

Student Number – 19011230

Software Development Project – UFCFFF-30-3

Module Leader – Steve Battle

University of the West of England

Word Count -



The Golf Blueprint

A Smarter Way to Play Golf –

Driven by Analytics

Table of Contents

The Golf Blueprint.....	2
Glossary	6
1. Introduction.....	7
1.1 Problem Statement.....	7
1.2 Project Aims and Objectives	8
2. Methodology	9
2.1 Agile Methodology	9
2.2 Risks and Mitigation Strategies	12
2.2.1 Technical Risks.....	12
2.2.2 User Adoption Risks.....	12
2.2.3 Data Protection Risks.....	13
2.2.4 Project Management Risks	13
2.2.5 Version and Document Control Strategy	13
3. Project Research	14
3.1 Secondary Research	15
3.2 Primary Research.....	16
3.2.1 Current Understanding of Course Management	17
3.2.2 Current Use of Golf Apps or Technology.....	18
3.2.3 Interest Levels in an App/Website such as The Golf Blueprint	19
3.2.4 Willingness to Participate in the Required Data Collection	20
3.2.5 Desirable Features of The Resource	20

3.3 Key Findings.....	21
3.3.1 Research Limitations	22
4. Requirements.....	22
4.1 User Stories and Use Cases	22
4.2 Use Case Diagrams	24
4.2.1 System Overview	24
4.2.2 Contribute Shot Data	25
4.2.3 Heatmap Visualisation.....	26
4.2.4 Preview Course.....	27
4.2.5 Course Maintenance Analysis	28
4.2.6 Progress Tracking	29
4.3 Functional Requirements	29
4.4 Non-Functional Requirements	33
5 Software Design	35
5.1 Software Architecture.....	35
5.1.1 Software Architectural Pattern	35
5.1.2 Client-Side Architecture.....	37
5.1.3 Server-Side Architecture	37
5.1.4 Communication Protocol.....	38
5.2 Database Design	38
5.2.1 Database Schema	38
5.3 User Interface Design.....	39

5.3.1 Design Principles	39
5.3.2 Colour Palette	40
5.3.3 Wireframes.....	41
5.4 Key Algorithms and Technical Components.....	50
5.4.1 Shot Tracking Algorithm	50
5.4.2 Data Visualisation Algorithm	53
5.4.3 Security Implementation	55
References	78

Glossary

Course Management - The strategic decision-making process during a round of golf, including club selection, target selection, and risk assessment. Good course management involves choosing safer options when appropriate and understanding when to take calculated risks.

Green - The closely mown area of grass between the tee and the green, providing the most advantageous position for the next shot.

Approach Shot - A golf shot played toward the green, typically from a distance of 50-200 yards from the hole.

Fairway - The closely mown area of grass between the tee and the green, generally providing the most advantageous position for the next shot.

Par – The score that a golfer is supposed to take on a particular hole. For example, a score of 4 on par 4, would mean that you scored par.

Birdie – A birdie is a score of one less than par, on any given hole. For example, a score of 2 on a par 3 would mean that you scored a birdie.

Bogey – A bogey is a score of one higher than the par on any given hole, For example, a score of 6 on a par 5 would mean that you scored a bogey.

Double+ Bogey – a double bogey is a score of 2 over the par for any given hole. This term refers to any score which is this, or higher.

SVG – SVG stands for Scalable Vector Graphics. A vector image format that uses XML to define two-dimensional graphics.

API – API stands for Application Programming Interface. It is a software intermediary that allows two applications to talk to each other.

1. Introduction

1.1 Problem Statement

Golf enjoys tremendous global popularity with over 42.7 million players worldwide (The R&A, 2024), yet players at all levels struggle with 'course management'—a crucial factor in scoring well. Drawing from my 13 years of experience and a handicap of 3, I've observed that whilst technical skills matter, strategic course management often determines performance.

Golf at every level is fundamentally a game of misses; no player hits perfect shots consistently, making it essential to manage where one misses. This creates a market opportunity for The Golf Blueprint to revolutionise amateur golfers' improvement through strategic guidance.

My experience at The Kendleshire Golf Club illustrates this concept perfectly. For instance, missing approach shots to the left of the second green typically yields higher scores than missing to the right, as recovery shots from the left are considerably more difficult. Many golfers overlook such subtle but significant details across the course. The Golf Blueprint aims to visually represent strategic insights for all 18 holes in an accessible format through utilising heatmaps, helping golfers of all abilities identify areas to target or avoid during their rounds, ultimately leading to lower scores through improved decision-making.

1.2 Project Aims and Objectives

The primary aim for this project is to develop a data-driven golf course management tool, that is specifically tailored to The Kendleshire Golf Club, allowing its members and others who may play there to make informed strategic decisions during their rounds of golf, allowing them to improve their scores and to greater their experience. The specific objectives are:

1. Design and create detailed animated top-down view recreations of every hole at The Kendleshire, which will not only be used to collect the shot data from the golfers at The Kendleshire, but to also display the findings from the data, in the form of heatmaps.
2. Design and implement a system to collect and analyse shot data from golfers. This system must be very simple, and easy to use to ensure that golfers will be happy to

take the time to input their data, and to ensure that the data collected is as accurate as possible.

3. Create a secure database to store golf shot data, user account information and golf course information, complying to GDPR regulations. The database will be the foundation for the resource and is essential to make it possible to provide meaningful, data-driven course management recommendations to the users.
4. Design an intuitive, and good-looking user interface, which displays to the user all the features of The Golf Blueprint. This is a very important consideration as the user interface will serve as the main point of interaction between the users and the resource's functions and capabilities.
5. Ensure that the resource meets the needs of the users, by conducting testing with at least 10 users, and gather feedback from the users to gain an understanding of what elements of the resource they enjoy, and what could be improved.

2. Methodology

2.1 Agile Methodology

For this project to run smoothly, and with as minimal issues as possible, it is essential that I use a software development methodology. A methodology forms a framework for planning and controlling the creation of a software project (Kute and Thorat, 2014). To accommodate

for the flexibility and adaptability that developing The Golf Blueprint requires, an agile methodology would be more appropriate. I am creating this project in my third year of university, and there could be periods of time where I am extremely busy with other academic projects, that could cause a lengthy delay to my progress. Using an agile methodology will not force me to follow fixed-length sprints, allowing for a more flexible workload to fit around my varying university workloads. My agile methodology of choice will be Kanban. Kanban requires full transparency of work and presents work items visually on a Kanban board (Radigan, 2024).

In Anderson's work (2010), he emphasises the importance of limiting work in progress (WIP) for maintaining the quality of work while preventing overload. If I can limit myself to having only a manageable number of tasks at any given time, I will be able to complete them to a higher standard (Sjøberg, 2018). The visual nature of Kanban boards will also allow me to quickly identify any bottlenecks in the development process and address them promptly (Lei *et al.*, 2017). This visibility is particularly valuable as it will help me prioritise tasks effectively when balancing multiple simultaneous university commitments.

Another significant advantage of Kanban for this project is its focus on continuous delivery, which (Ahmad *et al.*, 2013) identify as a key benefit of the methodology. Unlike methodologies that require waiting for sprint cycles to complete, Kanban allows features to be released as soon as they are ready. This will be useful for me as it will allow me to share any updates to my project supervisor, and to potential future clients at The Kendleshire.

To implement Kanban for this project, I have set up a digital Kanban board using Jira, Atlassian's project management software (see Figure 1.1). The board is organised into three columns: To Do, In Progress, and Done. The 'To Do' column contains all planned features and upcoming work, prioritised based on project requirements and dependencies. The 'In Progress' column, limited to three tasks to prevent WIP, shows current development activities. The 'Done' column provides a clear record of completed features and helps track project progress. This structure is complemented by Jira's additional features such as time tracking, issue linking, and automated workflows, which enhance project visibility and control. This structure aligns with key Kanban principles of workflow visualisation and process management (Anderson, 2010). This straightforward approach ensures I maintain a clear overview of the project's status while managing it alongside other academic commitments.

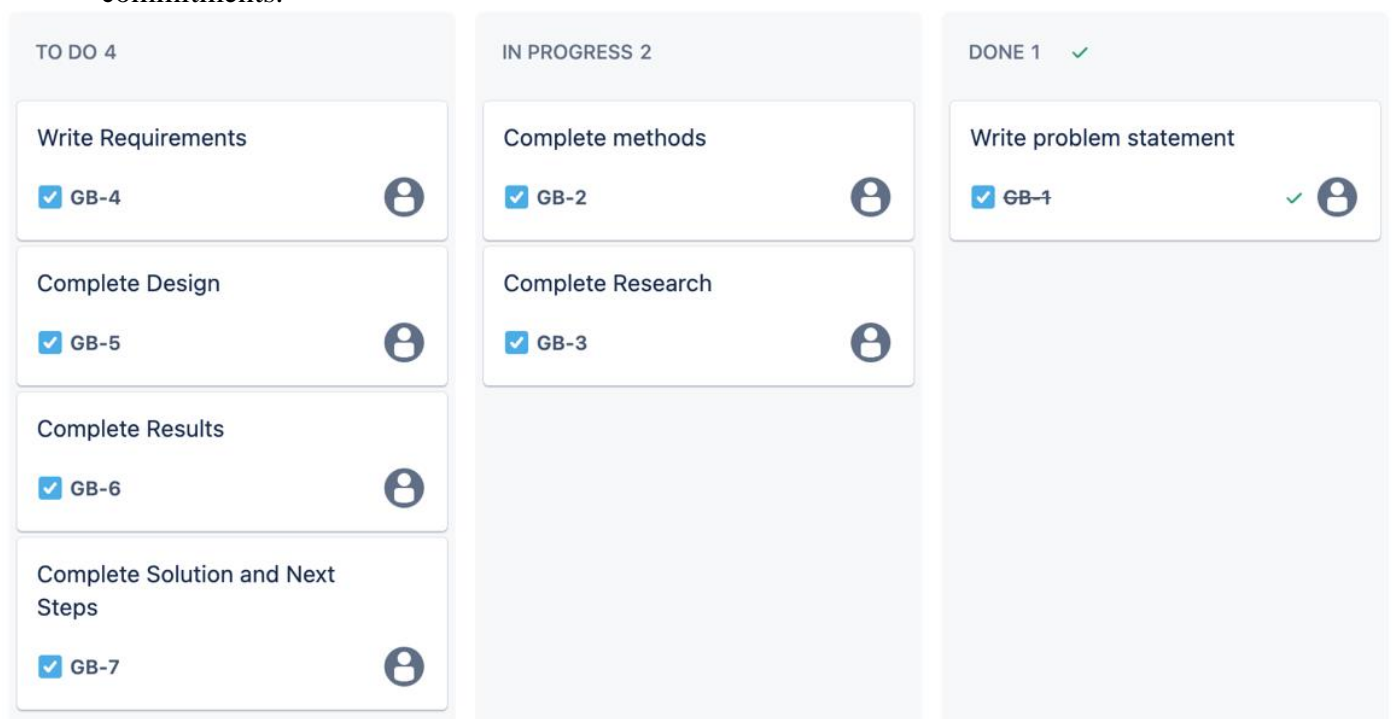


Figure 1.1: Kanban project board.

2.2 Risks and Mitigation Strategies

For The Golf Blueprint to be successful, it is essential to identify potential risks and develop appropriate mitigation strategies. There are various potential risks that could impact the project's success, and they must be carefully managed.

2.2.1 Technical Risks

The primary technical risk for The Golf Blueprint involves data accuracy and validation. Since the resource's ability to provide meaningful, data-driven course management insights relies entirely on user-submitted data, ensuring this data's accuracy is crucial. To mitigate this risk, multiple validation techniques will be implemented. The application will provide clear visual feedback for marking shot locations, combined with intuitive instructions for data entry. An intelligent validation system will be implemented to detect potential errors, including outlier detection to identify unusual patterns in shot data. Users will be limited to entering only two rounds of golf per calendar day to help prevent fabricated data from entering the database. Also, if a user attempts to enter an unrealistic number of shots for the same hole or records shot distances beyond normal playing capabilities, the system will challenge these entries, requesting verification before acceptance.

