a confirmation email so as to inform users that the reservation has been received by the service and confirmed.

There are some challenges with the interface design of booking systems. According to Pumpkin Web Design (2020), some of the several design issues include responsive design, which entails the way they appear on different devices of different sizes. A non-responsive booking system could have problems like slow loading times, incorrect images rendered,

buttons that appear too small to interact with, and text that is either too large or too small for the screen displayed on. Another challenge is insufficient feedback. In some booking systems, the user does not get immediate feedback, and as such, they are not sure if the action they took was registered correctly or not (Garcia, 2022).

Finally, a good booking system ought to have a simple user interface, a simple date selection process, safe payment methods, and the ability to send confirmation emails. Investing in developing and improving online booking systems is important to ensure the needs of the users are met by the service providers.

2.2 User Experience Design Principles

User experience can be seen from different perspectives; as a phenomenon, as a field of study, and as a practice. Essentially, "user experience" can be said to be how mobile applications perform in terms of their functionality. According to Joo (2017), "user experience is to improve user satisfaction, ease of use, deep understanding of users, and overall experience related to the emotions/thoughts, reactions, and behaviours that the users go through while using a particular system". Although aesthetic attractiveness often yields positive effects on the perceived usability of the product, this is the reason why user interface design and user experience go hand in hand (Nash, 2017). It carries the intention of making the interaction between the users and the system smooth and more natural and enabling the use of the application to be enjoyable in the long run (Sandesara, 2022).

Usability testing and user-centred design (UCD) are two examples of good user experience design principles. To ensure that the final product meets the users' needs and expectations, UCD involves putting the users at the centre of the design process by including them in the design and testing phases (Norman & Nielsen, 2021). The process of usability testing, on the other hand, involves watching users interact with the product to spot any usability problems and make corrections (Nielsen, 2012).

Popular applications like Instagram, Airbnb, and Dropbox are examples of applications with good UX. These programmes were crafted to meet the unique needs of their users and have an easy-to-use interface and clear design. The difficulties in UX design, however, include addressing the needs of various user groups, taking into account various cultural backgrounds, and making a design that is accessible to users with disabilities (Aljaroodi et al., 2022).

This user experience is important for many reasons. Firstly, it is important because it aims to provide positive experiences and build a better user satisfaction-conversion-retention

journey for the product (Gangadharan, 2019). Furthermore, it is important because it supports platform-specific designs, coherence and continuity, and a straightforward experience (Swain, 2020).

2.2 Interface Design

According to Hosamani (2022), “user interface refers to the concept of bringing together interaction designs, visual design, and information architecture”. It can also be described as “the way a system appears to the users, along with the visual design” (Sandesara, 2022). According to the research (Sakdulyatham et al., 2017), the new UIs can save time that the elderly use to achieve most of their activities, and regardless of age, most users prefer a system of higher usability rather than attractive decisions. Younger people place a higher value on greater time efficiency (Hosamani, 2017).

The best way to decide the suitability of a user interface is to consider the end users and the context of the mobile application. Zhang et al. (2009) define context as “any real-time information or entity that influences the interaction between mobile device users and mobile applications, as well as potentially changing the preferred behaviour of those applications,” and to determine this context, the user base needs to be discretely defined because the best interfaces are the ones with a close relationship to their context (Sandesara et al., 2022). In recent times, the aesthetics of any mobile application have been defined by the various frameworks on which they are built. According to Blair-Early and Zender (2008), the most common examples of desirable end-user experiences are often said to be “easy-to-use” and “intuitive.” Although these examples pose a problem, they are vague about the final experience and provide little to no indication of how that would be achieved.

A number of interrelated and important principles are taken into account when talking about interface design, such as the visual hierarchy, which is the idea that elements should be arranged according to priority (Singh, 2023). This makes it possible for elements to be strategically placed and messages to be conveyed effectively. Another is the layout; a strong layout design supports an interface's visual organisation. The ability to highlight significant content on an interface may be irrelevant to navigation but is helpful (Moreno, 2020). Last but not least, typography and iconography. Typography deals with the wise selection of fonts to improve an interface's usability and readability (Sapiro, 2020), whereas iconography uses a visual element to represent features or content (Katunzi, 2021). The icons should be straightforward, dependable, and user-friendly for an effective interface design (Katunzi, 2021).

An efficient interface can significantly improve the overall user experience by attracting new users and retaining existing ones (White, 2022). It enables effective user-application communication and guarantees that users' needs and expectations are met visually (Indeed Editorial Team, 2020). The application's success and the company's overall success are of utmost importance (Grace, 2022).

2.4 Similar applications

Pure Gym App

The PureGym application is a free application that can be found and downloaded on the Google Play Store and the Apple App Store. According to Weston (2020), the main goal of the application was to enhance the user experience and provide a competitive advantage. It was also a great way to engage and motivate users who don't use the Pure Gym facilities.

The application has great features that are available to members of the establishment, such as

- Contactless entry. This feature consists of a personalised QR code for the user to use on the entry scanner and gain access to the gym (PureGym, 2023). According to other research, this feature would be beneficial to users if they forget their pin or access card and need to quickly enter the gym.
- Live attendance tracker. This feature shows users how busy the gym is and allows them to plan their visit appropriately (PureGym, 2023). This feature can also help to control the number of people in the gym to avoid congestion in the gym because users would not visit the gym if there were a certain number of people there and they could see so easily and quickly.
- Membership management. Members can easily and quickly manage their memberships from their fingertips. It can range from switching gyms to updating payment information (PureGym, 2023).
- Workout recommendations. The application offers great classes of over 600 workouts that can be done in the gym or at home (PureGym, 2023). There is also a feature to build personalised workouts with instructions to fit the user's goals.

There are many great features, but these are very impressive features, all designed with a great interface to allow the users to easily navigate through the application and also have fun with their workouts.

Book the room

It's easier to reserve a conference room with the help of this application. This will make the process simpler and help to prevent any potential real-time conflicts (Praveen et al., 2020). This is an application that was developed using Flutter and consists of simple and straightforward features.

Users of the app can reserve a conference room based on the time slots that are open. According to the flowchart below, the admin will be notified if the booking is for less than two hours and the indicated time slot is for more than two hours. The user would then need to wait for a response.

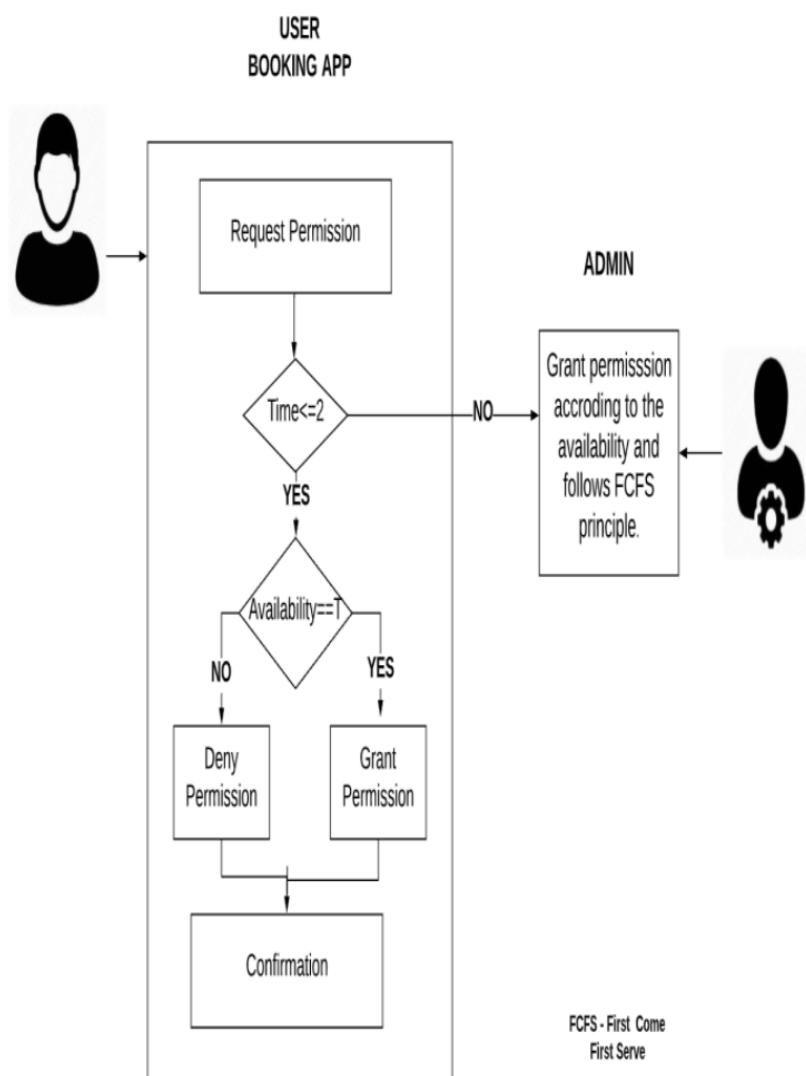


Figure 2.1: Workings of the application (Praveen et al., 2020)

Users of the application can quickly and easily find a date to reserve by using the calendar that is featured on the application's user interface (Praveen et al., 2020). The application has a clear and unambiguous user interface overall.

2.5 Summary

This body of literature examined booking systems and their designs, as well as user interface and experience design, including examples, challenges, and their significance. Analysing some related applications can help lay the groundwork for requirements. It is clear that effective user interface and user experience design have a significant impact on how well applications are used, as well as on user retention and business success in general.

3 Methodology

This chapter aims to compare software development methodologies and determine which is the best fit for this project and why. Sections 3.1 to 3.3 would look into the pros and cons of various software methodologies that were considered for management of this project, and Section 3.4 would briefly justify the choice of methodology for this project, weighing the pros and cons of the discussed methodologies and the implementation of the methodology.

3.1 Waterfall

One of the benefits of waterfall project management is that its goals are clear and concise, which results in a predetermined timetable and budget (Chandra, 2015). Another is that because the stages are not iterative, project management is simplified. As one task is completed before the start of the next, waterfall development saves a significant amount of time and resources while making it very simple to track progress.

On the other hand, this methodology can be disadvantageous, especially with large projects, because addressing unexpected bugs and errors can be difficult and time-consuming because the methodology is not flexible and earlier stages cannot be visited after completion. Also, due to the rigidity of the methodology, there is no guarantee of a working artefact until the last stage of the methodology, and completing a phase completely before moving to the next can prolong the project timeline (Hoory & Bottorff, 2021).

3.2 “XP” Extreme Programming

Extreme programming prioritises frequent releases and incorporates ongoing feedback loops to ensure the delivery of usable results quickly, which can be beneficial for getting quick feedback and executing changes on schedule (Kukhnavets, 2018). As it offers specific instructions on how to approach particular stages of the development process, XP is useful in instances where prescriptive approaches are preferred.

As opposed to this, XP relies on pair programming, which is impractical for this one-person project (Despa, 2014). Due to the extensive continuous testing, integration, and development changes required, it can be exceedingly time-consuming and demanding. Since XP involves a strong skill set and deep familiarity with the XP practises, due to the low level of programming language proficiency, XP may not be well suited for this project.

3.3 Iterative development

This methodology begins with simple implementations of a small set of requirements and further enhances several versions of the system until it is fully completed and ready to be deployed.

Some advantages are that it allows for the identification and resolution of bugs and errors during the early stages in order to prevent errors or defects from going into later stages of development, where they might be complex and costly to fix. It gives room for improvements and amendments, as there is an opportunity to gather feedback with every iteration. It saves time on documentation, and it is adaptable to changing requirements (Tutorials Point, 2019).

On the other hand, the incremental nature of this may result in some uncertainty about the final product until later in the iteration stages. There is also a high risk of some features being implemented incompletely or not at all; this can be a result of certain features depending on other features being completed in earlier iterations (Tutorials Point, 2019).

3.4 Chosen Methodology

Due to the nature of the project, variables such as the user's needs, the fact that it was a one-man developer project, and most crucially, the timeline and capacity for flexibility, were taken into account. Iterative development was found to be the best fit for this project after examining the various approaches and taking into account the need for adaptability and ongoing input. This is because, even though iterative development leaves the final product undetermined until later stages, this is not a major issue because, with just one person working on the project, it is simpler to monitor the progress and make changes as the project develops.

The requirements for the mobile application were first defined in order to apply this methodology. The method used to identify the features and functionalities that it must have in order to achieve its goals is demonstrated in Chapter 5. Priorities were assigned to the features, and after that, iterations were scheduled. Each iteration concentrated on certain features, which were then conceived, produced, and tested to find any problems and fix them. Until the application is finished, the entire process is repeated, incorporating feedback and making changes with each iteration.

4 Requirements

This chapter elaborates on the requirements for this project. Section 4.1 would look into the requirement elicitation processes, and Section 4.2 would look at the functional, non-functional, and user requirements, as well as the ranking of the requirements in order of priority (Must Haves, Should Haves, and Could Haves).

4.1 Requirements Elicitation

The nature of this project is an improvement of an existing application without any stakeholders or clients, and as such, self-reflection, brainstorming, user personas, interface analysis, user testing, and use cases and scenarios are the best methods to identify the needs and priorities of this project.

Interface analysis of the existing application: The existing user interface was examined, and the areas that could be improved or enhanced were identified. By doing this, the elements that are confusing to see or use and the opportunities for simplifying the user experience were identified. The application is not completely "unattractive," but some of the design choices are questionable. The application contains some redundant and underutilised features. These are considered below:

The user or staff login is not requested at the launch of the application; normal procedures would suggest that the login be requested at launch to track user activities and allow full access to the application.

One of the noticeable issues is that the app lacks proper organisation, with all the features and information placed on a single screen, as seen in Figure 4.1. This can make the app feel jagged and overwhelming for users. The absence of proper navigation further complicates the user experience. Ideally, the app should have distinct sections or tabs based on the category of tasks, allowing for easy navigation. The news and events block can also be optimised by displaying it on one page with a title and description. The app should also aim to minimise unnecessary scrolling and provide a more efficient user experience.

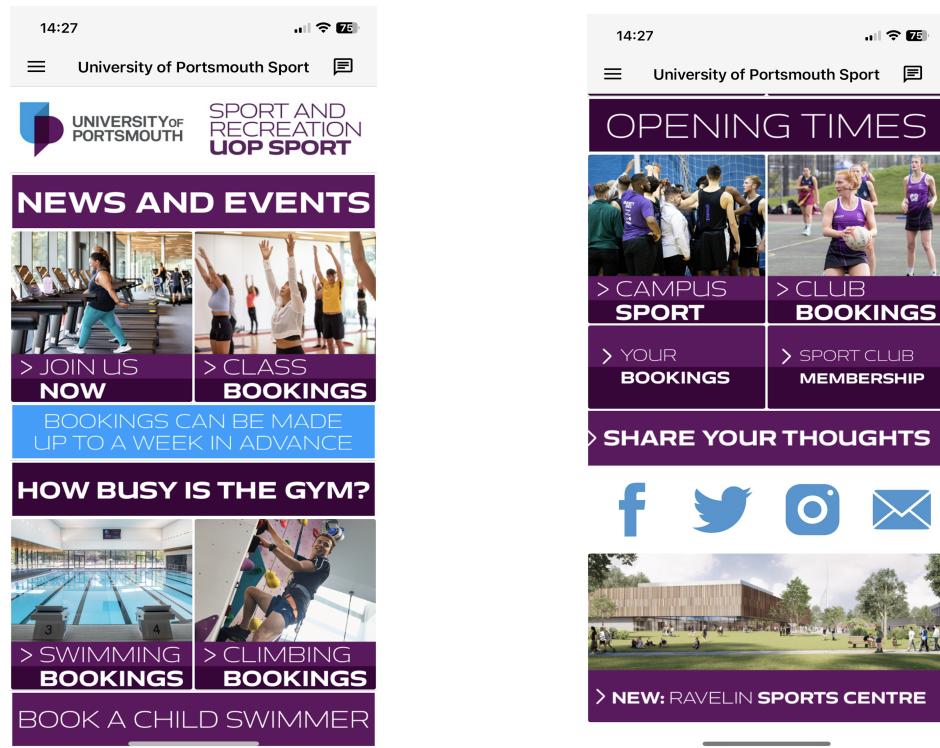


Figure 4.1: Front pages 1 and 2

In Figure 4.2, the navigation drawer also houses pages or tabs that are not necessarily pages, and such things are redundant as they are useless in the application.

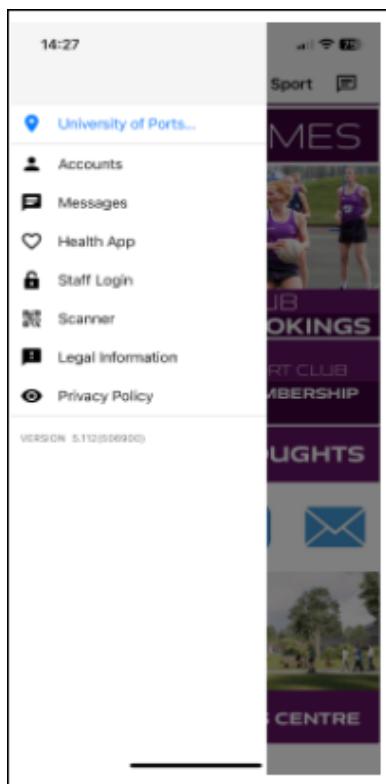


Figure 4.2: Navigation drawer



Figure 4.3: Account screen

In Figure 4.3, the account screen lacks sufficient information on the status of the user's membership, making it difficult for users to track their membership details or upgrade or downgrade their membership. The staff login tab should not be included in the dashboard, as it is irrelevant for regular members. Instead, it should be a separate login option specifically for gym staff.

From Figure 4.5, the scanner screen, about which little is known about the functionality, appears to be limited in use and could be replaced with a member QR code instead that can be used in place of the member bands and ID cards to get access through the gates.



Figure 4.5: Scanner screen

Gym Session				
Mon	Tue	Wed	Thu	Fri
24	25	26	27	28
18:45	18:45	18:45	18:45	18:45
19:00	19:00	19:00	19:00	19:00
19:15 77 sessions £5.00	19:15	19:15	19:15	19:15
19:30 78 sessions £5.00	19:30	19:30	19:30	19:30
19:45 42 sessions £5.00	19:45	19:45	19:45	19:45
20:00 79 sessions £5.00	20:00	20:00	20:00	20:00
20:15 77 sessions £5.00	20:15	20:15	20:15	20:15
20:30 87 sessions £5.00	20:30	20:30	20:30	20:30
20:45 81 sessions £5.00	20:45	20:45	20:45	20:45
21:00 118 sessions £5.00	21:00	21:00	21:00	21:00
21:15 101 sessions £5.00	21:15	21:15	21:15	21:15
21:30 120 sessions £5.15	21:30	21:30	21:30	21:30

Figure 4.6: Booking calendar

As seen in Figure 4.6, the booking calendar view is simple, informative, and helpful. However, the scroll between slots and overall interaction with the screen are shaky.

By analysing these features, requirements can be formed and improvements can be incorporated to enhance the UI/UX of the application and provide a smoother and more user-friendly experience for the users.

4.2 Functional Requirements

For this project and due to the timeframe, the requirements have been divided into priorities to make sure that by the deadline, the most important features are presented and the rest can be worked on at a later time.

4.2.1 Must Have Priority

MR-1 User Authentication: This feature is necessary as it enables the user's information to already be stored prior to making bookings and carrying out other tasks within the application, making the interaction with the application smoother as they wouldn't get into roadblocks of features that require them to sign in before use.

MR-2 Dedicated pages for features: Currently, the existing application has all the tasks and features clunked up on one page, as seen in Section 4.1. It would be important to have dedicated pages for the features and services so the users can quickly and easily locate them.

MR-3 Book multiple services: This feature is derived from the existing application; however, the interface design and experience may need some updating to make interaction seamless.

MR-4 Manipulation of Bookings: In the existing application, there are no options to edit the booking but only to cancel. In some cases, errors might have been made while booking, and a user might just need to make some microchanges and not go through the hassle of redoing the whole booking process.

MR-5 Updates and News Content: lucidly display the news, events, updates, and notices within the facility and centre as straightforwardly as possible. It is important for a user to see what is happening at the centre quickly prior to their visit. This is in the existing application; however, in 4.1, it is seen that it is unclear, there are many steps to do so, and some updates are hidden behind unrelated icons.

MR-6 Cloud Database: User, application, and booking data must be stored in the system's database. This was chosen since it can be easily accessed from any location with an internet connection and can be scaled up or down to match the application's demands. Strong security features are also provided.

4.2.2 Should Have Priority

SR-1 Membership Control: Granting users the liberty to amend and cancel their membership with the sports centre From some observations and user scenarios, it would be better for a user to have control over their account for a service they have paid for. The existing application has no information at all on the duration and expiration of the user's account, and this would be a good feature to have for more user retention.

SR-2 Notifications: Notifying the users of important changes within the facility and application They prompt the users on changes within the application in a short and simple manner.

The existing application has this feature implemented, and this is just a re-implementation of that same feature in a better way.

SR-3 Limit on bookings: Due to capacity limitations at the centre, users should be able to book only two slots per day. This enables effective decorum and the flow of service.

4.2.3 Could Have Priority

CR-1 Capacity Tracker: A page that displays an exact or estimated number of the current capacity of a service at the centre As seen in some similar applications, the capacity at the centre is displayed. This is to allow users to get a sense of how busy the area is and see when to go or not. It might not be a big thing, but it is a good quality-of-life feature that an application can have.

CR-2 Mobile payment feature: This allows users to make payment for their services within the application. Many applications now use Apple Pay or even Google Pay to allow users to pay for services with their mobile phones within the application. It could make payments easier for users.

CR-3 Workout and Food/Diet Recommendations, Plans, and Tracking: This is merely an additional feature that could be implemented in the application to help users with their fitness journey. It is like having a handheld personal trainer that can guide a user through workouts and dieting. This feature was inspired by other similar fitness and exercise applications.

4.3 Non-Functional Requirements

NF-1 Usability: the application must be designed in a manner that is easy to use, learn, and navigate, with clear and consistent user interfaces. This includes the simplicity of the design of the interface. The interface is expected to be designed in such a way that the users will be able to use it just from their interactions with other similar applications.

NF-2 Compatibility: The application must support a minimum of Android API 30 and also meet the Google Play requirements.

NF-3 Accessibility: The application must be accessible to users with disabilities. This includes minor implementations like the colour palette of the application to make sure it would be visible for colorblind users. Most of the other accessible options are provided by the operating system of the device, and the application would change accordingly to the user's device settings.

NF-4 Performance: This includes the application running smoothly relative to the device. It includes how quickly screens load up, how quickly data is sent and received from the database, and the overall responsiveness to user interaction.

NF-5 Legal and regulatory compliance: The application must comply with the relevant laws, which include the data protection laws, advertising standards, and accessibility guidelines.

NF-6 Security: The application must ensure that all user data is protected against unauthorised access or data breaches.

5 Design

This chapter describes the design layout of the system and also gives an insight into some of the design choices made. Section 5.1 would look into the use cases of the functionality in the application. Section 5.2 would look into the design of the system architecture. Section 5.3 would look into the structure of the database to describe how data is controlled and stored and the data model. Section 5.4 would discuss the choice of the theme for the application and consider accessibility options. Look into the various wireframes and discuss the reasoning behind their design.

5.1 Use Case

The use case diagram provides a high-level view of the application and helps to describe the different ways a user interacts with the application to carry out specific tasks. It shows the relationships between the user, the application, and the external environment that processes user requests. This is a useful tool to ensure that the application meets the requirements and supports the necessary functionality.