2.2.2 User Adoption Risks

The success of The Golf Blueprint heavily depends on user engagement and consistent data contribution. A significant risk exists that golfers at The Kendleshire might find the process of inputting shot data into The Golf Blueprint too time-consuming or complex, potentially leading to limited data collection and reduced effectiveness of the system. This risk will be addressed through careful interface design that will allow for simple and time efficient data

entry. Clear user instructions will be provided for accurate shot location marking, and the overall user experience will be developed to encourage regular participation.

2.2.3 Data Protection Risks

Due to the nature of the application, The Golf Blueprint will collect and store personal information from the users as well as golf shot data. This makes having a robust data protection strategy essential. The system will implement secure user authentication to protect personal information, alongside encrypted data storage for all user information. For the database, I will be using MongoDB, which has built in security features, meaning that all of the users' data will be completely secure.

2.2.4 Project Management Risks

There are several project management risks that need to be considered. The primary challenge lies in balancing the project development with my other academic commitments. I will mitigate this risk by utilising the features available with the Jira Kanban board that can help me keep on track with what parts of the project need to be completed at any given time. Jira has time tracking functionality, and by logging time spent on different aspects of the project, I can identify which components are taking longer than anticipated and adjust my development schedule accordingly.

2.2.5 Version and Document Control Strategy

The Golf Blueprint will utilise Git as its centralised version control system, hosted on GitHub for secure code management. The repository will maintain a main branch for stable code, with a development branch for ongoing work. New features will be developed in dedicated branches following the 'feature/description-of-change' naming convention. All code changes will adhere to a strict commit message format, beginning with a type identifier followed by a

concise description—for example, 'test(heatmap): implement shot density visualisation'—ensuring clear tracking of the project's evolution.

Beyond code management, all project documentation will be stored in a dedicated 'docs' directory within the repository, keeping documentation closely aligned with its corresponding code. This documentation will follow a consistent naming convention for chronological version tracking. Regular backups of both code and documentation will be maintained through GitHub's cloud storage, safeguarding against potential data loss whilst preserving the integrity and traceability of the entire project.

3. Project Research

This project utilises both primary and secondary research techniques to ensure that I developed an enhanced understanding of what I am looking to achieve, and to validate my proposed solution to the problem. The first part of the research phase of my project was secondary research. An important goal of mine for the secondary research was to not only explore the existing golf analytics technology that exists, but to also try and find a gap in the current market that The Golf Blueprint can fill.

To ensure the quality and reliability of the research that I found and used, I searched the UWE Library Database, and Google Scholar. Key search terms included: “golf analytics technology”, “importance of course management”, “global participation in golf”, “strokes gained” and “data used in professional golf”. This secondary research was able to provide me with an abundance of useful information, which all directed related to my project.

3.1 Secondary Research

The game of golf at all levels is continuously evolving, with technology and data analytics playing an increasingly crucial role in player development and performance optimisation. The PGA Tour's implementation of ShotLink technology in 2003 marked a significant turning point for golf, by collecting detailed data on every shot played in professional tournaments (Broadie, 2014). Broadie demonstrates how data analytics has revolutionised the understanding of golf performance, introducing new metrics such as "strokes gained" which compares different facets of a players' golf game to other players (Plummer, 2024). These metrics have become standard tools for not only professional player analysis but are also useful for weighing the value of different strengths, such as the power of exceptional ball-striking versus superior short game skills (Ehrlich and Kamimoto, 2024).

According to research published in the International Journal of Sports Science & Coaching, the integration of launch monitors and ball tracking technology has fundamentally changed how golfers practice and compete (Betzler *et al.*, 2012). Professional golfers utilise these resources frequently to analyse their own performance and arrange their practice schedule around their own weaknesses to improve and try to gain an edge on their competitors.

Course management has become a very big part of the modern professional game and is something that the world's top players always look to when trying to improve (MacKenzie, 2023). Professional golfers will also always without fail complete multiple 'practice rounds' before a tournament. During these practice rounds they identify which areas of the course they must avoid, and which areas of the course they should aim to utilise to benefit their scores, and in turn results (Stenzel, 2023). Understanding effective course management can help any golfer save some strokes every time that they play (Turner, 2023).

3.2 Primary Research

To complement the secondary research and to help validate the need for The Golf Blueprint, I conducted primary research targeting the specific user base at The Kendleshire Golf Club. This research was essential to capture real-world insights from potential users, understand their current course management approaches, and identify specific features that would provide the most value to them.

I utilised a Qualtrics survey as the data collection tool. The survey was distributed between February 15th and March 15th, 2025, using both digital channels (UWE Golf society WhatsApp group) and in-person recruitment at the clubhouse for regular members. This dual-distribution strategy ensured representation across different age groups and technology comfort levels, helping to hopefully avoid sampling bias toward younger golfers at university.

The survey (accessible at: https://uwe.eu.qualtrics.com/jfe/form/SV_5vdeIIMs1zUsifc) was designed following UWE ethical guidelines, with questions carefully crafted to avoid leading participants and to capture both quantitative metrics and qualitative insights. The survey structure addressed:

- Informed consent and participant rights
- Demographic data and playing ability (handicap range, frequency of play)
- Current understanding and application of course management principles
- Usage patterns of existing golf technology solutions
- Interest in The Golf Blueprint concept and data contribution willingness

- Feature preferences and priorities for the proposed application

The survey received responses from 86 members of The Kendleshire Golf Club. The respondents were formed of a diverse range of playing abilities, with most players (55%), having a current handicap of 11-20. The data also highlights that the respondents have good familiarity with the golf course, with 34% of them playing there once per week, and 38% playing there twice a month.

3.2.1 Current Understanding of Course Management

I asked the respondents their current understanding of course management, with five options ranging from ‘Very Good’ to ‘Very Bad’. I wanted to try and gain an understanding of this, as The Golf Blueprint aims to provide its users a much better understanding of the importance of good course management.

In the survey, 80% of respondents report currently having either an ‘Average’, a ‘Bad’ or a ‘Very Bad’ understanding of course management.

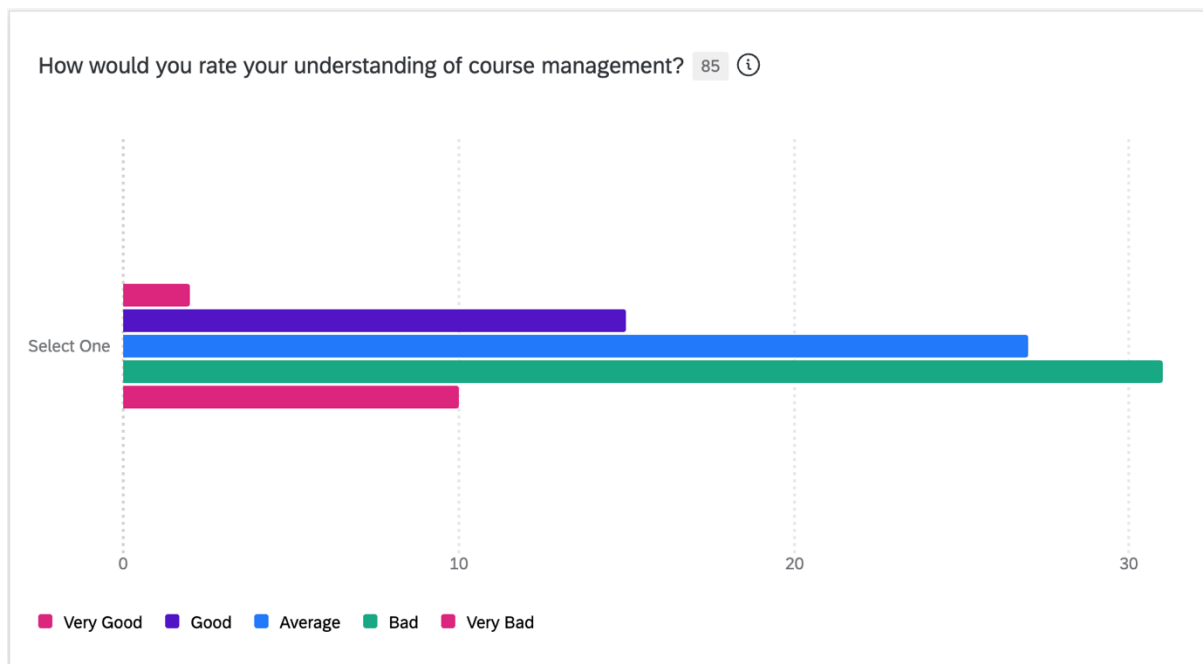


Figure 1.2: Graph showcasing the current understanding of course management

This data showcases a significant gap in current knowledge of course management of players at The Kendleshire. This signifies an opportunity that The Golf Blueprint could address to improve these statistics. As I have identified in the secondary research, employing good course management is an effective technique for improving at golf.

3.2.2 Current Use of Golf Apps or Technology

I asked the respondents whether they currently use any golf related golf applications or technology as a part of their golf game. I asked this question to gain an understanding of whether members at The Kendleshire may be likely or not to adopt a new golfing resource such as The Golf Blueprint, based on their current habits.

In the survey, a slight majority of 52% of respondents answered that they do currently use a golf related app or technology.

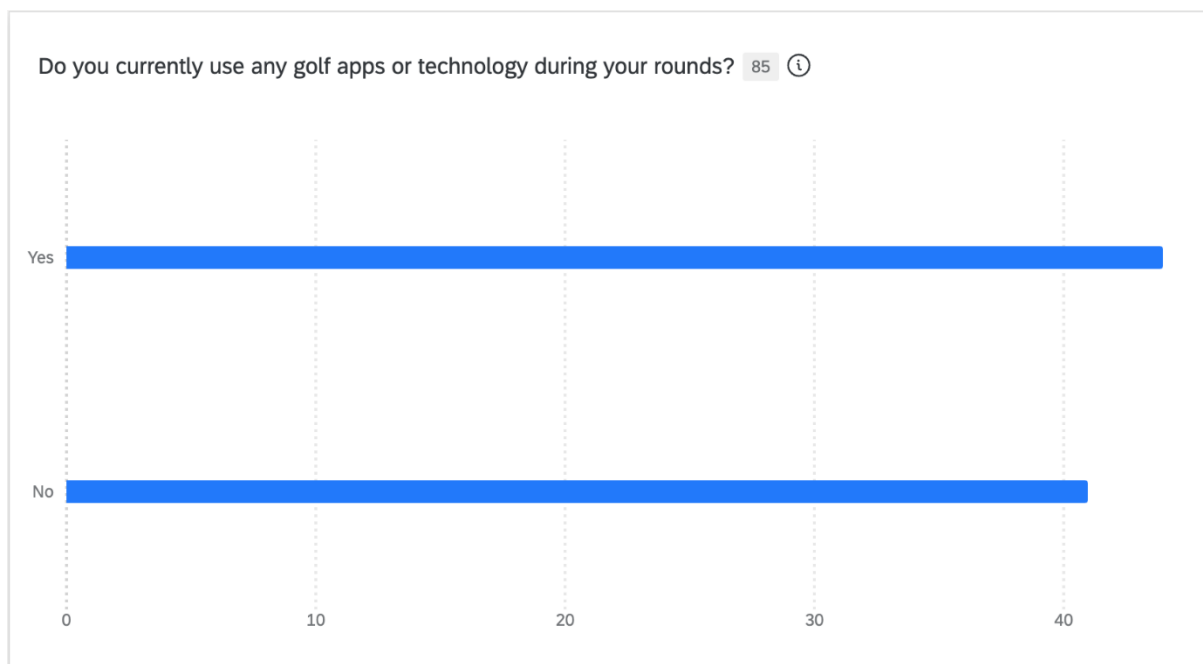


Figure 1.3: Graph showcasing the data of current adoption of golf apps/technology

When asked in detail what specific features of these apps that the respondents find the most useful, 10 respondents (12%) answered that they utilise these apps for getting yardages for their next shots while playing. Also, 21 respondents (25%) replied that they like to use these apps for entering and keeping track of their scores while they are playing.