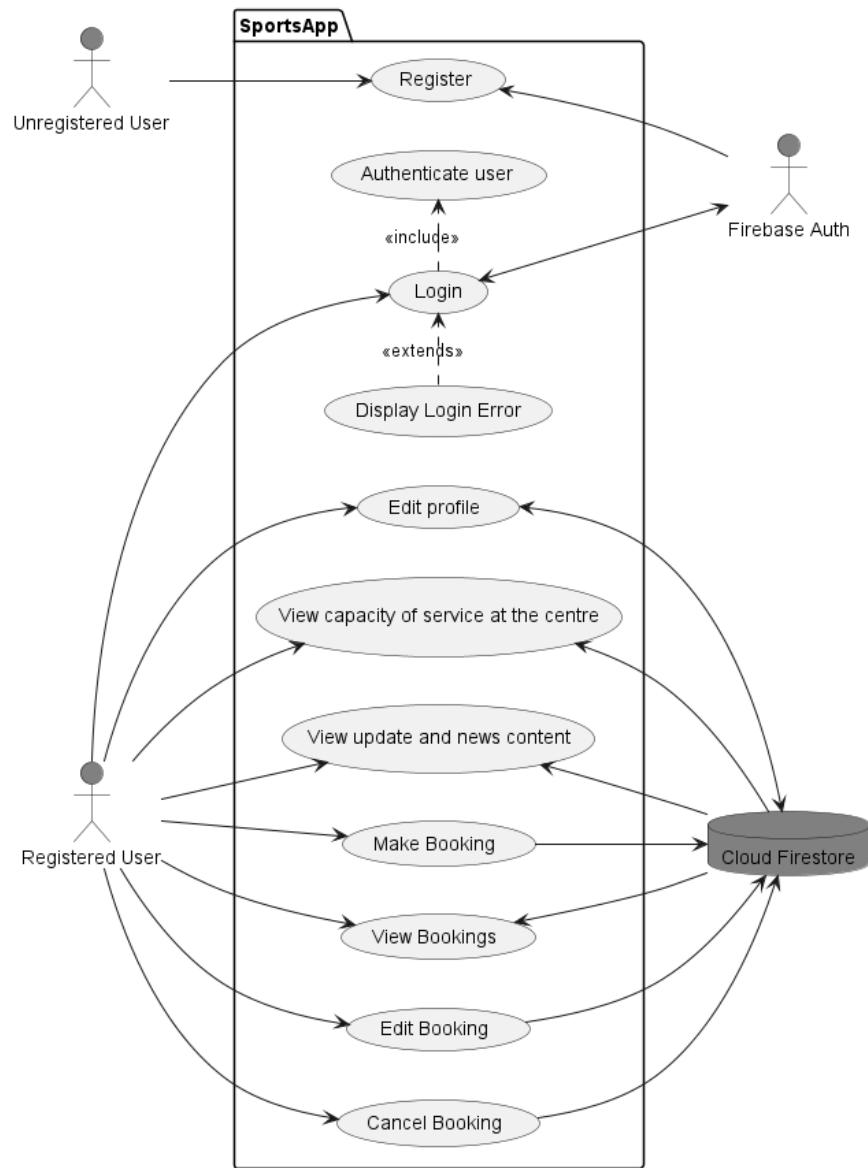
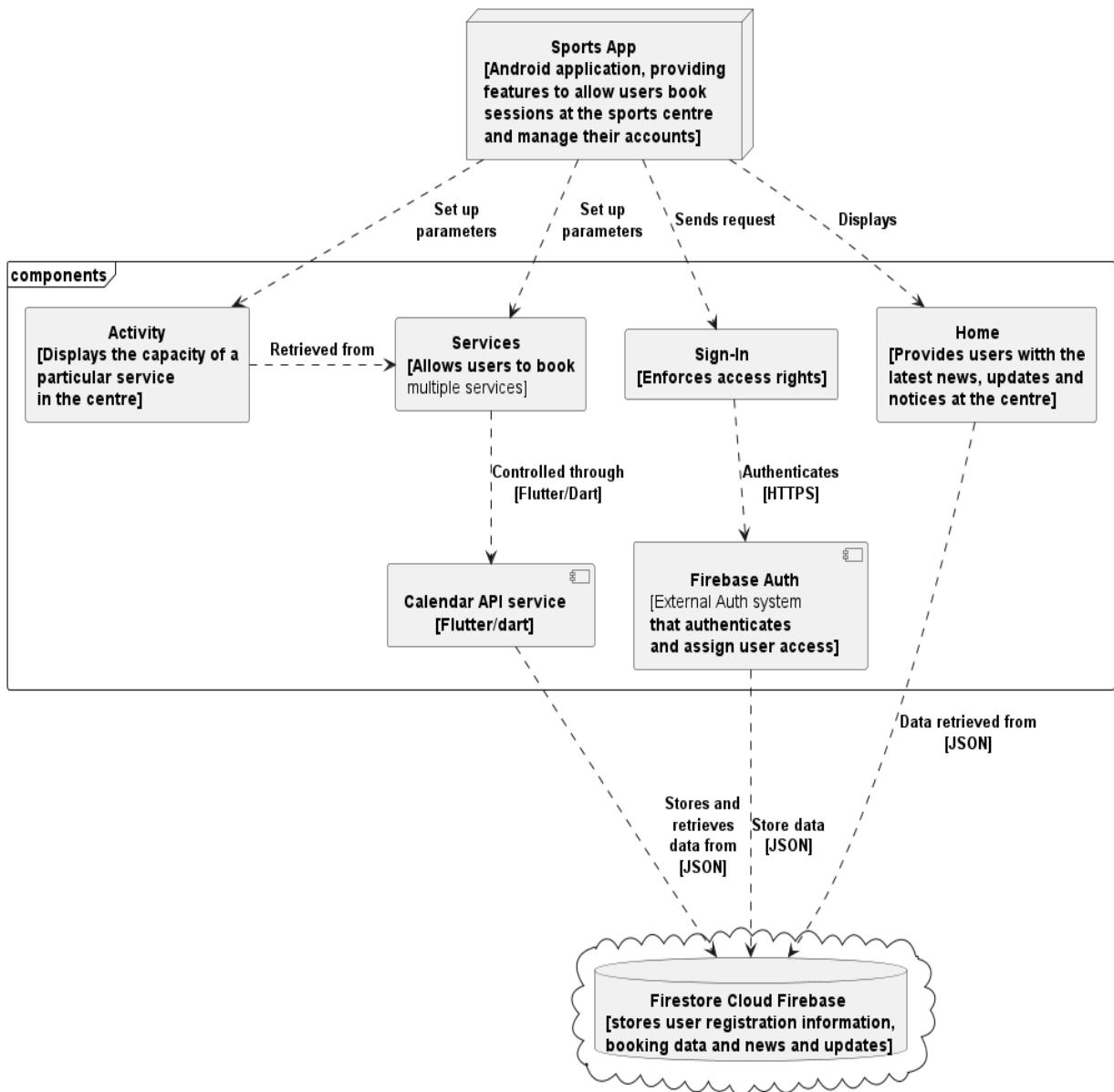


Figure 5.1: Sports app use case diagram

5.2 Architecture

The system is designed using a client-server architecture, which is made up of client-side and server-side components. The client-side includes screens like sign-in, home, services, activity, and settings. These screens give users the ability to complete tasks by interacting with the system. A secure server-side component called Firebase handles the user authentication; it takes care of the sign-in requests from the sign-in screen and guarantees this process is secure and dependable. Firebase also handles the storage of user information, app data, and booking-related information and ensures that data integrity is maintained and access to the database is controlled from the Firebase console.

The client-side component can retrieve data from the database and display it on the respective screens. This is done by submitting queries to the server-side component, which then retrieves the needed data from the database and sends it back to the client-side component for display. Overall, the client-server architecture provides efficient and reliable service for users by saving time and improving performance.

**Figure 5.2:** Client-server architecture

5.3 Database

For this mobile development project, the chosen database is the NoSQL Firebase Cloud Firestore, a fully managed NoSQL document-oriented database. NoSQL is chosen over traditional relational databases because it is horizontally scalable to handle more data and offers more flexible and dynamic data schemas.

Collections data model: there are 3 collections, namely; Users, Bookings, and News. Each contains documents, and each document contains fields with specific data types, as seen in Figure 5.3.

Users			Bookings		
PK	userId	String	PK	serviceId	String
	userName	String		serviceName	String
	userEmail	String		ServicePrice	number
				serviceDuration	number
				bookingStart	Date and Time
				bookingEnd	Date and Time
				userPhoneNumber	number
				userName	String
			FK	userId	String

News		
PK	Priority	number
	title	String
	description	String
	imageUrl	String

Figure 5.3: Collection data model

5.4 UI Design

This describes how the application could possibly look at the end of the final iteration, as well as how it would be interacted with. The goal is to categorise most of the features and rearrange the clustered items as seen in the existing application from Chapter 4.

5.4.1 Application Theme

The application is designed using the standard Material UI design for Android. This is a design that is common in many applications today, and it provides a familiar experience for the users. It has common elements, which are

- the app bar, which may contain the name of the screen or a search field.
- Bottom Navigation Bar, which is a quick way to navigate between screens of the application
- navigation drawer, which shows and hides some application navigations.

These are just some of the many components that make up the theme of the application. The icons used are also predefined material icons, this is to make sure that the users can decipher the meaning of the icon rather quickly.

5.4.2 Wireframes

This shows a simple visual blueprint that represents the basic structure, layout, and content of the application. It was used to map out the user interface and user experience of the digital product.

Each screen has a title in the top left corner of the screen in the app bar, and the bottom navigation bar is made to make switching between screens simple. The design is created in a way that makes the content and features organised and simple to find. If an application is attractive and easy to use, user interest will increase.

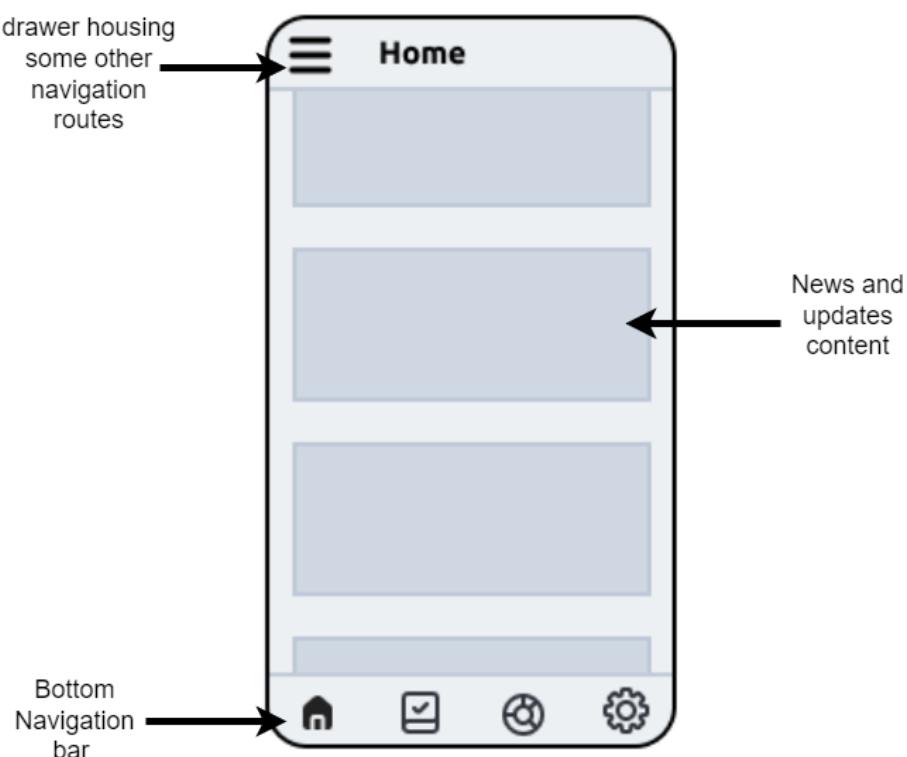


Figure 5.4: Home screen wireframe

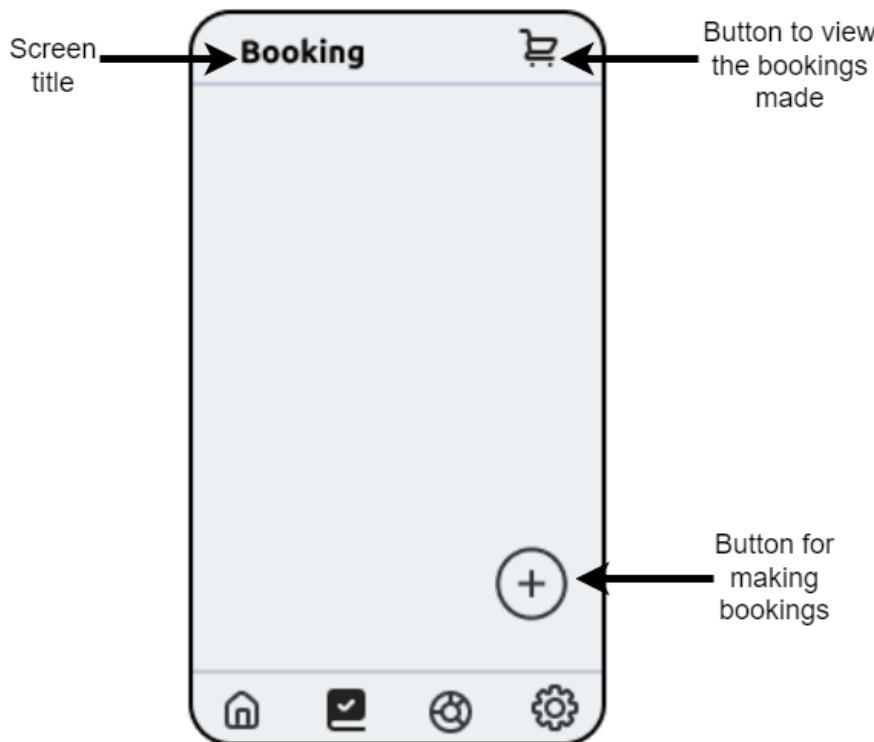


Figure 5.5: Booking screen wireframe

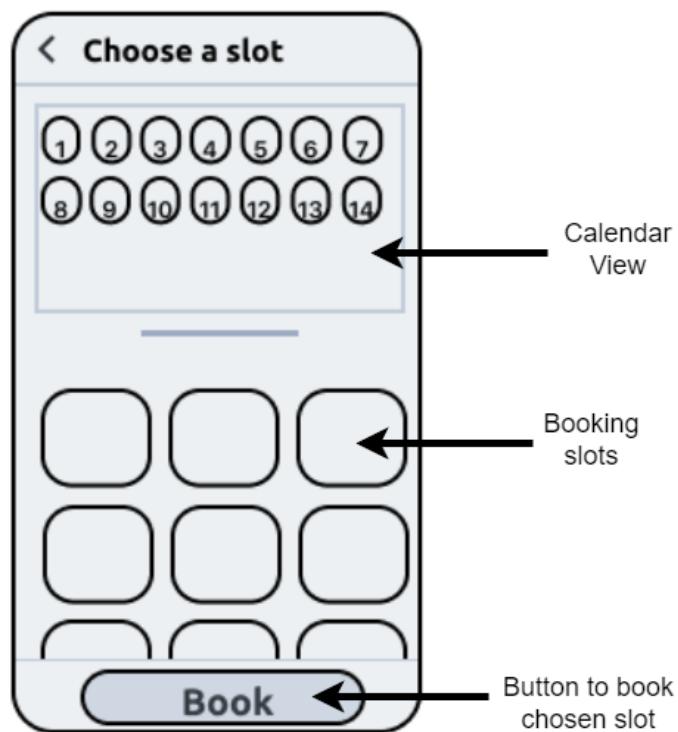


Figure 5.5: Booking calendar wireframe

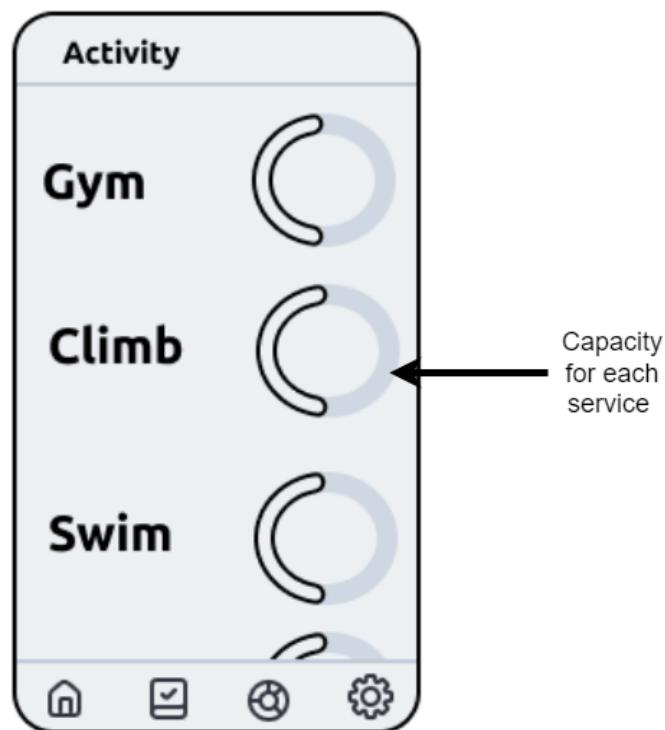


Figure 5.6: Activity screen wireframe

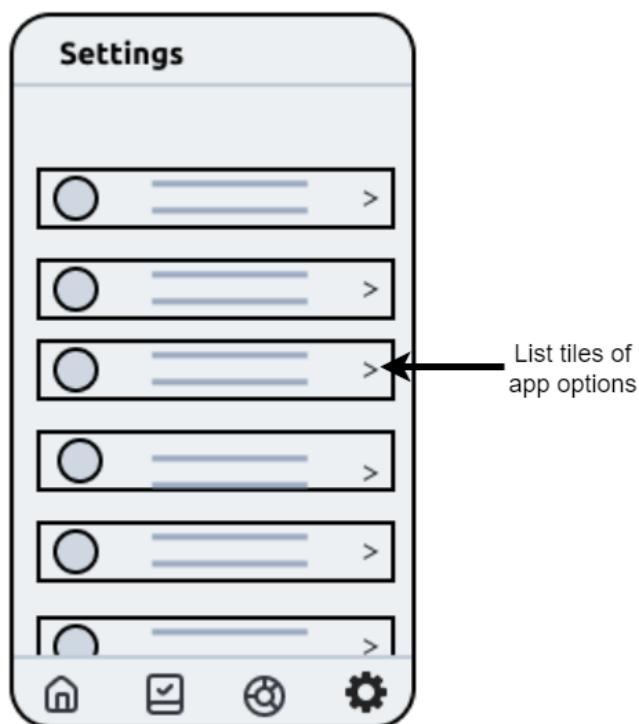


Figure 5.7: Settings screen wireframe

6 Implementation and Testing

This chapter details the overall journey of the implementation process of the application. Sections 6.1 to 6.3 would look into the various iterations involved in the implementation process, and their subsections would look into development, testing, and review. It entails the way the requirements and design are translated into actual code and any challenges or obstacles encountered during this process. This application was developed using Flutter (Dart) for Android specifically, but dependencies for iOS are in place but not tested due to the lack of the facility to do so. The chosen development platform was Visual Studio Code.

6.1 Iteration 1

This iteration involves setting up the project code environment and developing a basic design for the structure of the application to allow for navigation and authentication. This iteration attempted to implement the following requirements, as they are the foundational features of the application after which any other feature can be implemented.

- Authentication (MR-1)
- Bottom Navigation Bar (MR-2)
- Booking (MR-3)

6.1.1 Development

MR-2 Bottom Navigation Bar. For navigation, the bottom navigation bar was chosen against the navigation drawer because it is one of the most popularly used methods of navigation in mobile development and it also makes it easy to switch between top-level views/ screens in a single tap.

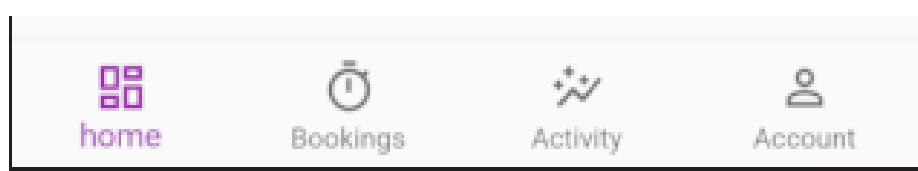


Figure 6.1: Bottom Navigation Bar

The navigation bar was designed so that there are four icons, which a user can click to navigate to different screens. The main screens are: Home, Bookings, Activities, and Account. As seen in Figure 6.1, each icon on the navigation bar has a callback function that navigates to the desired screen on tapping it. These icons also have a label attached to

them to make it easier for users to know what screen they are trying to navigate to, and in order to show that it is the current screen, the icon for the current screen is made to light up while the others are greyed out.

MR-1 Authentication. In this iteration, the initial screens for authentication were designed, the login screen and the sign up/register screen. This screen is displayed on the first launch of the application. A callback function was attached to the ‘register’ text button, which navigates to the register screen for new users, and also the ‘login’ button, which navigates to the bottom navigation bar class and is set to display the home screen class first. However, there is currently no backend infrastructure for this to correctly store user information.

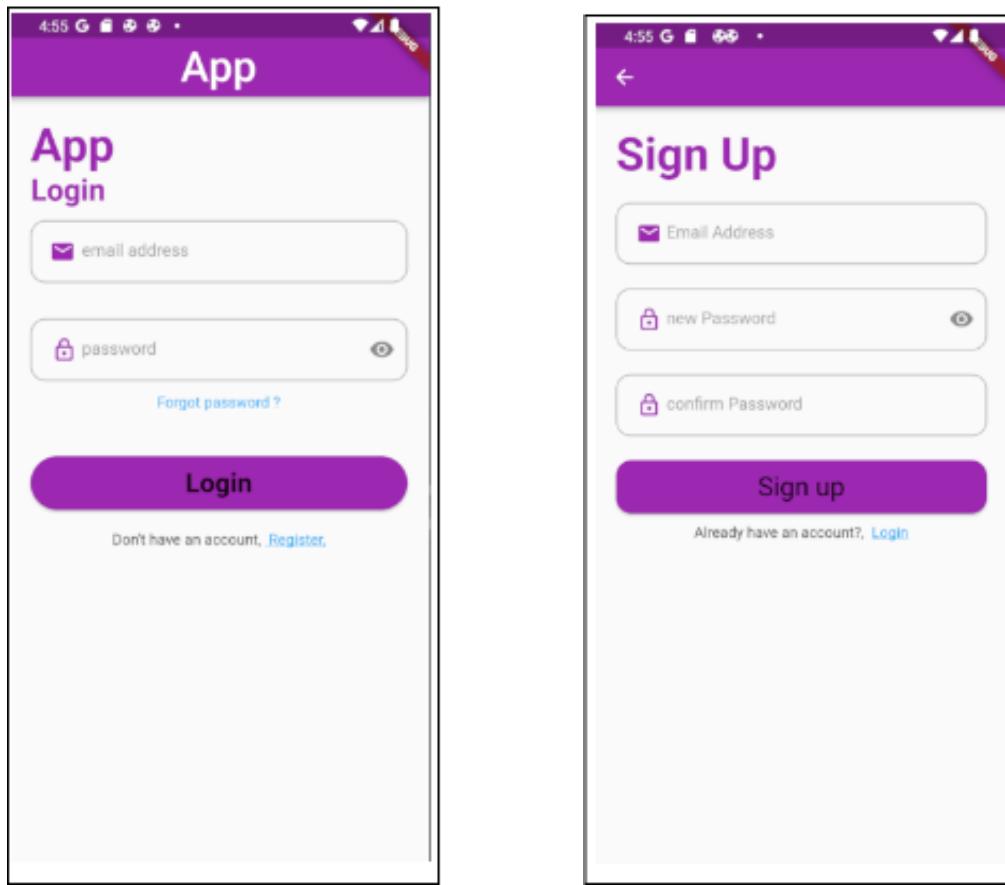


Figure 6.2: Login and Register Screen

MR-3 Booking. In the booking screen, as seen in Figure 6.3, a Floating Action Button (FAB) was placed at the bottom left of the page so that the users could reach it easily. This button would be used to make bookings.

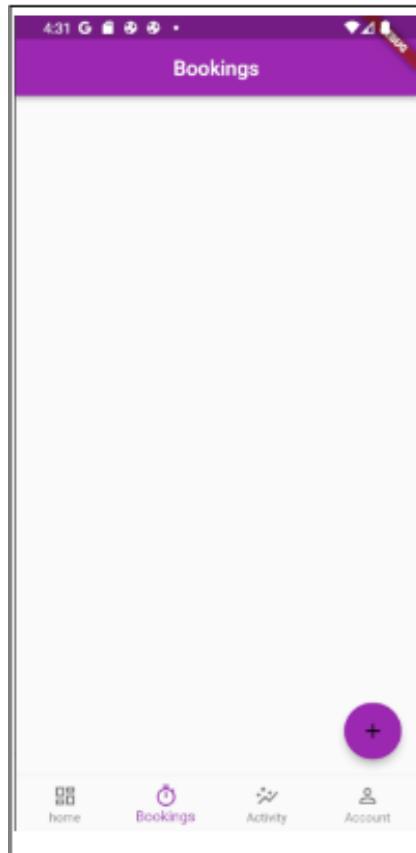


Figure 6.3: Booking Screen

On pressing the FAB, it expands, and, as shown in Figure 6.4, smaller buttons are revealed. The services that users can book using these buttons are represented by the illustrative icons. On clicking these buttons, a dialogue box is displayed, as seen in Figure 6.4, and on clicking 'Accept', the user would be directed to a page or a calendar, depending on the service chosen.

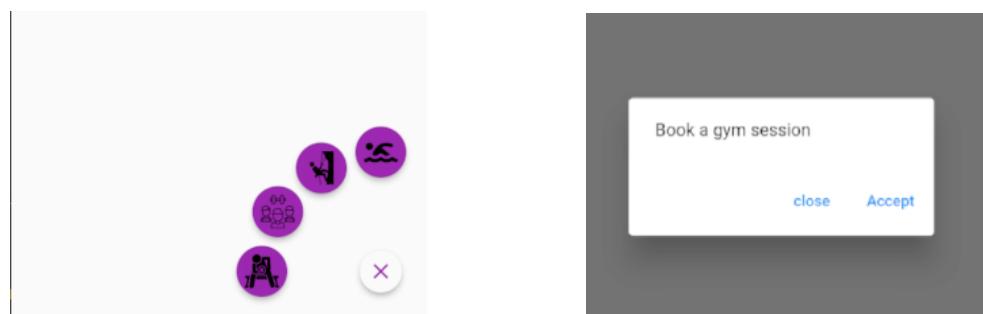


Figure 6.4: Expanded FAB and dialogue box

The initial calendar view for the booking, as seen in Figure 6.5, was added to the code base. This is the ‘syncfusion_flutter_calendar’ widget dependency, which allows a calendar to be displayed and interacted with for scheduling events by the user. This widget dependency saves time when creating a calendar view from scratch. No backend features have been implemented for this screen, and they are being contemplated. This was chosen because it has a similarity to the booking calendar found in the existing application (Figure 4.4).

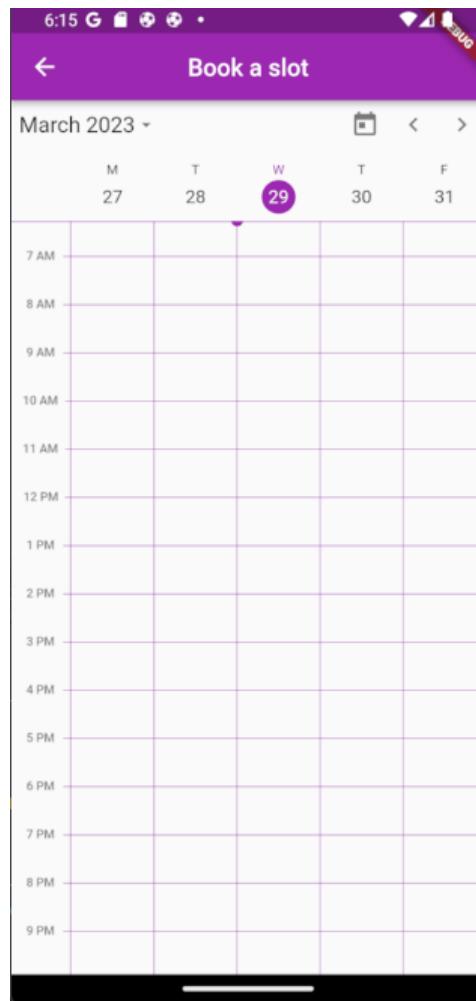


Figure 6.5: Booking calendar screen

6.1.2 Testing

Manual testing was conducted to ensure that the screens displayed correctly and the buttons navigated correctly.