These findings showcase that there is a significant proportion of golfers at The Kendleshire who regularly use golf applications or technology at the moment. This is positive for The Golf Blueprint, as it shows it could easily become a part of these golfers' routines.

3.2.3 Interest Levels in an App/Website such as The Golf Blueprint

I asked the respondents their interest levels in a resource such as The Golf Blueprint, with four options ranging from 'Very Uninterested' to 'Very Interested'. I wanted to ask this question to directly ask the prospective users whether they think it would be something that they would be likely to adopt.

The question provided very positive results, with 72 respondents (85%), responding with either 'Very Interested' or 'Interested'.

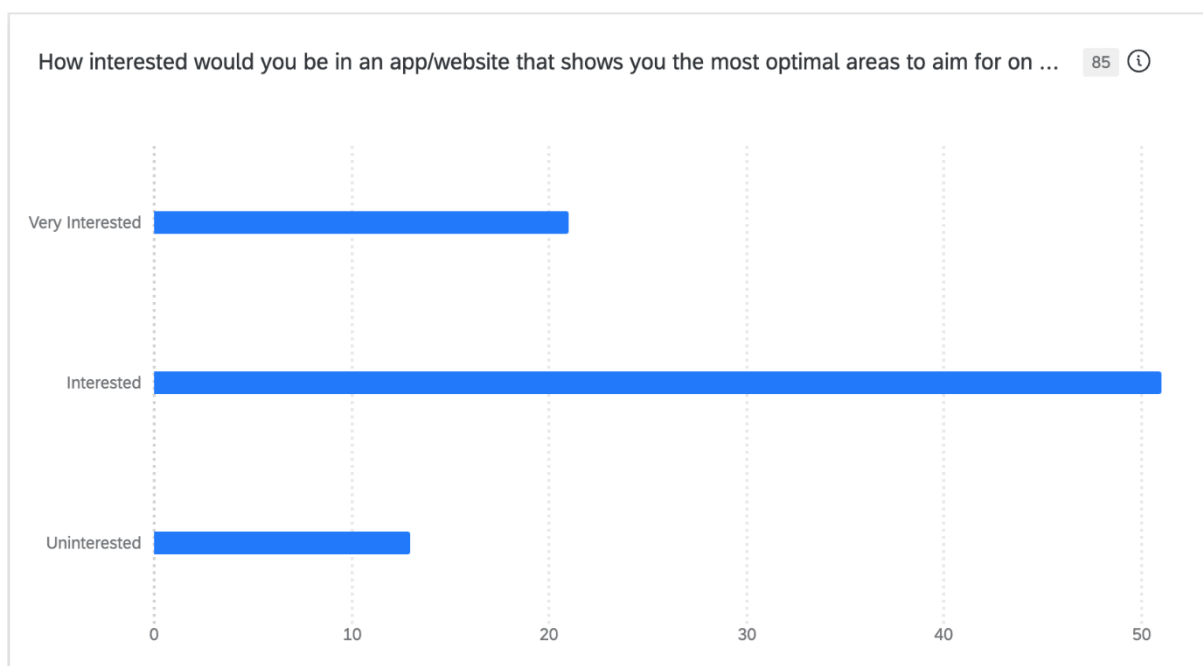


Figure 1.4: Graph showcasing the data of interest levels in The Golf Blueprint.

This data shows a very high level of interest from the respondents towards The Golf Blueprint. This supports my own beliefs that the resource would be very popular.

3.2.4 Willingness to Participate in the Required Data Collection

Finally, I wanted to gain some insight into whether the respondents would be willing to contribute their own data from their rounds of golf towards the resource. This is an important consideration because the success of the resource will rely on users supplying the database with accurate data for trends to begin to form.

This question also provided very positive results, with 75 respondents (88%), saying that they would be willing to provide data to benefit The Golf Blueprint's accuracy and usefulness.

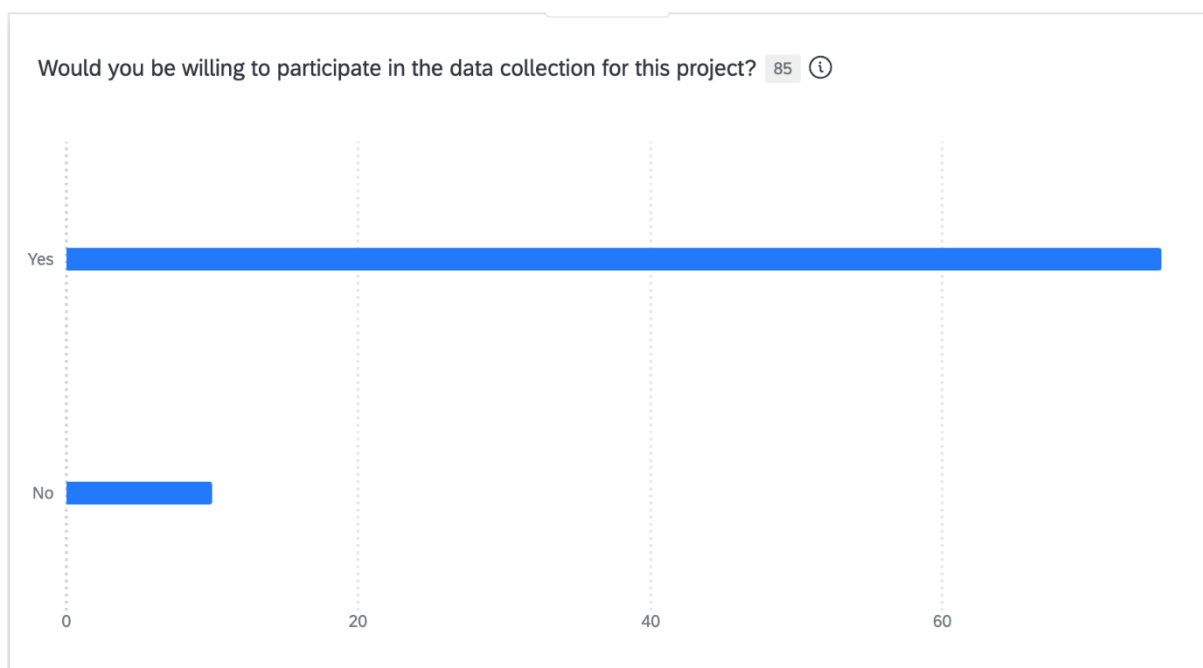


Figure 1.5: Graph showcasing the data of willingness to participate in data collection.

This strongly suggests that the potential future users of The Golf Blueprint would be happy to contribute their own data, which would help to create a sustainable resource.

3.2.5 Desirable Features of The Resource

The final question of the survey asks the respondents specifically what features they would like to see on the website/resource. I wanted to make sure that I was directly asking the target audience of this resource what features may be important to them, to hopefully help me with my design phase of the software development.

The qualitative data gained from this question provided me with the following information around their wishes. 12 respondents (14%) specifically mentioned that they would want the resource to be easy/simple to use. This is a very important consideration for me as I will be targeting a wide age range of users, who may have differing technological capabilities. This has highlighted the necessity for me to create an easy-to-use interface to ensure user satisfaction.

3.3 Key Findings

The comprehensive research conducted through primary and secondary research provide clear validation for The Golf Blueprint concept and helps to give specific direction for its development.

The primary research confirms a significant gap between current golf technology offerings and golfers' strategic needs, with 80% of respondents, regardless of current golfing ability reporting to have an 'Average' or worse understanding of course management. Although 52% of respondents currently use golf technology, these applications focus primarily on distance measurement (12% of users) and score tracking (25% of users), with little support for course management decision-making. This gap directly correlates with the 85% interest level in The Golf Blueprint concept, validating the core premise of the project.

I must make sure that I create a well-designed user interface that is easy to use. I will be relying on the users to input their own golfing data to create as useful a resource as possible, making retaining the users a necessity. The interface must allow for users to quickly enter in their shot data, to prevent it from taking up too much time from their day.

3.3.1 Research Limitations

While the research provides strong validation for The Golf Blueprint concept, some limitations should be acknowledged. Despite efforts to include diverse participants, the sample may over-represent more engaged club members, with 46% of respondents playing at least once a week or more. This could mean that although very keen golfers may be interested in the resource, I might need to adopt a different approach for gaining traction with those who do not play as much.

Another limitation was the number of participants involved in the survey. Although I was able to gather data from 86 members, this is still a small sample size for a club that has over 600 active members. The data that I have collected could fail to represent the views and wishes of the entire member base.

4. Requirements

4.1 User Stories and Use Cases

User stories capture what the users of the resource will want to be able to accomplish. They use the following format: “As a [type of user], I want [an action] so that [benefit/value].”

1. As a golfer who has limited knowledge of course management, I want to be able to see heatmap visualisations of every hole so that I can make better decisions during my rounds.
2. As a regular golfer at The Kendleshire, I want to be able to contribute my own data towards the database so that I can help to build an accurate and useful resource for all to use.

3. As a low-handicap golfer, I want to be able to study the heatmap visualisations so that I can identify any areas of the course that may be beneficial that I have not considered before.
4. As a high-handicap golfer, I want to be able to try and identify which areas of the course are danger areas, so that I can begin to improve my scores when I play.
5. As a golfer who is planning to visit The Kendleshire for a golfing holiday, I want to be able to study the course before I arrive so that I can gain an advantage over my peers.
6. As a golf coach at The Kendleshire, I want to be able to access the heatmap visualisations during my lessons, so that I can give the best advice to students as possible.
7. As a grounds maintenance staff member, I want to be able to see the heatmap visualisations so that we can address whether certain areas are too unfair, or too forgiving.
8. As a new golfer, I am not so interested in course management. I want to be able to enter my scores into the database so that I can track my progress as I improve.

4.2 Use Case Diagrams

4.2.1 System Overview

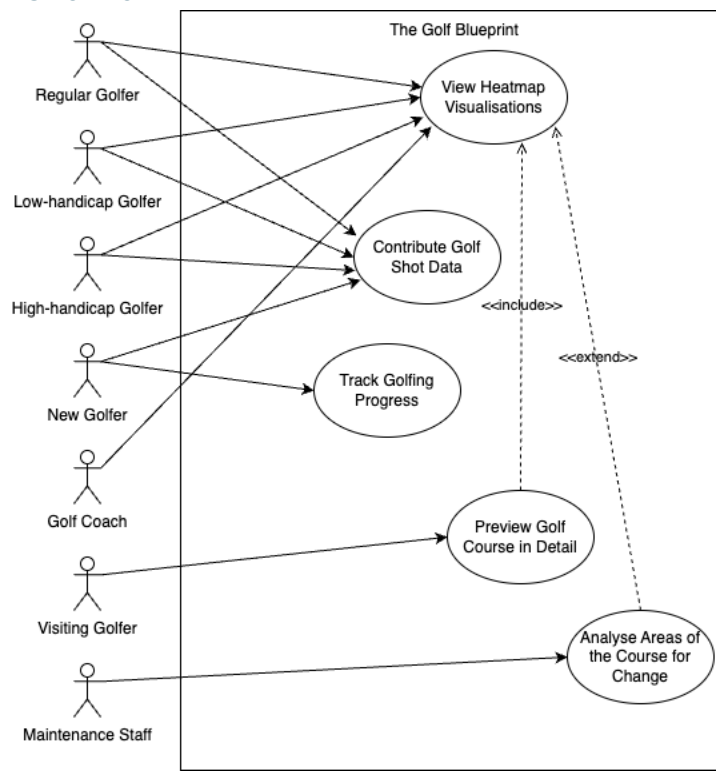


Figure 4.1: The Overview Use Case Diagram

This diagram provides an overview of the entire program. It displays all user types and main functions and how they interact with each other.