Description	Expected Outcome	Results
(Navigation) from: login screen To: Home Page on clicking login	The application displays the home screen after clicking the login button.	Success
(Navigation) from: login screen to Register screen	The application displays the Register screen after clicking the "Register" text button.	Success
(Navigation) To: Booking screen	The booking screen is displayed with the FAB.	Success
(Navigation) To: Activity screen	The activity screen is displayed correctly.	Success
(Navigation) To: Account screen	The account screen is displayed correctly.	Success
(Navigation) To: the syncfusion calendar view	The calendar view is displayed after clicking "Accept" in the alert dialogue box.	Success

Table 6.1: Iteration 1 test and results

6.1.3 Review

This iteration was completed on time, and all the goals for the iteration were ticked off and working. There was a little feedback gathered after this iteration. It was noted that the booking screen appeared empty with little to no information on the task the user is expected to carry out, and what the button should do was not described anywhere. It was also noted that the illustrative icons from the expansion of the FAB do not have much information on the service they represent, and the dialogue box that appears on clicking them makes the interaction process a bit jagged and just increases the time required to complete a short task. It was advised that these services be explicitly shown with context so that the users can know what the screen is about and what task to carry out instantly. The reduction of white space on the screen would make the screen look lazy and bland and cause the user to lose interest.

The changes from the feedback would be worked on in the next iteration.

6.2 Iteration 2

This iteration attempted to set up the database structure and backend functionalities of the following features and also focused on working on the changes mentioned in Section 6.1.4.

- Authentication (MR-1)
- Bottom navigation bar redesign (MR-2)
- Cloud Database (MR-6)
- Updates and news content (MR-5)

6.2.1 Development

MR-6 Cloud Database. The database setup for the project was the Firebase Cloud Firestore, hosted by the Google Firebase service. The Firestore dependencies were installed and added to the code files for use. A model folder was created to store the model classes for the collections in the database, and the collection for the news and updates was manually populated in the Firebase console for display on the home screen. This is a NoSQL database, and as such, data is stored by mapping the appropriate field name to a JSON format, which is then passed to the Firestore. The JSON mapping is then stored in a collection of documents. For instance, under the ‘Users’ collection, each registered user would have a specific document with a respective JSON mapping that contains their user credentials.

Data can also be retrieved from the database and then used to populate the UI of the application. This Firestore reference is first initialised, and then the collection or document data to be retrieved is selected. The snapshot is then used to access the data in the collection and return a map of the document fields and values. For the purpose of handling potential exceptions, a try-catch block is implemented. This database was straightforward to set up with the documentation provided by Google Firebase and the help of online tutorials.

MR-1 Authentication. This iteration focused on the backend implementations for authentication. First, the project was integrated with Firebase in order to manage authentication from the Firebase console. For the login, the email and password fields have controllers that enable text written into them to be passed into the Firebase email and password variables to be stored. An email validator dependency was added to the code base and embedded in the text fields, which checks that the email is correct and valid. There was a struggle trying to manually create a function to validate the email, and this dependency made it easier and quicker. Firebase was then initialised in the code base, and then callback functions were attached to the ‘login’ button. The function takes in the input

email and password from the text fields and then checks if the user is in the list of authenticated users; if yes, it navigates to the bottom navigation bar as stated in iteration 1.

For new users, the register screen is displayed. This screen works just like the login screen with an extra input, ‘username’. Once these text fields are filled with the appropriate information and the register button is tapped, the user is added to the authentication list with the details provided, and the user is also added to the Users collection in the Firestore cloud Database.

MR-2 Bottom Navigation Bar Redesign. Based on the feedback gathered from iteration 1, some design changes were implemented. As seen in Figure 6.6, the bottom navigation bar was redesigned to match the updated material design theme. The screens were also redesigned to match the theme of the new navigation bar; this would be seen in further feature implementations.

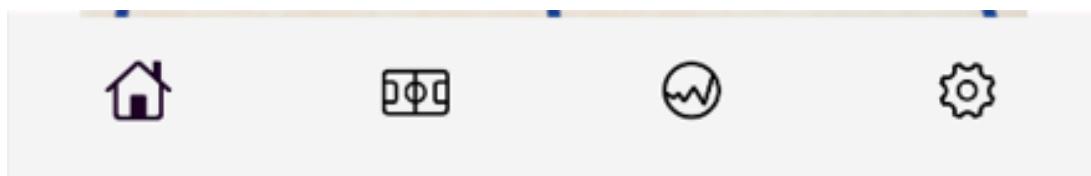


Figure 6.6: New Bottom Navigation Bar

As seen from some similar applications, the label of the screen is contained at the top left in the appBar, and as such, it would not be necessary to put it with the icons in the bottom navigation bar. The icons were carefully chosen so that they could represent the screen correctly and also be identified by any user. The Services screen, formerly called the Booking screen, was redesigned. From the comments received after iteration 1, the services were represented in detailed grid views, as seen in Figure 6.7, in order to give some information about the particular service and reduce the time it would take to find and book one. This was aimed at improving the user experience and familiarity with the application.



Figure 6.7: Services screen

MR-5 Updates and news content. This feature is contained on the home screen. It is a listview, which consists of an image frame and a title right below it. This screen was populated with data from the "News" collection in the Cloud Firestore database. A snapshot was created to access the data in the documents in the News collection and convert it to a string. These strings are then passed to variables that are used to populate the widgets.

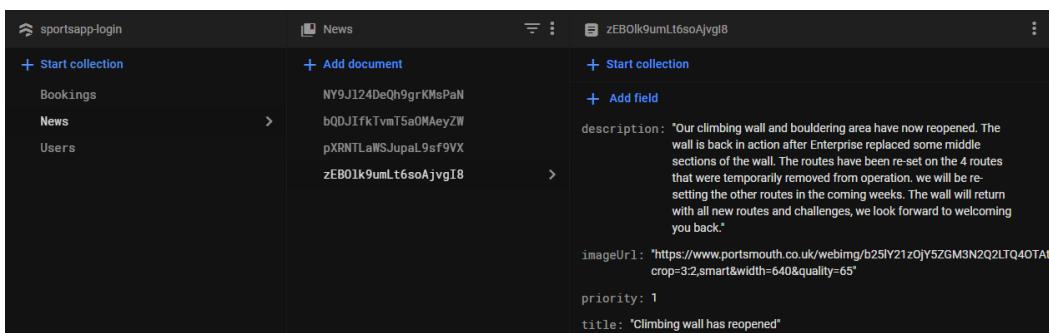


Figure 6.8: News collection in the Firestore database

The image is a Google URL, and the title and description are strings. They were manually entered into the database, specifying each field and its values. These could have been hard-coded, but it was a better idea to have them stored and retrieved from the database to ensure minimal changes if an update needs to be added or removed from the screen. The list was ordered according to their priority; this is so that the users would see the most important information without scrolling too far.

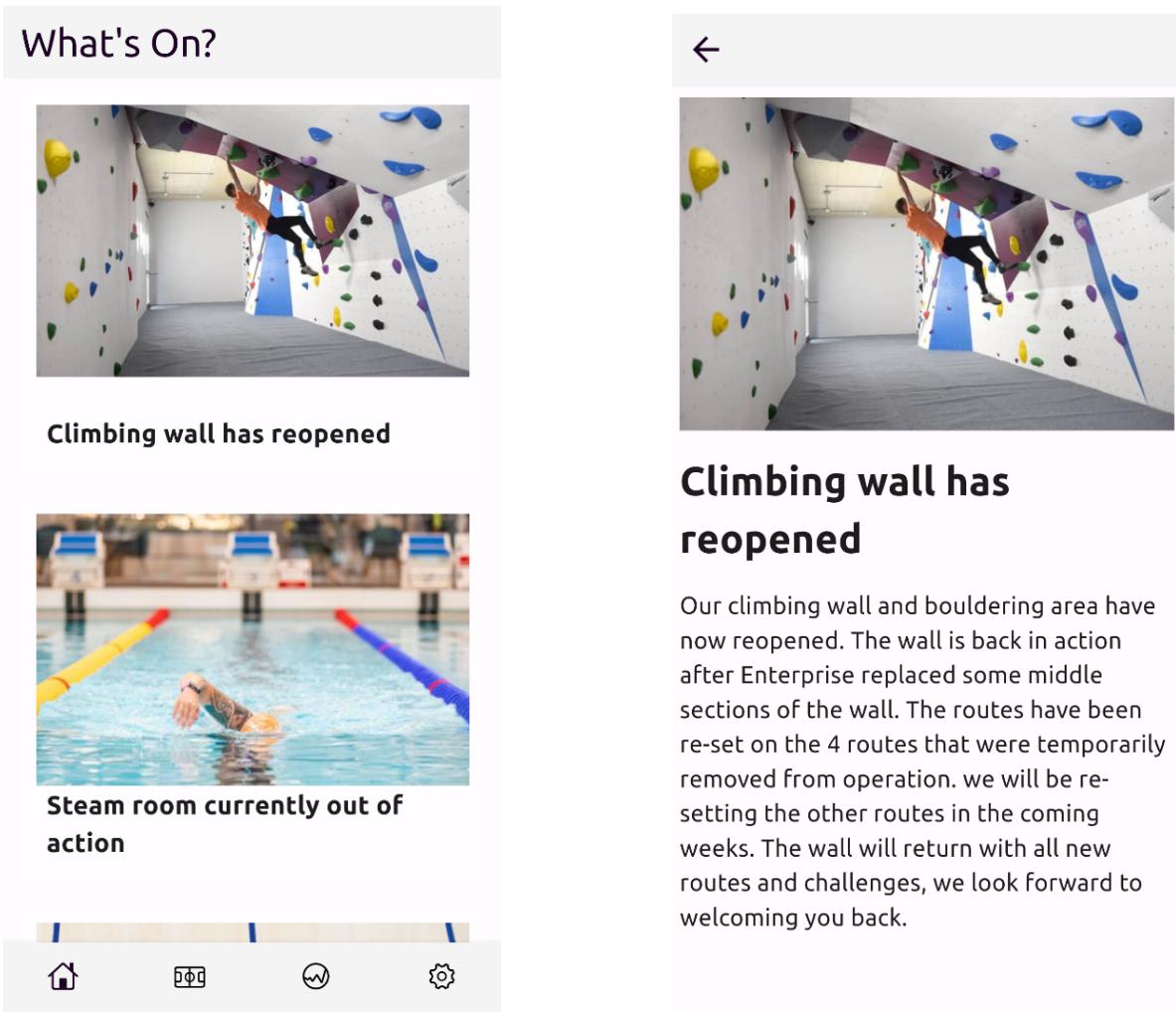


Figure 6.9: Home screen and description screen

A callback function was attached to each list view, which navigates to a new screen and displays the image and title from the list view, as well as a brief description of the notice, as shown in Figure 6.9. This was an adaptation of the existing application, but it was redesigned and made easily accessible.

6.2.2 Testing

Manual testing by running the emulator and interacting with the application was conducted to test the features. There were no flagged problems or errors.

Feature	Description	Results
Authentication (MR-1)	User able to register with valid email address and password	Success
Authentication (MR-1)	User able to log in with correct credentials	Success
Authentication (MR-1)	User not allowed to register twice with the same credentials	Success
Authentication (MR-1)	User can log out of the account	Success
Authentication (MR-1)	Flags wrong login details	Success
Authentication (MR-1)	User can reset password	Success
Authentication (MR-1)	Recovery email sent to email address provided in text box	Success
Cloud database (MR-6)	Database setup correctly in the project	Success
Cloud database (MR-6)	Store user data from register screen in Firestore database	Success
Cloud database (MR-6)	Retrieve user data from Firestore database	Success
Cloud database (MR-6)	Store updates and news content in Firestore database	Success
Cloud database (MR-6)	Retrieve updates and news content from Firestore database	Success
Updates and news content (MR-5)	Content from database is displayed correctly in the home screen list view widget	Success

Table 6.2: Iteration 2 test and results

6.2.3 Review

This iteration was completed but not timely; some features took longer than expected due to a lack of expertise in the programming language. However, the goals for the iteration were met and functional. There was satisfactory feedback for this iteration, and it was noted to put a bit more focus on the design of the entire application as it focused on UI/UX improvements and the implementation of the booking feature. The advice from this iteration would be applied in the subsequent iteration.

6.3 Iteration 3

1. Booking (MR-3)
2. Edit and delete bookings (MR-4)
3. Limit on bookings (SR-3)

6.3.1 Development

MR-3 Booking. In iteration 1, a booking calendar view was designed and implemented. In this iteration, that view was replaced and redesigned with the widget dependency ‘Booking Calendar.

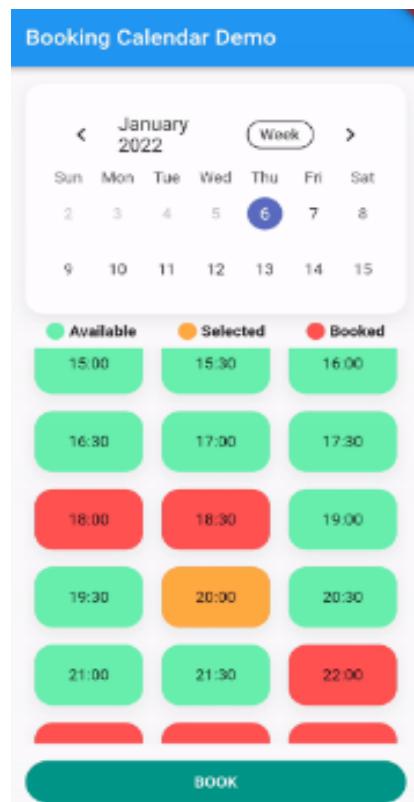


Figure 6.10: Booking calendar widget dependency

As opposed to trying to modify the syncfusion calendar to fit, this widget dependence was more directly tied to the objectives that needed to be achieved and was also easier to develop and work with. Although it needed to be set up correctly to match the requirements, The booking service instance was created, and the arguments were assigned variables. Figuring this out at the start took a good amount of time, but with more trial and error, it was figured out. The user details are retrieved from Firebase and passed to the user arguments. The service name, duration, and booking start and end are set manually to match the

opening times of the sports centre and the normal booking procedures. The colour coding that was standard for the dependency was changed.

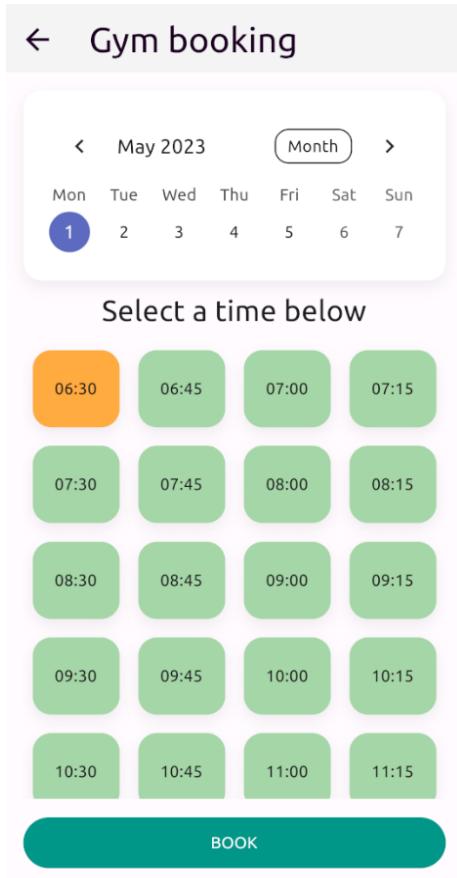


Figure 6.11: The updated booking calendar view

From Figure 6.11 above, after repeated attempts, the key located directly beneath the calendar was deleted and replaced with plain text because it was impossible to really change its colour or size. The slot colours chosen were believed to convey the messages needed for the user to know how to interact with the booking calendar. Green was chosen to show the availability of slots; orange was chosen to show a slot was selected or tapped by the user when booking; red was chosen to show that a slot is fully booked and cannot be booked anymore; and grey was chosen for the time that has passed. For the grey slot, there seemed to be an issue with the widget; when the time passes, the slots no longer appear in the booking calendar. This was a struggle, and after long hours of debugging, it is still unknown why the calendar removes the slots. There seems to be a problem with the widget for the grey since, after some time has passed, the slots vanish from the booking calendar. After many hours of troubleshooting, the reason why the calendar removes the slots remains a mystery. The widget was then connected to the Firestore database. The arguments stated above were mapped into a JSON format. When a slot is clicked, it represents the start time

of the booking. Each booking is for an hour, and as such, the end time would be an hour from the time of the slot selected. When the Book button is tapped, a callback function is set to wait for 2 seconds while a linear progression bar labelled "Please wait" is displayed. The function then converts the fields to a JSON mapping, and the booking information is added to the "Bookings" collection in the database. Each booking was designed to be a new document in the collection, with the user ID and the service name as the indexes.

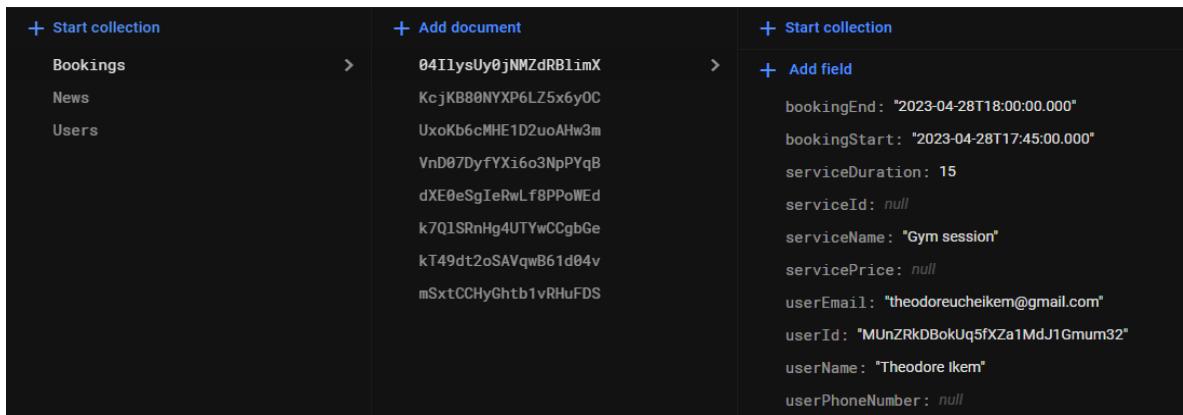


Figure 6.12: Documents in the "Bookings" collection

The list of bookings for that particular user is then retrieved from the database and displayed in the "Your Bookings" screen. This screen is displayed right after the booking is confirmed as successful by a Snackbar.

MR-4 Edit and delete bookings. After the booking system was functional in storing booking information in the database, the ability for users to edit and delete bookings in the application was implemented. This feature is embedded in "Your Bookings," and a trailing menu icon button was added to the lists. On pressing this icon, a popup menu that shows two options, Edit and Cancel, is displayed, as shown in Figure 6.13. A function was added to the edit option such that when it is clicked, the user is taken back to the booking calendar and can choose a slot again and book. Once the book button is clicked, the current booking is deleted and replaced with the new one in the database, and when the cancel option is clicked, the booking is deleted from the database and a snack bar is displayed if successful.

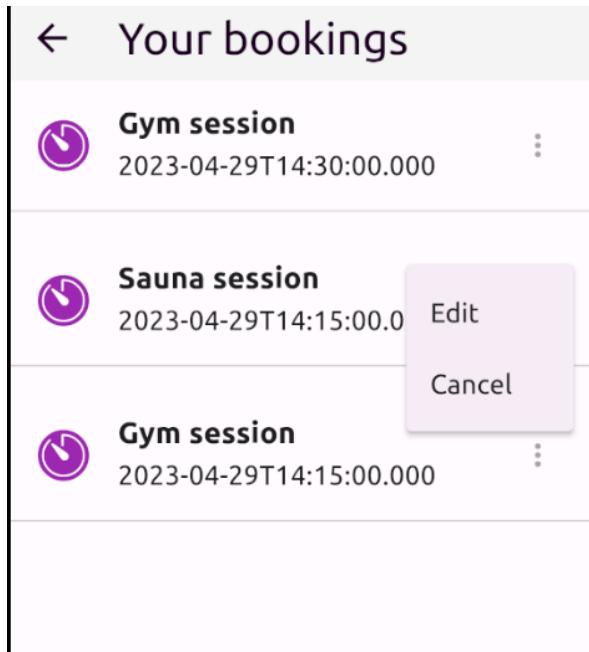


Figure 6.13: Popup menu

This was fairly simple to accomplish because all that was required was to just obtain the snapshot ID that is used to build the list and remove that reference from the Firestore database. However, implementing the edit feature was a struggle because the default booking calendar does not replace but only creates new booking documents, and when this was clicked, it would redo the whole initial booking process instead of recreating the instance of a particular booking. This would still be worked on to figure out a workaround for this problem.

SR-3 Limit on bookings. A limit on the number of bookings per user was put in place to avoid overbooking and guarantee an equitable allocation of time slots. For each user, there was a daily cap of two bookings. Through the addition of a check in the booking creation endpoint, this limit was enforced on the backend. The booking calendar API provided an error message if a user attempted to make more than two bookings in a day, indicating that they had reached their daily booking limit.

A message informing the user that their daily limit has been reached was displayed to the user using an alertdialog box when a user attempted to make a third booking in a day. To ensure this, a user account was created and signed in. This account was then used to create two bookings across the different services, and on creating the third booking, the user was not allowed to do so and was presented with the alertdialog.

By implementing this, users are prevented from overbooking services and are guaranteed equal access to time periods.

6.3.2 Testing

Unit testing was created for the booking creation endpoint. The booking create function was tested by verifying that it behaves correctly by providing the correct BookingService object, and an expectLater function checked that the function does not throw an exception and then further verified that it navigates to the correct screen on success. Then it was tested again with an invalid BookingService and verified that the error was handled correctly and that the error snack bar was displayed correctly. The same procedures were followed for the implemented features listed in 6.3.

Features	Description	Results
Edit and delete booking (MR-4)	User is able to delete a booking, and it is reflected in the database.	Success
Edit and delete booking (MR-4)	Users are able to edit a booking, and the updated information is reflected in the database.	Fail
Booking (MR-3)	Users are able to make bookings and save them to the database.	Success
Booking (MR-3)	User bookings displayed on the "Your Bookings" page from the database.	Success
Booking(MR-3)	The colour of the slot turns grey when time passes.	Fail
Booking (MR-3)	The colour of the slot turns red on unavailable bookings.	Success
Booking (MR-3)	The colour of the slot turns orange on selection.	Success
Limit on booking (SR-3)	Users are not able to create more than two bookings per day.	Fail
Limit on booking (SR-3)	error on making the third booking per day.	Fail

Table 6.3: Iteration 3 test and results

6.3.3 Review

This iteration was one of the most challenging; most features here were on a complex level due to the low expertise in the programming language, and it took a slightly longer time to grasp some context, work through them, and complete the iteration overall. It was very insightful, and there was a lot learned during this time.

The booking feature was implemented very clearly in terms of displaying user bookings and the functionality of the booking calendar. However, some issues were encountered with this feature, which allowed users to make more than two bookings per day and also not be able to correctly edit already-made bookings.

Overall, despite the fact that the application is still in its early stages of development, we made great strides in this iteration towards achieving our UI/UX and booking functionality goals. But more effort is needed to fix the problems that still exist and enhance the user experience as a whole.

7 Evaluation

This chapter focuses on conducting an assessment of the functionality and stating the extent to which it has been met and the reasons to which it has not been met in relation to the pre-defined requirements. This chapter presents the findings of the evaluation and justifies the means by which the project outcome was assessed.

MR-1 User Authentication: With the help of Firebase, this feature was successfully implemented, and users can register and login with email and password, as well as with Google.

MR-2 Dedicated pages for features: This feature was implemented successfully. This feature was made feasible by the use of the bottom navigation bar.

MR-3 Book multiple services: This feature was partially implemented. The main services that can be booked are currently those that make use of the booking calendar view. The other services were manually updated, and due to this, there was little chance of getting much data on what was available. However, this feature can be met as soon as enough data is gathered.

MR-4 Manipulation of Bookings: This feature could be considered completely implemented because, in comparison to the existing application, there has only been a need to cancel bookings if it is impossible to show up so that other users can make bookings as well. Cancellation and deletion of bookings were fully implemented, but editing the bookings was only partially implemented.

MR-5 Updates and News Content: This requirement was not fully implemented. The news and updates display correctly in the app, and they have been designed to update flexibly from the database.

MR-6 Cloud Database: The cloud database was fully implemented and allows user, app, and booking data to be stored and retrieved from the cloud efficiently. This feature was a must to implement because it enhances the functionality of other features.

SR-1 Membership Control: This requirement was not implemented; however, research is still being done on how to implement this feature, and steps have been taken to make a start on it.

SR-2 Notification: This requirement was not met. This was omitted during the entire implementation process in order to focus on more important features. However, this is a simple feature that can be implemented if given more time.

SR-3 Limit on Bookings: This requirement was partially implemented. This is due to the fact that work has already been started on it, but there have been obstacles regarding the programming that hinder the completion of this feature. With more time and practise, this feature should be done and perfected.