4.2.2 Contribute Shot Data

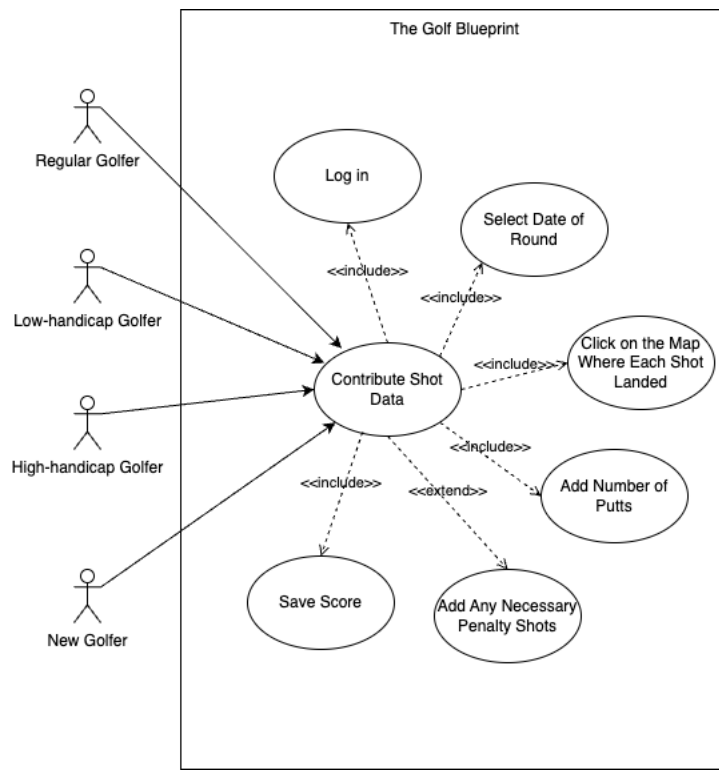


Figure 4.2: The Contribute Shot Data Use Case Diagram

This shows how different types of golfers input their data from their rounds of golf into the system including the required steps of logging in, selecting dates, marking shot locations, and recording scores.

4.2.3 Heatmap Visualisation

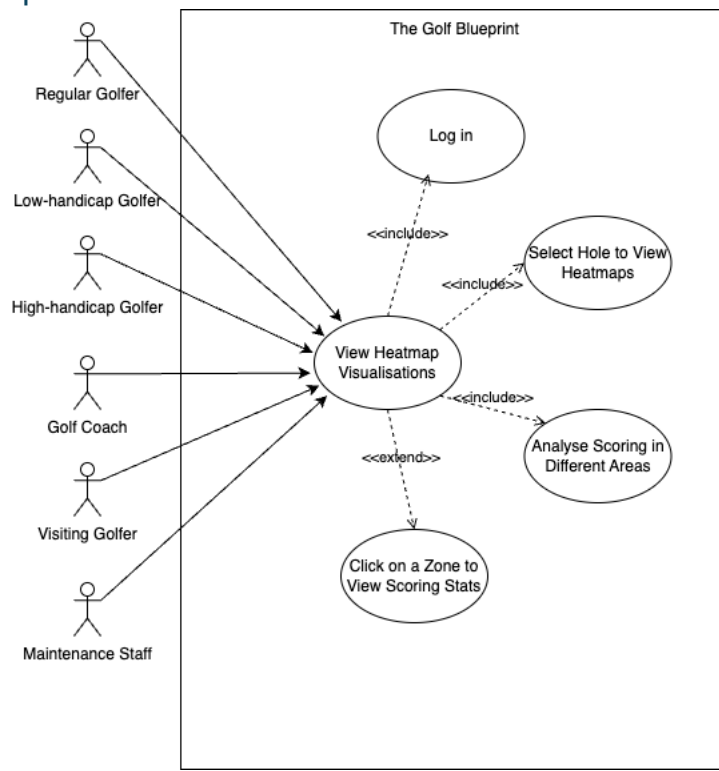


Figure 4.3: The Heatmap Visualisation Use Case Diagram

This diagram shows all of the users who are able to access the heatmap visualisations and the associated functions, including hole selection, and viewing detailed statistical analysis of different areas of the course.

4.2.4 Preview Course

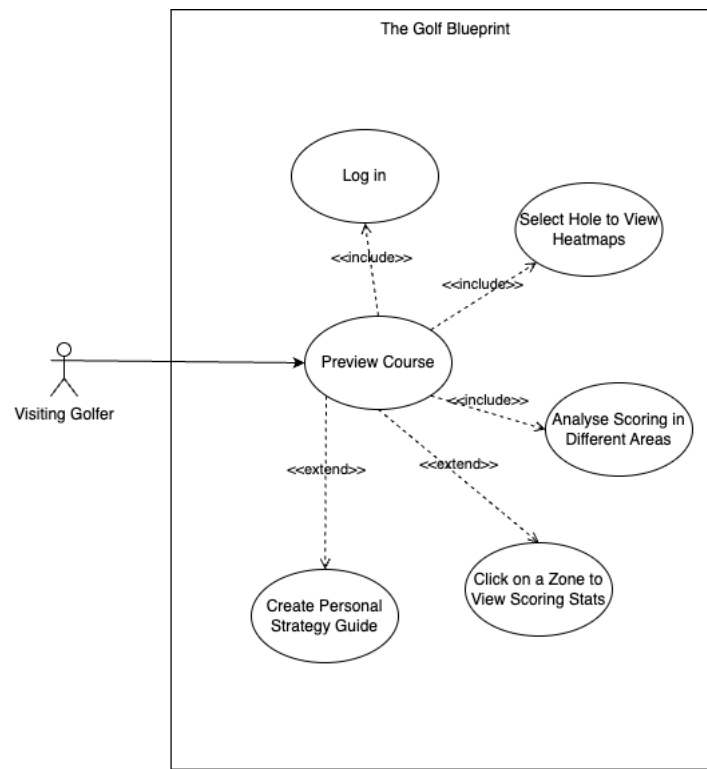


Figure 4.4: The Preview Course Use Case Diagram

This diagram shows how golfers who are planning on visiting The Kendleshire would be able to use the resource to remotely preview the course. It includes viewing hole heatmaps, analysing scoring patterns and creating personalised strategy guides for their visit.

4.2.5 Course Maintenance Analysis

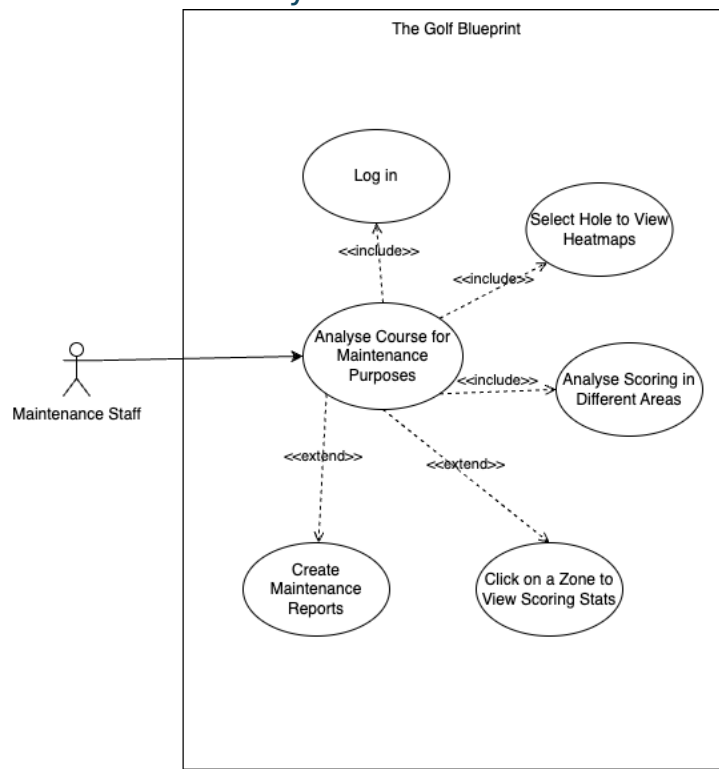


Figure 4.5: The Maintenance Analysis Use Case Diagram

This diagram illustrates how maintenance staff would use the resource to analyse how the course is playing at the moment, with the ability to view heatmaps for each hole, and view detailed statistics for every zone throughout the course, allowing them to create their own maintenance reports to potentially alter the course if required.

4.2.6 Progress Tracking

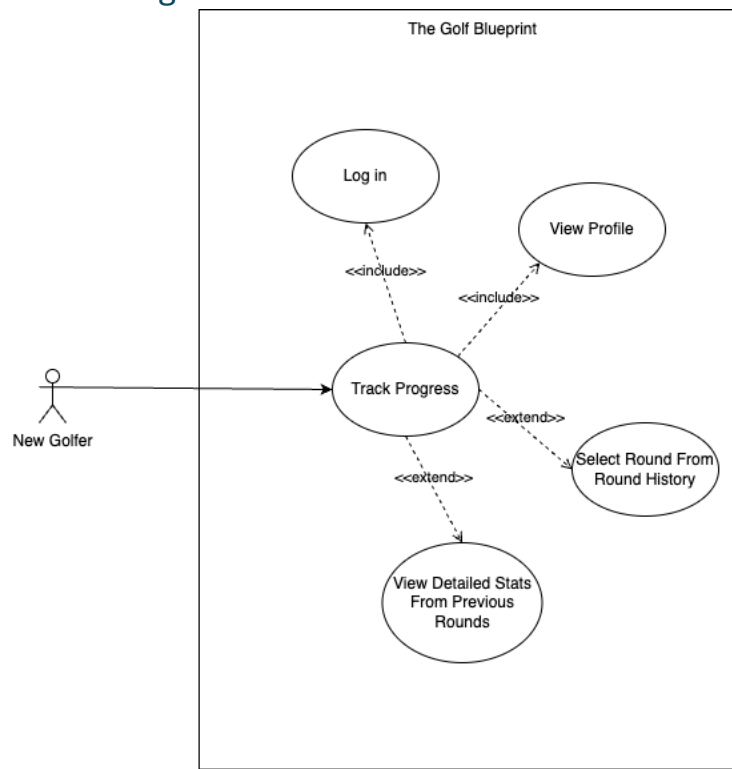


Figure 4.6: The Progress Tracking Use Case Diagram

This diagram shows how a new golfer who is less interested in course management would be able to use the resource in order to track their broader progress as they play more rounds of golf.

4.3 Functional Requirements

This section outlines the functional requirements for The Golf Blueprint, systematically derived from research findings and user stories. These requirements represent specific capabilities the system must provide to deliver value to the users. The requirements are prioritised using the MoSCoW method, using ‘must have’, ‘should have’, ‘could have’, and ‘will not have’ (Brush, 2023).

- **Must have:** Requirements that are critical to project success - these deliver the minimum viable product and address core user needs. Without these, The Golf

Blueprint would fail to deliver its fundamental value proposition of improving course management through data visualization.

- **Should have:** Important requirements that significantly enhance the system but are not absolutely critical. These features provide substantial value and should be included unless they would jeopardize delivery of the "must have" requirements.
- **Could have:** Desirable features that would provide additional value but could be deferred if necessary. These enhancements would improve the user experience but are not essential to achieve the project's primary objectives.
- **Won't have:** Features that were considered but explicitly excluded from the current version. These have been documented to set clear expectations and provide a roadmap for future development.