CR-1 Capacity Tracker: This feature was partially implemented. This was because the feature was dependent on the SR-3 feature. There has been a start on this feature, but due to the timeframe, it was not implemented and tested for correctness.

CR-2 Mobile payment feature: This feature was not implemented. It was a feature that could have been added to the application to make payments easier, but it is reliant on the **SR-1** feature, and because that has not been fully implemented, this cannot be implemented.

CR-3 Workout and Food/Diet Recommendations, Plans, and Tracking: This requirement was not met and was later regarded as an implementation that could be adopted in future work. Due to the timeframe, this was not considered for implementation anymore.

NF-1 Usability: This was achieved, and the application is simple in design and the interface is clear and consistent.

NF-2 Compatibility: The application was made fully compatible with Android devices running at least Android API 30.

NF-3 Accessibility: This requirement was partially met. Implementation on this was done but was not fully tested to be functional; however, the minimum accessibility requirements were sorted, like colours and visibility.

NF-4 Performance: The requirement was met, and the application was tested fully to ensure that the performance on several devices was smooth. The screens loaded up on time, and the overall responsiveness was great.

NF-6 Security: This requirement was met as all user data is protected using Google Firebase security. User data access is user-specific, while application-shared data can be accessed by all users.

8 Conclusion

The objective of the project was to address UI/UX and usability issues in the existing UoP sports application. An attempt to find a solution to this issue was made, and the application produced has achieved the majority of the goals set to solve this issue, with the application focusing more on user interface design, overall experience, and the improvement of some core functionalities. From the research on user-centred design principles in Chapter 2 and the use of iterative development justified in Chapter 3, a number of improvements and features have been effectively implemented in the produced application.

Chapters 2 and 4 presented the background and detailed the needs for the project. This was very important because this stage was crucial to the development of a solution to meet the aims and objectives of the project. The application had an initial design that evolved through the development stages through consideration of the user experience.

Despite the many challenges faced, the methodology used ensured that development was not rigid and that requirements were met at a good pace. It also paved the way for solving obstacles that occurred during the project. It also helped in the management of time and ensured that certain objectives were met at the appropriate time. In as much as some objectives were not completed within the specified timeframe, it was able to make sure that the spillover time was not a lot, and those objectives were met to begin new ones. And finally, we ensured that the outcome produced from the project was very decent. In order to ensure that the workload was minimised and that each iteration produced positive outcomes, the project was divided into manageable chunks.

8.1 Future Work

Due to time constraints and issues concerning coding proficiency, some features could not be implemented, and a few new features were considered after evaluation.

Booking: The unfinished features would be continued to make sure they were all fully functional. The booking feature could be further enhanced to make it more user friendly and useful. There could be features like appointment reminders for users about the bookings they have made and also mechanisms that can track user behaviours, engagement, and activity in the application.

Membership control: This feature would be very beneficial to users. It would make it easy for users to control their spending on the services. The existing application currently does

not have any way of quickly checking the status of memberships, and sometimes payments may come as a surprise to users. This could be done by integrating a dashboard within the application. Further work could be done to achieve this feature and possibly have it linked with the actual application.

Workout and food recommendations: Further work could explore a feature where methods for personalised workouts and diet recommendations based on user data are developed. A questionnaire could be used to create user preferences for recommendations. This feature could help users engage more with their fitness journeys.

Integration with external applications: As already noticed from the existing application, there are other applications that are integrated with the sports booking application. Further work could explore ways to integrate the application with external applications such as the UoP fitness app, Apple fitness, and other productivity applications.

8.2 Reflection

At the start of the project, there was a bit of fear, knowing the amount of work required and considering the application was being developed in a new programming language. However, the process, being a difficult one, was also a great learning curve. Core programming skills and project management skills evolved during this period.

Many issues arose throughout the project, and the methods by which solutions were discovered evolved. Skills in researching increased, and thoughts could now be arranged critically to get the best results. There were attempts made to get in touch with the team behind the creation of the current application to gain some understanding of how it operates and functions as well as perhaps integrate systems to make the process of putting improvements into place easier, but the sports department didn't respond. It would be wonderful if the team heard about the improvements, and there might even be some integrations with the current application systems, as well as the possibility of taking them on as clients. Even if this was not achieved, the project still commenced and was created as a standalone, and hopefully it will be put forward to the sports department and feedback can be gathered. The project management was a bit challenging, and some project deadlines were not met and spilled over; this is due to module deadlines clashing with some of the project deadlines in that same timeframe. This, however, was mitigated by making use of and adjusting the kanban board provided on GitHub and trying as much as possible to stay in coordination with the gantt chart hypothetically designed at the start of the project.

Overall, I feel confident in programming, project management, and, most importantly, reporting the journey. This was a wonderful experience, and I hope to continue it in the future.

References

- Akshay, V., Kumar S., A., Alagappan, R. M., & Gnanavel, S. (2019). BOOKAZOR - an Online Appointment Booking System. *2019 International Conference on Vision towards Emerging Trends in Communication and Networking (ViTECoN)*, 1–6. <https://doi.org/10.1109/vitecon.2019.8899460>
- Aljaroodi, H. M., Adam, M. T. P., Teubner, T., & Chiong, R. (2022). Understanding the Importance of Cultural Appropriateness for User Interface Design: An Avatar Study. *ACM Transactions on Computer-Human Interaction*, 26(7). <https://doi.org/10.1145/3517138>
- Anshar, M., Sadjad, R. S., Dewiani, Hanan, M., Prayudha, R., & Abry, M. (2020). Design and Implementation Monitoring and Booking Systems for Smart Parking at Engineering Faculty Campus. *IOP Conference Series: Materials Science and Engineering*, 875(1), 012036. <https://doi.org/10.1088/1757-899x/875/1/012036>
- Blair-Early, A., & Zender, M. (2008). User Interface Design Principles for Interaction Design. *Design Issues*, 24(3), 85–107. <https://doi.org/10.1162/desi.2008.24.3.85>
- Campbell, C. (2016, November 13). *Benefits of an online booking system - System Bookings*. System Bookings. <https://www.systembookings.com/benefits-online-booking-system/>
- Chandra, V. (2015). Comparison between Various Software Development Methodologies. *International Journal of Computer Applications*, 131(9), 7–10. <https://doi.org/10.5120/ijca2015907294>
- Constantine, L. L., & Lockwood, L. A. D. (2002). Usage-centered engineering for Web applications. *IEEE Software*, 19(2), 42–50. <https://doi.org/10.1109/52.991331>
- Despa, Mihai Liviu. (2014). Comparative study on software development methodologies. *Database Systems Journal*, 5, 3.

- Gangadharan, P. (2019, May 21). *The Importance of User Experience Design*. Medium; UX Planet. <https://uxplanet.org/the-importance-of-user-experience-design-988faf6ddca2>
- Garcia, E. (2022, March 22). *Top 11 User Interface Design Mistakes and How to Avoid Them*. Studio by UXPin. <https://www.uxpin.com/studio/blog/user-interface-designs/>
- Grace, M. (2022, January 10). *What is user interface design? And why is it important?* [Www.linkedin.com.](https://www.linkedin.com/pulse/what-user-interface-design-why-important-mathew-grace)
- Hoory, L., & Bottorff, C. (2021, October 26). *What Is Waterfall Methodology And How Do I Use It?* [Forbes](https://www.forbes.com/advisor/business/what-is-waterfall-methodology/) [Advisor.](https://www.forbes.com/advisor/business/what-is-waterfall-methodology/)
- Indeed Editorial Team. (2020, February 5). *What is a user interface? Definition, types and user interface example.* [Indeed](https://www.indeed.com/career-advice/career-development/user-interface) [Career](https://www.indeed.com/career-advice/career-development/user-interface) [Guide.](https://www.indeed.com/career-advice/career-development/user-interface)
- Joo, H. S. (2017). A Study on the development of experts according to UI / UX understanding. *KOREA SCIENCE & ART FORUM*, 31, 401–411. <https://doi.org/10.17548/ksaf.2017.12.30.401>
- Katunzi, S. (2021, November 26). *Iconography rules to follow in UI design*. Medium. <https://bootcamp.uxdesign.cc/iconography-rules-to-follow-in-ui-design-63733e7f74d6>
- Kukhnavets, P. (2018). Disadvantages and advantages in extreme programming. In *Hygger*.
- Moreno, L. (2020, December 3). *Fundamentals of layout in user interface design (UI)*. Medium. <https://uxdesign.cc/fundamentals-of-layout-in-interface-design-ui-3a9dba31f1#:~:text=The%20layout%20is%20the>
- Muniz, javier. (2017, November 12). *12 most important features your online booking system should have*. [Bookinglayer.](https://www.bookinglayer.com/article/the-12-most-important-features-your-online-booking-system-should-have)

- Nash, M. (2017). HCI design and age groups. *Hohonu-A Journal of Academic Writing*, 15, 39–42.
- Nielsen, J. (2012, January 3). *Usability 101: Introduction to usability*. Nielsen Norman Group; Nielsen Norman Group. <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- Norman, D., & Nielsen, J. (2021). *The Definition of User Experience (UX)*. Nielsen Norman Group. <https://www.nngroup.com/articles/definition-user-experience/>
- Pışırge, A., & Peker, S. (2021, September 1). *A UML-Based Conceptual Model for Appointment Booking Systems*. IEEE Xplore. <https://doi.org/10.1109/UBMK52708.2021.9558929>
- Praveen, A., Nanda, K., Rajith, N., Giriraj, N., Radhika, R., Mahesh, N., Vishnu, K., Anjali, T., & Sarath, S. (2020, July 1). *Conference Room Booking Application using Flutter*. IEEE Xplore. <https://doi.org/10.1109/ICCSP48568.2020.9182183>
- Pumpkin Web Design. (2020, February 27). *Design considerations for online booking systems*. Pumpkin Web Design Manchester. <https://www.pumpkinwebdesign.com/web-design-manchester/design-considerations-for-online-booking-systems/>
- PureGym. (2023). *The PureGym App*. PureGym. <https://www.puregym.com/app/>
- Sakdulyatham, R., Preeyanont, S., Lipikorn, R., & Watakakosol, R. (2017). User Interface on Smartphone for Elderly Users. *International Journal of Automation and Smart Technology*, 7(4), 147–155. <https://doi.org/10.5875/ausmt.v7i4.1339>
- Sandesara, M., Bodkhe, U., Tanwar, S., Alshehri, M. D., Sharma, R., Neagu, B.-C., Grigoras, G., & Raboaca, M. S. (2022). Design and Experience of Mobile Applications: A Pilot Survey. *Mathematics*, 10(14), 2380. <https://doi.org/10.3390/math10142380>
- Sapio, D. (2020, December 16). *10 Principles for Typography in UI Design*. Medium. <https://uxdesign.cc/10-principles-for-typography-usage-in-ui-design-a8f038f43ffd>
- Singh, S. (2023, March 16). *Visual Hierarchy in UI Design*. Code Writers. <https://medium.com/code-writers/visual-hierarchy-in-ui-design-30d302a47048>

- Soobia.et.al., S. (2019). Analysis of Software Development Methodologies. *International Journal of Computing and Digital Systems*, 8(5), 445–460.
<https://doi.org/10.12785/ijcds/080502>
- Swain, A. (2020, September 16). *Why UX is IMPORTANT!* Coffee with the UI Girl.
<https://medium.com/the-ui-girl/why-ux-is-important-a6d9b9abb378>
- tutorialspoint. (2019). *SDLC Iterative Model*. [Www.tutorialspoint.com.](http://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm)
[https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm](http://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm)
- Weston, L. (2020). *App analysis: PureGym*. [Www.healthclubmanagement.co.uk.](http://www.healthclubmanagement.co.uk/)
[https://www.healthclubmanagement.co.uk/health-club-management-features/App-an](http://www.healthclubmanagement.co.uk/health-club-management-features/App-analysis-PureGym/34592)
alysis-PureGym/34592
- White, L. (2022, February 12). *5 Reasons Why a Good User Interface is Important* -. Codersera. <https://codersera.com/blog/why-a-good-user-interface-is-important/>
- Wigley, L. R. (n.d.). *What Makes a Good Online Booking System?* [Www.designer-Websites.co.uk.](http://www.designer-websites.co.uk/)
[https://www.designer-websites.co.uk/blog/post/online-booking-system-tips](http://www.designer-websites.co.uk/blog/post/online-booking-system-tips)
- Zhang, D., Adipat, B., & Mowafi, Y. (2009). User-Centered Context-Aware Mobile Applications—The Next Generation of Personal Mobile Computing. *Communications of the Association for Information Systems*, 24(1).
<https://doi.org/10.17705/1cais.02403>

Appendices

Appendix A: Project Initiation Document



School of Computing Final Year Engineering Project

Project Initiation Document

Theodore Uchechukwu Ikem
Recreating the UoP Sports
application

1. Basic details

Student name:	Theodore Uchechukwu Ikem
Draft project title:	Recreating the UoP sports App
Course and year:	Computer Science 2022/2023
Project supervisor:	Matthew Poole
Client organisation:	
Client contact name:	

2. Degree suitability

This project would be a display of some of the key critical thinking skills acquired over the years. It would also be a medium to show core programming skills and software (mobile) development skills.

The project identifies a problem and aims to analyse the problem and create possible hypotheses aimed towards getting a viable solution. There would be factors of trial and error doing this project but that is what it is about, a series of testing would be carried out also to make sure that requirements are met, justified and validated.

3. Outline of the project environment and problem to be solved

Explain the context of the project - a bit like setting the scene:

For engineering projects without a client:

This project would be investigating some of the functional issues in the existing UoP sports app like the account management issues, sophisticated bookings and also design flaws. This project would produce an overall better user experience, allowing a user to have full control of their membership and smoother bookings.

4. Project aim and objectives

The overall aim of the project is to design a fresh mobile application that can be adopted by UoP sports. The current application is functional and can be used to make bookings for sports, classes, swimming, climbing sessions and the gym. Although, this existing app has a few limitations where members of the gym cannot easily manage and cancel their memberships. Users cannot also check the activity of the sports centre, that is, how busy the

gym is. The app also has some features that are just unused or underutilised. This project would be diving into some of those issues and also make important UI/UX changes from the existing application. It aims to make managing memberships and making bookings easier and smoother.

To meet this aim, the requirements (functional, non-functional and user) would need to be gathered and validated along the lifetime of the project.

Simple draft requirements of the project

Must have →

- **Login/user authentication:** The application must have a login / sign up system that allows users to register or login to their accounts, in order to track all access to the service and also provide services to those users based on their authentication/authorisation status.
- **Booking feature:** The application must allow users to book time slots for the various activities provided at the centre, in order to control the number of users using the services and prevent a situation where it becomes packed due to free-entry. Members of the gym would have free booking for the services included in their membership while non-members would have a price attached to the bookings.
- **Clear User interface:** The application must have a simple, yet attractive UI that allows the users easily access and use all the features provided by the application.

Should have →

- **Membership Control feature:** The application should allow users to have a good control over their accounts and services they want to use. The application should allow users to cancel, amend and track their membership when they have to.

Could have →

- **Activity tracking:** The application could have an activity tracking feature that allows users to either see exactly how busy the gym is or a rough estimate on how many people are currently in use of the gym and other services.
- **Sports/ workout logging and recommendations:** The application could have a workout logging feature that allows users to log their workouts for that particular day and also recommend workouts based on the body part they choose to train.
- **Food / diet logging and tracking :** The application could have a feature where the users can log and track their food (caloric) intake and get scores about the total number of calories and protein had for the day.

5. Project constraints

This project's system cannot be tested on real gym users and some extra-functional implementations are bound to be in the evolution process even after the timeframe. The database would be hypothetically populated.

Getting a knowledge of the availability of the activities that are to be booked.

6. Facilities and resources

There would be use of some of the University provided computer systems, but I would mostly be in use of my personal computer. I would make use of some of the google developer tools and resources. The use of version control systems i.e. GitHub would be used.

7. Log of risks

Shown in the table below

Description	Impact	Mitigation/Avoidance
Technical Blocks	Cause delays to the project timeline	Ensuring work is started early and done as frequent as possible
Illness	Late submission, Delay in work progress	
Loss of data		Using version control systems like GitHub to save the work progress securely.

8. Project deliverables

A functional mobile / cross platform application catered mainly for android as at development stage.

The requirements to be delivered with this project are as stated above in the **project aim and objectives**.

The documents to be produced are the project report, SDLC containing some of the functional and non-functional requirements and design specifications (The wireframe or prototyping) and evidence of automated testing (widget, unit and integration testing).

9. Project approach

The project would be managed using the scrum (agile) methodology in conjunction with the gantt chart to visualise tasks against time. However, there would be tests each week on certain added features and progress to the application with my supervisor. There would be prior research into the programming language and framework being used, there would also be research into similar past projects relating to this project and existing systems.

10. Project plan

A more detailed Gantt chart is attached as an appendix.

The main stages and tasks of the project are :

- Definition of core requirements

- Setting up the environment where the project would be implemented (IDE and dependencies where necessary).
- Carry out research on previous similar existing or discontinued projects
- Plan and write a literature review.
- Take a note of all the occurrences along the lifetime of the project.
- Write a project report.
- Clean up the whole document for presentation.

For these above tasks, the skills needed would be programming skills and a level of proficiency in dart and flutter. The skills that are not there are going to be acquired through quick learning and applied to the project along the way.

11. Supervisor Meetings

There is an agreement to meet every two weeks, mainly at the same time, the subsequent meeting is confirmed at the end of the previous one.

Use Google calendar to effectively narrow out the issue of tight schedule and ensure the meeting is logged. Towards the deadline, meetings are expected to be more often. Emails are also used for any immediate opinions and feedback.

12. Legal, ethical, professional, social issues (mandatory)

The project being worked on does not impose any security threat to anyone and any organisation.

The project has been extensively reviewed and these things have been taken into account and is considered safe and legal.

Appendix B: Gantt chart

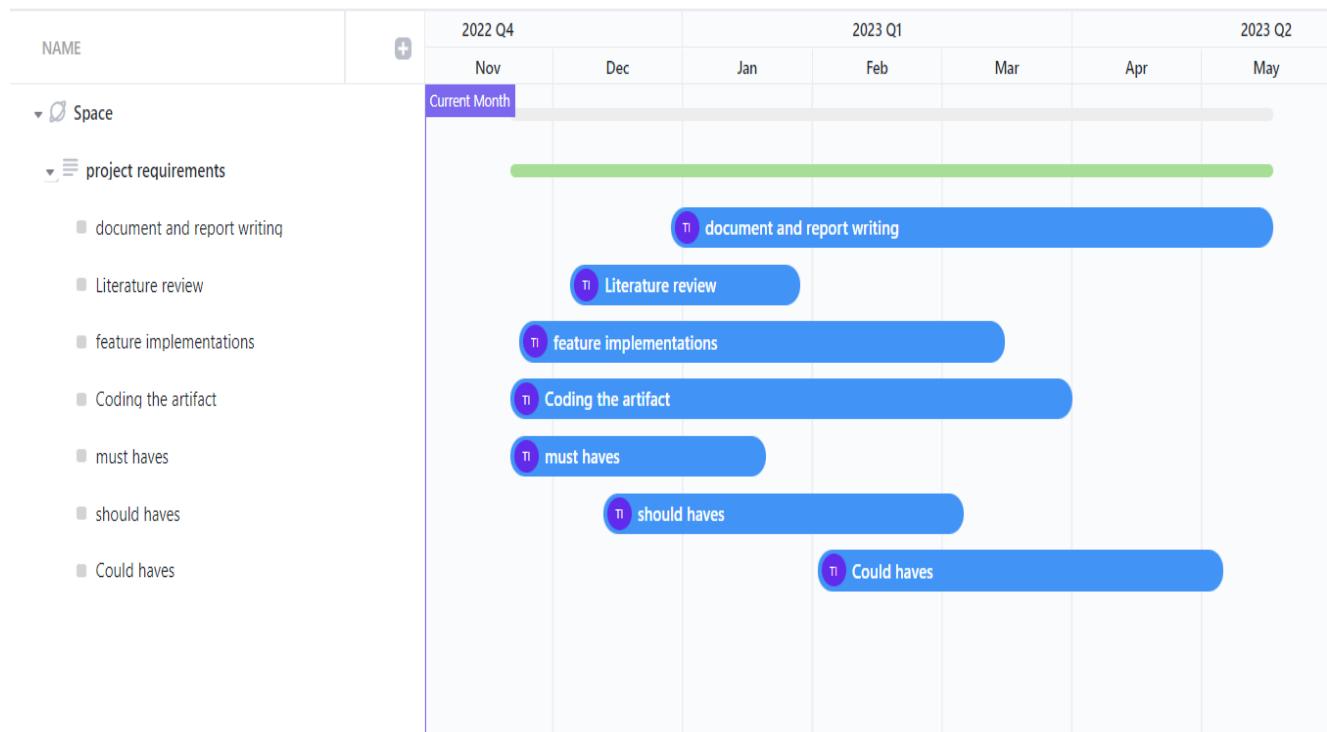


Figure B: Original Gantt chart

Appendix C: Ethics Form

Certificate of Ethics Review

Project title: Recreating the UoP Sports Application

Name :	Theodore Ikem	User ID:	968909	Application date:	20/10/2022 02:29:21	ER Number:	TETHIC-2022-10392 6
---------------	---------------	-----------------	--------	--------------------------	------------------------	-------------------	------------------------

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are **Haythem Nakkas, Dalin Zhou**

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- [University Policy](#)
- [Safety on Geological Fieldwork](#)

It is also your responsibility to follow University guidance on Data Protection Policy:

- [General guidance for all data protection issues](#)
- [University Data Protection Policy](#)

Which school/department do you belong to?: **School of Computing**

What is your primary role at the University?: **Undergraduate Student**

What is the name of the member of staff who is responsible for supervising your project?: **Matthew Poole** Is the study likely to involve human subjects (observation) or participants?: No

Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No

Are there risks of significant damage to physical and/or ecological environmental features?: No

Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No

Does the project involve animals in any way?: No

Could the research outputs potentially be harmful to third parties?: No

Could your research/artefact be adapted and be misused?: No

Will your project or project deliverables be relevant to defence, the military, police or other security organisations and/or in addition, could it be used by others to threaten UK security?: No

Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy. Supervisor comments:

Supervisor's Digital Signature: **Matthew Poole** Date: 22/10/22

Appendix D: Use Case Tables

These are techniques used in the development of a software application to capture and represent the functional requirements of a proposed application (Beimel et. al, 2018). It consists of a series of steps that represent a particular scenario and can be illustrated with use case diagrams, providing an insight into how the system should behave and what it should do in different scenarios.

Name	Make a Gym Session Booking
Actors	Logged-in user, Firebase database
Description	A user attempts to make a new gym session booking in the services page
Frequency of use	High
Triggers	The user clicks on the 'gym booking' button
Pre-conditions	<ul style="list-style-type: none"> 1. The user has successfully completed the login procedures 2. The home screen has been displayed correctly
Post-conditions	<ul style="list-style-type: none"> 1. The applications database is updated 2. The user is notified 3. The capacity is updated.
Basic flow	<ul style="list-style-type: none"> 1. The user navigates to the services tab. 2. The user clicks the 'gym' button and clicks on the 'gym booking' button. 3. The booking calendar is displayed. 4. The user selects an available time. 5. The user clicks the 'Book' button at the bottom of the page. 6. The booking is stored in the database. 7. The user gets a notification for the success of the booking.
Alternative path	<ul style="list-style-type: none"> 1. The booking fails due to some connectivity issues 2. There are no available times to book 3. The booking chosen has been fully booked at the instance of making that booking.

Name	Cancel a Booking
-------------	-------------------------

Actors	Logged-in user, Firebase database
Description	The user attempts to cancel an already made booking
Frequency of use	High
Triggers	The user clicks the cancel button at the bookings page.
Pre-conditions	The user has made a booking and now wants to cancel it
Post-conditions	<ul style="list-style-type: none"> 1. The booking is deleted from the database 2. The user is notified 3. The capacity is updated.
Basic flow	<ul style="list-style-type: none"> 1. The user navigates to the 'Services' page. 2. The user clicks on the 'cart' icon at the top right of the page. 3. The bookings are retrieved and displayed on the page. 4. The user clicks on the booking that is to be deleted. 5. The 'cancel' button is then clicked. 6. The booking is then deleted from the Firebase database.
Alternative path	<ul style="list-style-type: none"> 1. There are no bookings on the 'your bookings' page. 2. The time for the booking is already past 3. No connectivity with the firebase database

Name	Login to the application
Actors	Logged-in user, Firebase authentication
Description	The user attempts to login to their account on the application.
Frequency of use	Low (once)
Triggers	The user requests to login
Pre-conditions	The user has opened the application and wants to interact with it
Post-conditions	<ul style="list-style-type: none"> 1. The user is logged into the system 2. The user has access to all functionality of the system.
Basic flow	<ul style="list-style-type: none"> 1. The user accesses the application 2. The user is prompted to input their account credentials 3. The user enters their correct email address and password 4. The login is authenticated with the Firebase authentication 5. The user is granted access to the applications functionality.
Alternative path	<ul style="list-style-type: none"> 1. Invalid email address or password. 2. The user does not have an account and needs to register.

Name	Edit Profile
-------------	---------------------

Actors	Logged-in user, Firebase Database
Description	A user attempts to edit their profile account
Frequency of use	Medium
Triggers	The user clicks on the 'edit profile' button
Pre-conditions	<ul style="list-style-type: none"> 1. The user has an active account on the application 2. The user is logged in to the application
Post-conditions	<ul style="list-style-type: none"> 1. The profile has been edited 2. The changes are saved
Basic flow	<ul style="list-style-type: none"> 1. The user navigates to the 'Settings' page 2. The user clicks on the 'Edit profile' button below their name 3. The user is prompted to make the changes to their details 4. The user enters the changes 5. The user clicks the 'save' button 6. The changes are then updated in the Firebase database
Alternative path	<ul style="list-style-type: none"> 1. The changes cannot be saved at that time 2. Connectivity to the database not consistent