User Account Management		
ID	PRIORITY	Requirement
FR1	Must have	The system must allow users to create a secure account providing name, email address, and password
FR2	Must have	The system must authenticate users through email and password login
FR3	Must have	The system must store user profiles including username, email and password
FR4	Should have	The system should support password reset functionality
FR5	Should have	The system should support editing account details
FR6	Could have	The system could store more detailed user information such as handicap, playing frequency, and preferred tees

Figure 4.7: Table showcasing the User Account Management Functional Requirements

Data Collection and Input		
ID	PRIORITY	Requirement
FR7	Must have	The system must provide interactive hole visualisations for all 18 holes at The Kendleshire Golf Club
FR8	Must have	The system must allow users to mark precise shot locations on each hole visualisation
FR9	Must have	The system must record the date for each round.
FR10	Must have	The system must allow for the user to enter number of putts and penalty shots.
FR11	Must have	The system must store the shot data in a secure database.
FR12	Must have	The system must validate input data to prevent unrealistic or incorrect entries
FR13	Should have	The system should limit users to entering a maximum of two rounds per calendar day
FR14	Should have	The system should allow users to review and edit their shot data before final submission
FR15	Could have	The system could support bulk data entry for multiple rounds in a single session

Figure 4.8: Table showcasing the Data Collection and Input Functional Requirements

Visualisation and Analytics		
ID	PRIORITY	Requirement
FR16	Must have	The system must generate color-coded heatmaps showing shot distribution patterns for each hole
FR17	Must have	The system must visualize different areas of each hole as optimal, suboptimal, or hazardous
FR18	Must have	The system must provide statistical analysis of scoring outcomes based on shot locations
FR19	Should have	The system should calculate and display average scores for specific areas of each hole
FR20	Could have	The system could generate comparative visualizations between user-specific data and aggregated data

Figure 4.9: Table showcasing the Visualisation and Analytics Functional Requirements

Round History		
ID	PRIORITY	Requirement
FR21	Must have	The system must store and display individual scoring history
FR22	Must have	The system must allow users to view each round specifically
FR23	Must have	The system must allow users to view detailed statistics about each historic round
FR24	Should have	The system should compare user performance against course averages
FR25	Could have	The system could generate achievement badges for round milestones

Figure 4.10: Table showcasing the Round History Functional Requirements

Administration		
ID	PRIORITY	Requirement
FR26	Must have	The system must secure all user data according to GDPR requirements
FR27	Must have	The system should allow administrators to update course visualisations if the course layout changes
FR28	Could have	The system could generate usage statistics to track system adoption

Figure 4.10: Table showcasing the Administration Functional Requirements

Won't Have		
ID	PRIORITY	Requirement
FR29	Won't have	The system won't provide live GPS tracking of users on the course
FR30	Won't have	The system won't integrate with existing golf handicap systems in this version
FR31	Won't have	The system won't include a social networking component for users to connect
FR32	Won't have	The system won't include a mobile app version in the initial release (web responsive only)

Figure 4.11: Table showcasing the Won't Have Functional Requirements

The requirements for The Golf Blueprint were systematically derived through a structured approach that ensured coverage of all users' needs while maintaining traceability to research findings.

Requirements were elicited through multiple complementary techniques:

- **Survey Analysis:** The survey of 86 members at The Kendleshire Golf Club provided quantitative data on user needs, specifically highlighting the desire for improved course management capabilities and ease of use.

- **User Story Development:** Eight detailed user stories were created to capture the diverse perspectives of different user groups, from high-handicap golfers to maintenance staff.
- **Use Case Modelling:** Six detailed use case diagrams were developed to visualize system interactions, helping identify functional boundaries.

4.4 Non-Functional Requirements

Non-functional requirements define the quality attributes of The Golf Blueprint. These requirements are organised according to the ISO/IEC 9126 Software Engineering Product Quality standard (ISO, 2024), ensuring comprehensive coverage of all quality aspects. Like the functional requirements, these are also prioritised using the MoSCoW method.

Usability		
ID	PRIORITY	Requirement
NFR1	Must have	The system must allow new users to complete the shot data entry process for a single hole within 2 minutes after minimal instruction
NFR2	Must have	The system user interface must be accessible and usable by golfers of ages 18+
NFR3	Should have	The system should present information in a clearly visible format suitable for outdoor use in daylight conditions
NFR4	Should have	The system must provide clear visual feedback when marking shot locations
NFR5	Should have	The system should use consistent golf terminology and iconography throughout the interface

Figure 4.12: Table showcasing the Usability Non-Functional Requirements

Functionality		
ID	PRIORITY	Requirement
NFR6	Must have	The system must accurately represent all 18 holes at The Kendleshire Golf Club with correct scaling and landmark placement
NFR7	Must have	The system must generate heatmap visualisations that accurately reflect underlying data patterns
NFR8	Should have	The system should support concurrent use by at least 50 users without performance degradation
NFR9	Should have	The system should function in multiple different window sizes

Figure 4.13: Table showcasing the Functionality Non-Functional Requirements

Reliability and Performance		
ID	PRIORITY	Requirement
NFR10	Must have	The system must handle temporary loss of internet connectivity by storing data locally until reconnection
NFR11	Must have	The system must process shot data input within 1 second of submission
NFR12	Should have	The system should generate heatmap visualisations within 5 seconds of request
NFR13	Should have	The system should load hole visualisations within 3 seconds on standard 4G connections
NFR14	Could have	The system could support database queries for up to 10,000 rounds of golf without performance degradation

Figure 4.14: Table showcasing the Reliability and Performance Non-Functional Requirements

Maintainability		
ID	PRIORITY	Requirement
NFR15	Must have	The system must log all system errors with contextual information for troubleshooting
NFR16	Must have	The system must be built using widely supported web technologies to ensure long-term maintainability
NFR17	Should have	The system should support course visualisation updates without requiring significant code changes
NFR18	Should have	The system should include comprehensive documentation for all data structures and code

Figure 4.15: Table showcasing the Maintainability Non-Functional Requirements

Security		
ID	PRIORITY	Requirement
NFR19	Must have	The system must implement GDPR-compliant data protection measures
NFR20	Must have	The system must require secure authentication for accessing user data
NFR21	Must have	The system must provide users with the ability to download or delete their personal data

Figure 4.16: Table showcasing the Security Non-Functional Requirements

5 Software Design

5.1 Software Architecture

5.1.1 Software Architectural Pattern

The Golf Blueprint implements a client-server architecture following the Model-View-Controller (MVC) pattern to ensure separation of concerns and maintainability. This architecture consists of three primary components:

1. **Client-Side Application (View and Controller)** – A responsive HTML/CSS/JavaScript web application that provides the user interface and handles user interactions
2. **Server-Side API (Controller)** – A Node.js Express server that processes requests and mediates between the client and database
3. **Database (Model)** – A MySQL database that stores all persistent data

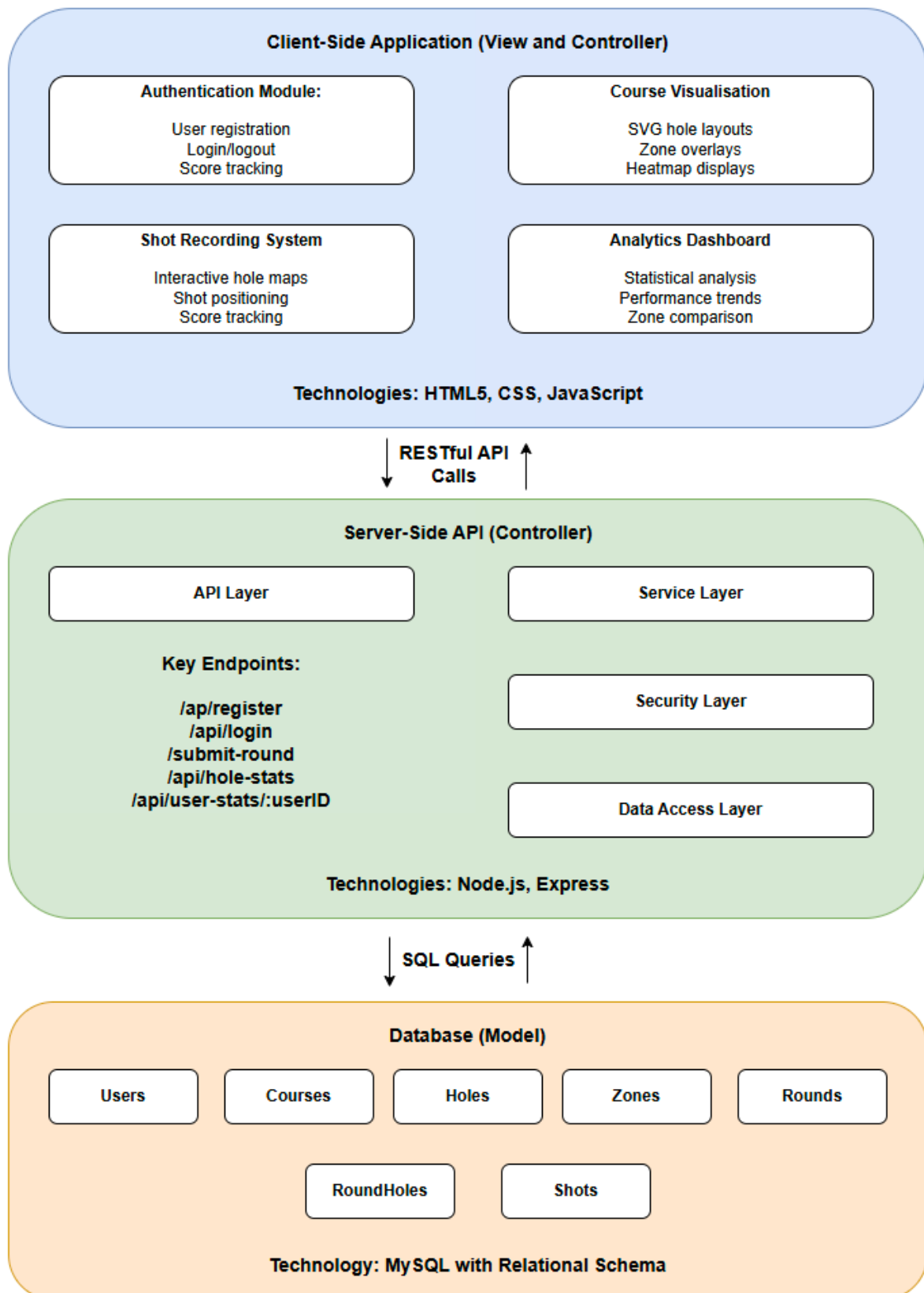


Figure 5.1: This image showcases the software architecture diagram for The Golf Blueprint

5.1.2 Client-Side Architecture

The client-side application follows a component-based design pattern with the following components:

1. Authentication – Manages user login, registration, and session maintenance using localStorage for client-side management
2. Hole Visualisation – Renders hole layouts using SVG images, with SVG zones overlaid to allow users to interact with each hole and its unique features independently
3. Shot Recording System – Allows for users to enter in the exact location of their golf shots
4. Analytics Dashboard – Processes and displays statistical data from the database in the form of heatmaps

The client-side technology stack includes the following:

- HTML5 for website structure
- CSS for responsive styling
- JavaScript for user interactivity

5.1.3 Server-Side Architecture

The server-side implementation uses Node.js with Express to provide RESTful API endpoints. The server architecture includes the following:

1. API Layer – Express routes that handle the HTTP requests and responses
2. Service Layer – Business logic for processing data and implementing rules
3. Data Access Layer – Database connection management and query execution
4. Security Layer – Authentication, authorisation and data protection

The client and server architecture was specifically designed to support the functional requirements FR8 and FR11, surrounding entering golf shot data into the database successfully.

5.1.4 Communication Protocol

The client and server communicate using a RESTful API pattern over HTTP:

Endpoint	Method	Purpose
/api/register	POST	Create new user accounts
/api/login	POST	Authenticate users
/api/submit-round	POST	Save completed round data
/api/hole-stats	GET	Retrieve statistical data for analysis
/api/round-count	GET	Get count of rounds for a specific hole
/api/user-stats/:userId	GET	Retrieve user performance statistics
/api/user-rounds/:userId	GET	List rounds completed by a user
/api/round-scorecard/:roundId	GET	Get detailed scorecard for a specific round

Figure 5.2: Table showcasing the communication protocol

5.2 Database Design

5.2.1 Database Schema

The Golf Blueprint utilises a relational MySQL database with a normalised schema to help support the data storage and interpretation requirements, while maintaining data integrity.

The schema consists of the following:

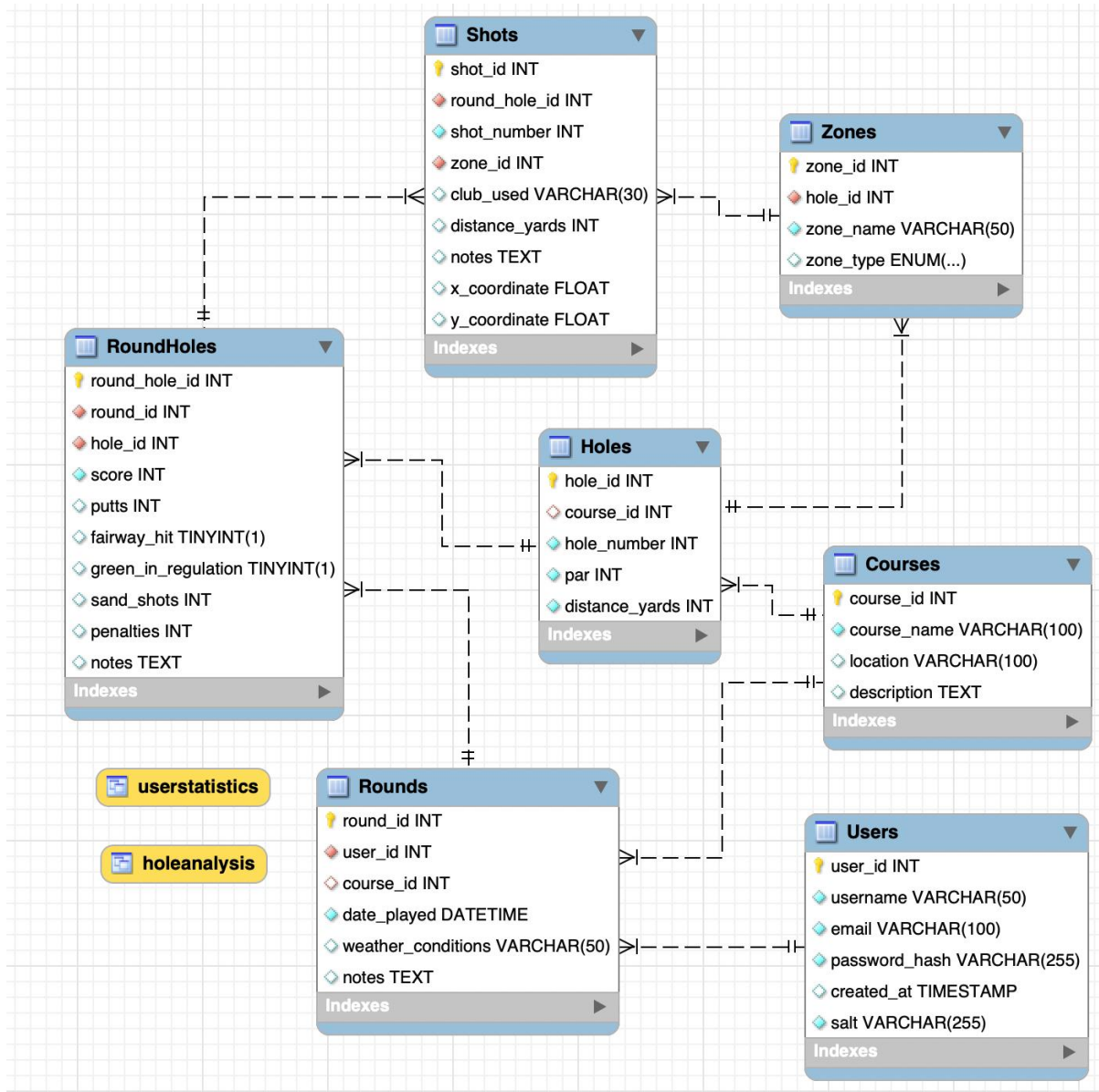


Figure 5.3: Entity Relationship diagram for the database

The database for the system was designed to specifically meet the functional requirements FR8, FR9, FR10, FR11, FR16 and FR18.

5.3 User Interface Design

5.3.1 Design Principles

The user interface must be designed in such a way that it would allow for users of all technological ability to quickly and accurately input their round data into the secure database,

allowing for the most useful data to be produced. The design must also take into consideration how the users will then view the data that is stored in the database in the form of heatmaps. I will be sticking to the following principles:

1. **Simplicity** – Clean, uncluttered layout that focus on the task at hand
2. **Consistency** – Uniform elements and patterns across all pages
3. **Feedback** – Clear visual cues for actions and state changes
4. **Accessibility** – Readable text, sufficient contrast and intuitive controls

These principles directly align with the findings from my primary and secondary research, as well as the functional and non-functional requirements.

5.3.2 Colour Palette

I will be using a colour palette that will give the resource a balance between professional aesthetics and a golf-themed visual identity.

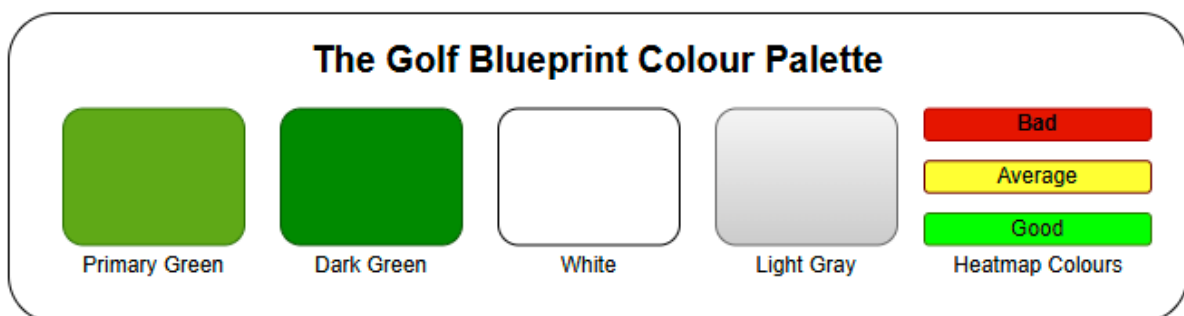


Figure 5.4: Image showcasing the colour palette for the design

The primary green colour is a bold, but not too striking colour that will be familiar to many who play golf. Helping to connect the design to the purpose of the resource. This is complemented by the darker green colour, which will be used for navigation elements and buttons, providing a proficient level of contrast for accessibility. The interface will also utilise white and light grey backgrounds to ensure optimal readability of the information.

Also, a combination of red, yellow and green will be used for displaying the data in the form of heatmaps. This will make the resource very easy to use for all users, providing a familiar combination of colours for 'Good', 'Average' and 'Bad' zones.

5.3.3 Wireframes

Homepage

The homepage presents a clean, focused design with a concise proposition statement and a single button which will allow the user to easily navigate to the page that lets them record a round. This page will welcome users to the website and will allow them to navigate to other pages quickly and intuitively. Each page uses the same navigation bar, allowing for users to transition between pages with ease.

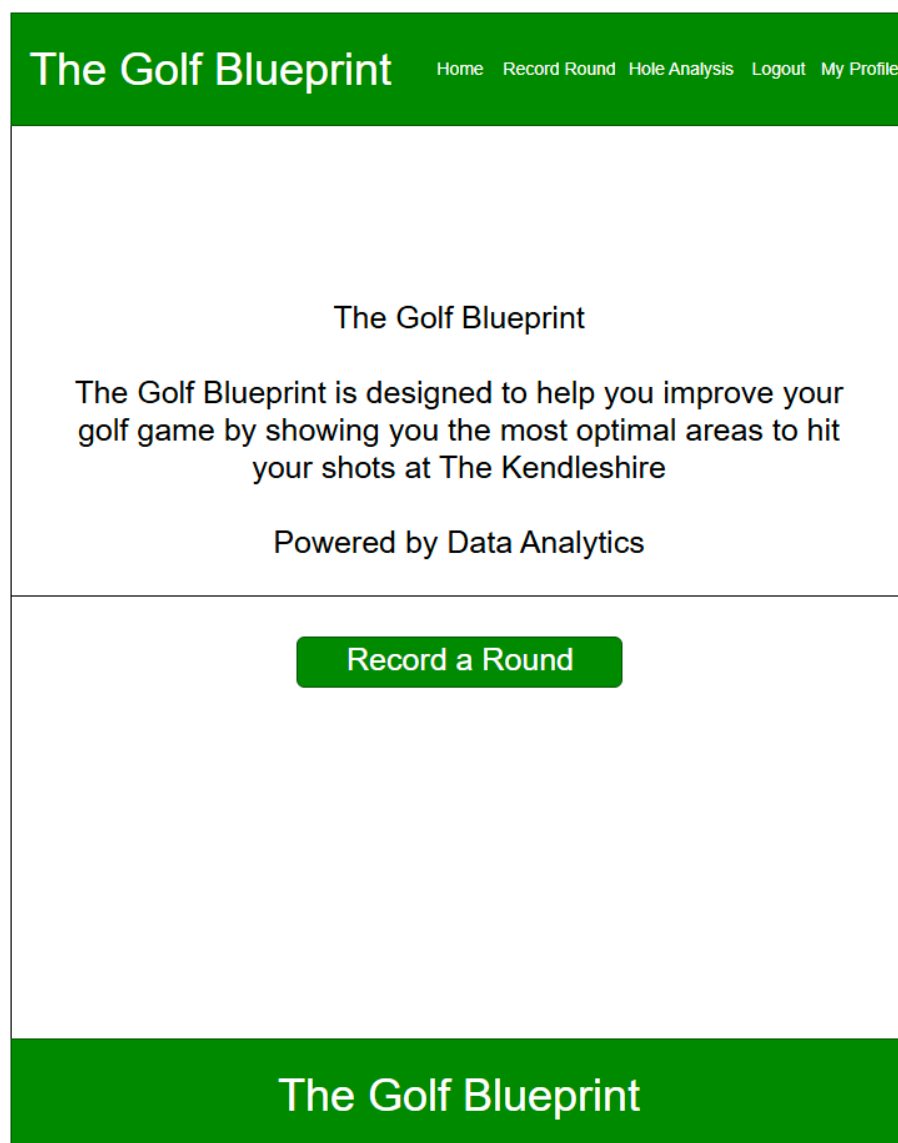


Figure 5.5: Wireframe showcasing the Homepage of The Golf Blueprint

Login and Register Page

The Login and Register pages will have a very clean design, with very straightforward and recognisable forms, only containing essential fields and clear labelling. This will allow for quick access to the system, requiring minimum interaction, hopefully averting some of the previous noted risks associated with user-adoption.

The wireframe shows a web page for 'The Golf Blueprint'. At the top is a green header bar with the site name 'The Golf Blueprint' on the left and navigation links 'Home' and 'Login' on the right. The main content area is white and contains a central rounded rectangle with a green border. Inside this rectangle, the word 'Register' is displayed in green. Below it are four input fields, each with a label above it: 'Username', 'Email', 'Password', and 'Confirm Password'. After the last input field is a solid green button with the word 'Register' in white. At the bottom of the rounded rectangle, the text 'Already have an account? Login here' is displayed.

Figure 5.6: Wireframe showcasing the Register page

The Golf Blueprint

HomeLogin

Login

Email

Password

Login

Don't have an account?
Register Here

Figure 5.7: Wireframe showcasing the Login page

Round Details Page

The Round Details page also has an intentionally minimal design, including the course name (The Kendleshire), and the date that the round was played on. When clicking on the data played box, it will open a calendar style window, allowing for the user to quickly enter what day they played, also helping to minimise any errors when entering the data. Finally, the page has a prominent 'Begin Scoring' button, which will let the users quickly get onto the next page where they can begin to enter their data.

The wireframe shows a web page for 'The Golf Blueprint'. At the top is a green header bar with the site name 'The Golf Blueprint' in white, followed by navigation links: 'Home', 'Record Round', 'Hole Analysis', 'Logout', and 'My Profile'. The main content area is white and features a green heading 'Record a Round'. Below this is a rounded rectangular form containing three elements: a label 'Course Name:' above a text input field, a label 'Date Played:' above a date input field with the placeholder 'DD/MM/YYYY', and a green 'Begin Scoring' button. The page is framed by a green footer bar at the bottom with the text 'The Golf Blueprint' in white.

Figure 5.8: Wireframe showcasing the Round Details page

Shot Input Page

The shot input page is an extremely important page for the resource. This page is where the user will input their shots from their round of golf. In order for the resource to collect the most accurate data possible, it is essential that this page is well designed to allow the users to know exactly what they are doing when using it. It features a box which gives the details of the hole they are currently scoring, and provides concise, but clear instructions on how to use the page.

The wireframe shows a web page for 'The Golf Blueprint'. The top navigation bar is green with white text for the site name and links to Home, Record Round, Hole Analysis, Logout, and My Profile. The main content area is divided into two columns. The left column contains a summary box for 'Hole 1' (Par 4, 295 Yards), followed by instructions to click on the hole and add strokes. It includes a 'Shot Information' section with a 'Shot Counter' and an 'Undo Last Shot' button, a 'Putts' section with '+' and '-' buttons, and a 'Penalty Shots' section with an 'Add Penalty Shot' button. The right column is a large rounded rectangle labeled 'SVG Image of Hole 1'. The bottom of the page features a green footer with the site name.

The Golf Blueprint

[Home](#) [Record Round](#) [Hole Analysis](#) [Logout](#) [My Profile](#)

Hole 1

Par 4 **295 Yards**

Hole 1

Click on the hole exactly where your shots landed.

Add putts and penalty strokes separately using the buttons below

Shot Information
Shot Counter:

Undo Last Shot

Putts:

+ -

Penalty Shots

Add Penalty Shot

SVG Image of Hole 1

The Golf Blueprint

Figure 5.9: Wireframe showcasing the Shot Input page

Round Summary Page

The round summary page provides the user with a quick glance overview of their entire round, using a simple grid layout allowing them to see their scores for each hole. The page will also show their total final score at the bottom of the page, allowing them to ensure that it is accurate before submitting the round.

The Golf Blueprint

HomeRecord RoundHole AnalysisLogoutMy Profile

Round Summary

Hole 1 Score:	Hole 2 Score:	Hole 3 Score:	Hole 4 Score:
Hole 5 Score:	Hole 6 Score:	Hole 7 Score:	Hole 8 Score:
Hole 9 Score:	Hole 10 Score:	Hole 11 Score:	Hole 12 Score:
Hole 13 Score:	Hole 14 Score:	Hole 15 Score:	Hole 16 Score:
Hole 17 Score:		Hole 18 Score:	

Total Score:

Submit Round

The Golf Blueprint

Figure 5.10: Wireframe showcasing the Round Summary page

Hole Analysis Page

The hole analysis page is the most important page within the entire system. It is where the findings of the collection of data will be displayed back to the users. This page is where the primary aim for the project will be achieved. Because of this importance, it is essential that it is a well-designed page, allowing the users to see how it could be possible to improve their golf scores through course management. The page uses a two-column layout, with the interactive hole visualisation heatmap on the left, and the statistical breakdown for the hole on the right, using colour-coded zones to identify the optimal or sub-optimal landing areas.

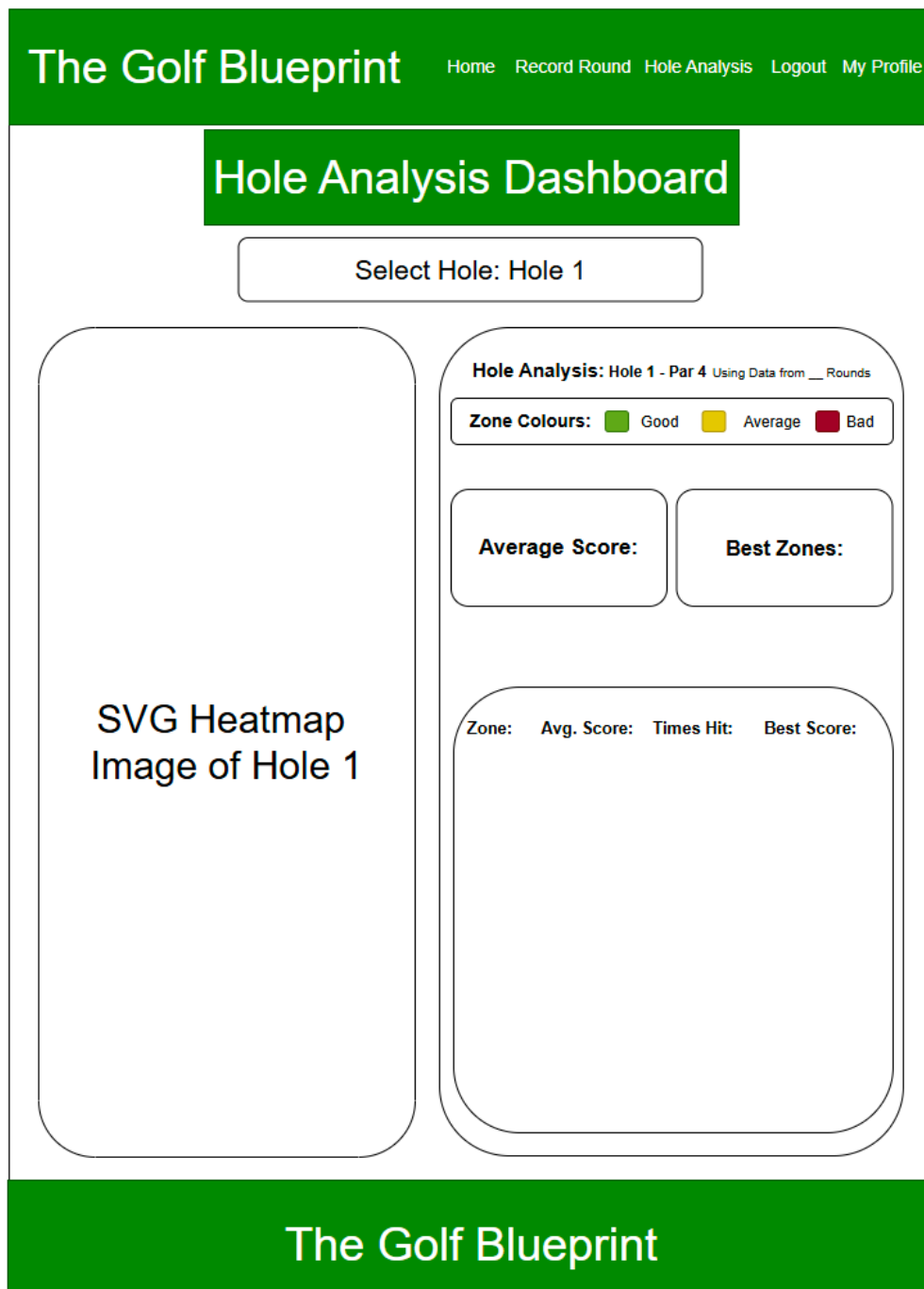


Figure 5.11: Wireframe showcasing the Hole Analysis page

Profile Page

The profile page presents a minimalist design which focuses on key performance metrics of number of rounds played, and the user's best recorded score. This is followed by recent round history, with each one having a prominent 'view' button to allow for easy access to the scorecards related to that specific round.

The wireframe shows a profile page for 'The Golf Blueprint'. At the top is a green header with the site name and navigation links. The main content area includes a profile picture placeholder, account details, performance metrics, and a list of recent rounds with 'View' buttons. A green footer contains the site name again.

The Golf Blueprint Home Record Round Hole Analysis Logout My Profile

Profile Picture

Account Name:

Email Address:

Rounds Played:

Best Score:

Recent Rounds:

DD/MM/YYYY	Score:	View
DD/MM/YYYY	Score:	View
DD/MM/YYYY	Score:	View
DD/MM/YYYY	Score:	View

The Golf Blueprint

Figure 5.12: Wireframe showcasing the Profile page

Scorecard Page

The scorecard page utilises a traditional golf scorecard layout familiar to all golfers, with hole-by-hole scores displayed alongside more detailed statistics such as number of pars and birdies. The design is simple and easy to read, allowing users to quickly analyse their previous rounds.

The Golf Blueprint

[Home](#) [Record Round](#) [Hole Analysis](#) [Logout](#) [My Profile](#)

Round Scorecard

Date: DD/MM/YYYY
Course: The Kendleshire

Score:

Pars: __ Birdies: __
Bogeys: __ Double Bogey+: __

Hole	Par	Score	Total
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

Back to Profile

The Golf Blueprint

Figure 5.13: Wireframe showcasing the Round Scorecard page

The wireframes for the design of the system were specifically designed in order to meet the functional requirements FR1, FR2, FR3, FR7, FR21, FR22, FR23 and the non-functional requirements NFR2 and NFR3.

5.4 Key Algorithms and Technical Components

5.4.1 Shot Tracking Algorithm

The shot-tracking system uses the following code to accurately record the users' shots:

1. Capture click coordinates relative to the hole container with the SVG image inside it
2. Determine the specific zone that was clicked on
3. Create a shot object with position and zone information
4. Add the shot marker to the visualisation for the user to check it was the correct location
5. Update the shot counter and score totals
6. Store the shot data in the client's local storage

This algorithm allows for the users to accurately and easily record where their shots finished in a systematic way, allowing for easy data analysis, directly supporting the core objective of the project. It was developed to directly align with the functional requirements FR8, FR10 and FR11.

```

/**
 * Shot Tracking System for The Golf Blueprint
 * Core implementation of the interactive shot data collection
 */

// Track shot data and user interactions
let shotCount = 0;
let puttCount = 0;
let penaltyCount = 0;
let shots = [];
let actionHistory = [];

/**
 * Records a shot when user clicks on the hole visualisation
 */
function recordShot(event) {
    // Calculate exact coordinates where user clicked
    const container = document.getElementById('fullHoleView');
    const rect = container.getBoundingClientRect();
    const x = event.clientX - rect.left;
    const y = event.clientY - rect.top;

    // Identify which course zone was clicked (fairway, rough, etc.)
    const zone = getZoneAtPosition(event.target);

    // Create shot data object with complete information
    const shot = {
        number: ++shotCount,
        x: x,
        y: y,
        zone: zone,
        xPercent: (x / rect.width) * 100, // For responsive scaling
        yPercent: (y / rect.height) * 100,
        type: 'shot'
    };

    // Store shot data and update history
    shots.push(shot);
    actionHistory.push({ type: 'shot', shotIndex: shots.length - 1 });

    // Create visual marker on the hole visualisation
    addShotMarker(shot);

    // Update UI and save data
    updateDisplay();
    saveHoleData();
}

```

Figure 5.14: First code snippet showcasing the shot tracking system

```

/**
 * Creates visual indication of shot location
 */
function addShotMarker(shot) {
    const container = document.getElementById('fullHoleView');

    // Create numbered marker element
    const marker = document.createElement('div');
    marker.className = 'shot-marker';
    marker.textContent = shot.number;

    // Position precisely where user clicked
    marker.style.left = shot.x + 'px';
    marker.style.top = shot.y + 'px';

    container.appendChild(marker);
}

/**
 * Saves shot data to persistent storage
 */
function saveHoleData() {
    const holeNumber = getHoleNumber();

    // Prepare complete hole data
    const holeData = {
        holeNumber: holeNumber,
        shotCount: shotCount,
        puttCount: puttCount,
        penaltyCount: penaltyCount,
        holeScore: shotCount + puttCount + penaltyCount,
        shots: shots,
        actionHistory: actionHistory
    };

    // Save to browser's localStorage
    let roundData = JSON.parse(localStorage.getItem('currentRound') || '{}');
    roundData[`hole${holeNumber}`] = holeData;
    localStorage.setItem('currentRound', JSON.stringify(roundData));
}

// Initialise tracking system when page loads
document.addEventListener('DOMContentLoaded', function() {
    // Load existing data if available
    loadHoleData();

    // Set up click handler for the hole visualisation
    const holeView = document.getElementById('fullHoleView');
    if (holeView) {
        holeView.addEventListener('click', recordShot);
    }

    updateDisplay();
});

```

Figure 5.14: Second code snippet showcasing the shot tracking system

5.4.2 Data Visualisation Algorithm

The heatmap visualisation which is used for shot analytics uses the following process:

1. Fetches zone performance data from the API
2. Calculates performance metrics for each zone (relative to median)
3. Apply appropriate colour coding based on performance
4. Adjust colour opacity depending on magnitude of difference
5. Apply styles to the zone elements in the SVG visualisation

This algorithm was specifically developed to incorporate the functional requirements FR16, FR17, FR18, FR19.

```

/**
 * Analytics Heatmap System for The Golf Blueprint
 */

// Statistical visualisation algorithm for golf course zones
function updateZonePerformanceVisualization(zoneData, holeNumber) {
  const container = document.getElementById('hole-visualization');

  // Reset previous visualisation state
  container.querySelectorAll('.zone').forEach(zone => {
    zone.style.backgroundColor = '';
    zone.style.opacity = '0';
    zone.style.display = 'none';
  });

  if (!zoneData || zoneData.length === 0) return;

  // Calculate statistical reference points
  const scores = zoneData.map(zone => zone.avgScore);
  const minScore = Math.min(...scores);
  const maxScore = Math.max(...scores);

  // Use median as performance baseline
  scores.sort((a, b) => a - b);
  const medianScore = scores.length % 2 === 0
    ? (scores[scores.length/2 - 1] + scores[scores.length/2]) / 2
    : scores[Math.floor(scores.length/2)];

  // Apply performance-based color coding to each zone
  zoneData.forEach(zone => {
    // Handle different zone naming conventions between front/back nine
    const zoneId = parseInt(holeNumber) < 10
      ? zone.zone.toLowerCase().replace(/ /g, '-') // Front nine uses hyphens
      : zone.zone.toLowerCase().replace(/ /g, ''); // Back nine has no hyphens

    const zoneElement = document.getElementById(zoneId);

    if (zoneElement) {
      // Color based on performance relative to median (threshold = 15% of range)
      const scoreDiff = zone.avgScore - medianScore;
      const threshold = (maxScore - minScore) * 0.15;

      // Apply color coding based on statistical performance
      if (zone.avgScore < medianScore - threshold) {
        // Better than average - green with opacity based on how much better
        const opacity = 0.5 + Math.min(0.4, Math.abs(scoreDiff / threshold) * 0.4);
        zoneElement.style.backgroundColor = `rgba(59, 243, 46, ${opacity})`;
      } else if (zone.avgScore > medianScore + threshold) {
        // Worse than average - red with opacity based on how much worse
        const opacity = 0.6 + Math.min(0.3, Math.abs(scoreDiff / threshold) * 0.4);
        zoneElement.style.backgroundColor = `rgba(255, 102, 102, ${opacity})`;
      } else {
        // Average performance - orange
        zoneElement.style.backgroundColor = `rgba(255, 152, 0, 0.5)`;
      }

      // Make zone visible and add tooltip with statistics
      zoneElement.style.display = '';
      zoneElement.style.opacity = '1';
      zoneElement.title = `${zone.zone}: Avg. Score ${zone.avgScore.toFixed(1)} (${zone.count} shots)`;

      // Add interactivity for user exploration
      zoneElement.addEventListener('click', () => highlightTableRow(zone.zone));
    }
  });
}

// API interaction to fetch performance data
function fetchHoleData(holeNumber) {
  // Fetch statistical data for the selected hole
  fetch(`/api/hole-stats?hole=${holeNumber}&shot=all&datePeriod=all-time`)
    .then(response => response.json())
    .then(data => {
      if (data?.zones?.length > 0) {
        // Process data into visualisations
        updateZonePerformanceVisualization(data.zones, holeNumber);
        updateZoneStatsTable(data.zones, holeNumber);
        updateScoreChart(data.zones, holeData[holeNumber].par);
      } else {
        // Handle no data case
        document.getElementById('score-chart').innerHTML =
          '<div class="loading-placeholder"><p>No data available for this hole</p></div>';
      }
    })
    .catch(error => {
      console.error('Error fetching data:', error);
    });
}

// Initialise dashboard with event listeners
document.addEventListener('DOMContentLoaded', function() {
  document.getElementById('hole-select').addEventListener('change', updateDashboard);
  updateDashboard(); // Load initial hole data
});

```

Figure 5.15: Code snippet showcasing the heatmap data visualisation

5.4.3 Security Implementation

Security is implemented into the system through multiple layers:

1. **Protected Route Enforcement** – the ‘checkPageAccess()’ function validates whether a user is authorised to access certain pages:
 - Identifies protected pages (round details, shot analysis, hole pages)
 - Verifies authentication status before allowing access
 - Redirects unauthorised users to the login page

```
/**
 * Client-side security implementation for The Golf Blueprint
 * This module handles authentication verification and protected route access
 */

/**
 * Check if current page requires authentication and enforce access control
 */
function checkPageAccess() {
  // Determine current page path
  const currentPath = window.location.pathname;

  // Define protected pages that require authentication
  const isRoundDetails = currentPath.includes('round-details.html');
  const isShotAnalysis = currentPath.includes('shot-analysis.html');
  const isHolePage = currentPath.includes('/holes/hole');

  // These pages require authentication
  const requiresAuth = isRoundDetails || isShotAnalysis || isHolePage;

  if (requiresAuth) {
    // Check if user is logged in by verifying user data exists
    const userData = localStorage.getItem('user');

    if (!userData) {
      // Security measure: User is not authenticated but trying to access protected page
      // Display error and prevent access to protected content
      displayAuthError();
      return false;
    }
  }

  return true;
}
```

Figure 5.16: Code snippet showcasing the ‘checkPageAccess()’ function

2. • **Authentication State Management** - The ‘updateAuthNavigation()’ function:
 - Checks for valid session data in localStorage
 - Includes error handling to prevent security issues with malformed data
 - Updates UI elements based on authentication status

```

/**
 * Updates UI elements based on authentication status
 */
function updateAuthNavigation() {
  const authLinkElement = document.getElementById('auth-link');
  const profileLinkElement = document.getElementById('profile-link');

  // Get protected feature navigation elements
  const recordRoundLinks = document.querySelectorAll('.nav-links a[href="round-details.html"], .nav-links a[href="../round-details.html"]');
  const shotAnalysisLinks = document.querySelectorAll('.nav-links a[href="shot-analysis.html"], .nav-links a[href="../shot-analysis.html"]');

  // Check authentication status
  const userData = localStorage.getItem('user');

  if (userData) {
    try {
      // Parse user data to verify it's valid
      const user = JSON.parse(userData);

      // Update UI for authenticated user
      if (authLinkElement) {
        authLinkElement.innerHTML = '<a href="#" class="nav-link" onclick="logout()">Logout</a>';
      }

      // Show protected navigation links for authenticated users only
      profileLinkElement.style.display = 'block';
      recordRoundLinks.forEach(link => {
        if (link.parentElement) link.parentElement.style.display = 'block';
      });

      shotAnalysisLinks.forEach(link => {
        if (link.parentElement) link.parentElement.style.display = 'block';
      });
    } catch (error) {
      // Security measure: Invalid user data format in localStorage
      // Treat as unauthenticated to prevent security issues
      handleLoggedOutState(authLinkElement, profileLinkElement, recordRoundLinks, shotAnalysisLinks);
    }
  } else {
    // User is not authenticated - restrict access to protected features
    handleLoggedOutState(authLinkElement, profileLinkElement, recordRoundLinks, shotAnalysisLinks);
  }
}

```

Figure 5.17: Code snippet showcasing the ‘updateAuthNavigation()’ function

3. • Secure Logout Process - The ‘logout()’ function:
 - Removes authentication data from browser storage
 - Updates UI to reflect logged-out state
 - Redirects to public area after logout


```

/**
 * Secure logout implementation
 */
function logout() {
    // Remove authentication data from browser storage
    localStorage.removeItem('user');

    // Update UI to reflect unauthenticated state
    updateAuthNavigation();

    // Provide user feedback
    alert('You have been logged out successfully.');
```

```

    // Security measure: Redirect to public area after logout
    let redirectPath = 'index.html';
    if (window.location.pathname.includes('/holes/')) {
        redirectPath = '../index.html';
    }

    window.location.href = redirectPath;
    return false;
}

// Initialise security checks when page loads
document.addEventListener('DOMContentLoaded', function() {
    // Apply authentication-based UI updates
    updateAuthNavigation();

    // Enforce access control for protected pages
    checkPageAccess();
});

```

Figure 5.18: Code snippet showcasing the 'logout ()' function

The code prevents unauthorised users from seeing protected features by:

- Hiding navigation links to protected features
- Displaying appropriate login/logout options

These security related implementations incorporate the functional requirements FR1, FR2 and FR3

6. Results

6.1 Implementation

The Golf Blueprint was successfully implemented as a fully working web application with all of the core functionality specified in the requirements. This section presents the key implantation outcomes, demonstrating how the produced software has met the objectives and requirements.

6.1.1 SVG Image Creation

An important first step for the implementation of the system, was recreating all of the 18 individual holes at The Kendleshire as SVG files. This is what allows the user to interact with the specific holes, by breaking each hole down into hole specific zones/areas. Allowing the user to interact with these zones is necessary for collecting, storing, and reproducing accurate golf shot data from each hole, in order to generate meaningful heatmap visualisations, and in turn course management recommendations, as set out in the requirements.

In order to recreate each hole as accurately as possible, I gathered satellite imagery of the holes using Google Earth and traced each hole on top of their respective satellite picture by drawing shapes that matched the features of each hole, such as fairways, greens and lakes.



Figure 6.1: Screenshot of the satellite image of Hole 2



Figure 6.2: Screenshot showing the SVG recreation of Hole 2

Once each hole had been successfully recreated using shapes in an SVG file, I could then begin to draw the hole specific zones. In order to ensure maximum accuracy, I opted to hand draw each zone onto every hole. This was a time-consuming approach; however I felt it was necessary to provide the most accurate data collection as possible. This is because different holes have different features, and a blanket approach for zone drawing would not be suitable for the purposes of data collection and analysis.

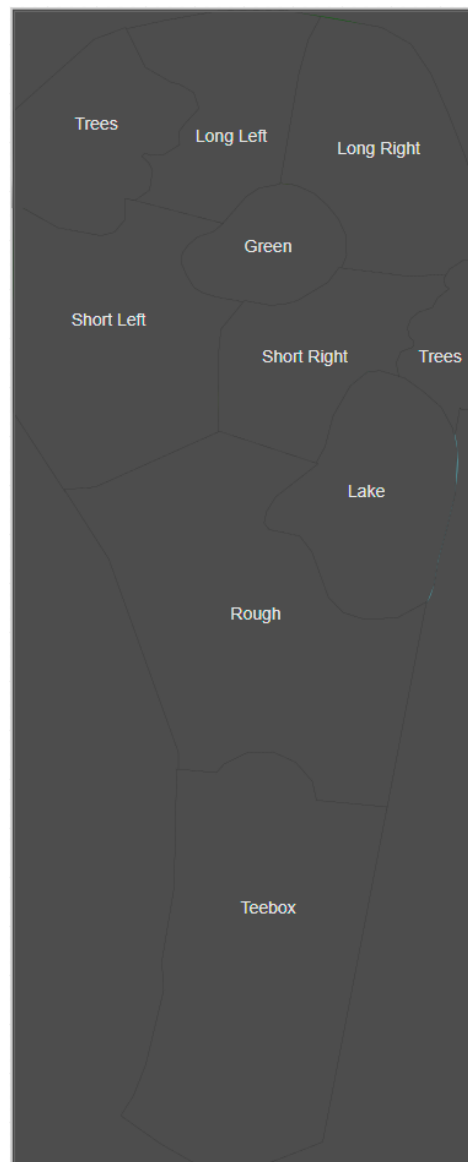


Figure 6.3: Screenshot showing the SVG image for Hole 2, with the hole specific zones drawn on top. This image has text containing the correlated zone for explanatory purpose

6.1.2 User Interface Implementation

The user interface was implemented in line with the wireframe designs documented in the design section. Figure 6.4 showcases the homepage of The Golf Blueprint, which provides users with a clear and simple introduction to the system. The minimalistic and clean design follows the colour palette outlined in the design section, creating a professional appearance that is utilised throughout the entire system. As a part of the wider user interface of the system, an intuitive navigation bar runs across the top of the page, allowing for the user to quickly navigate the application. This is a feature across all pages of the system.

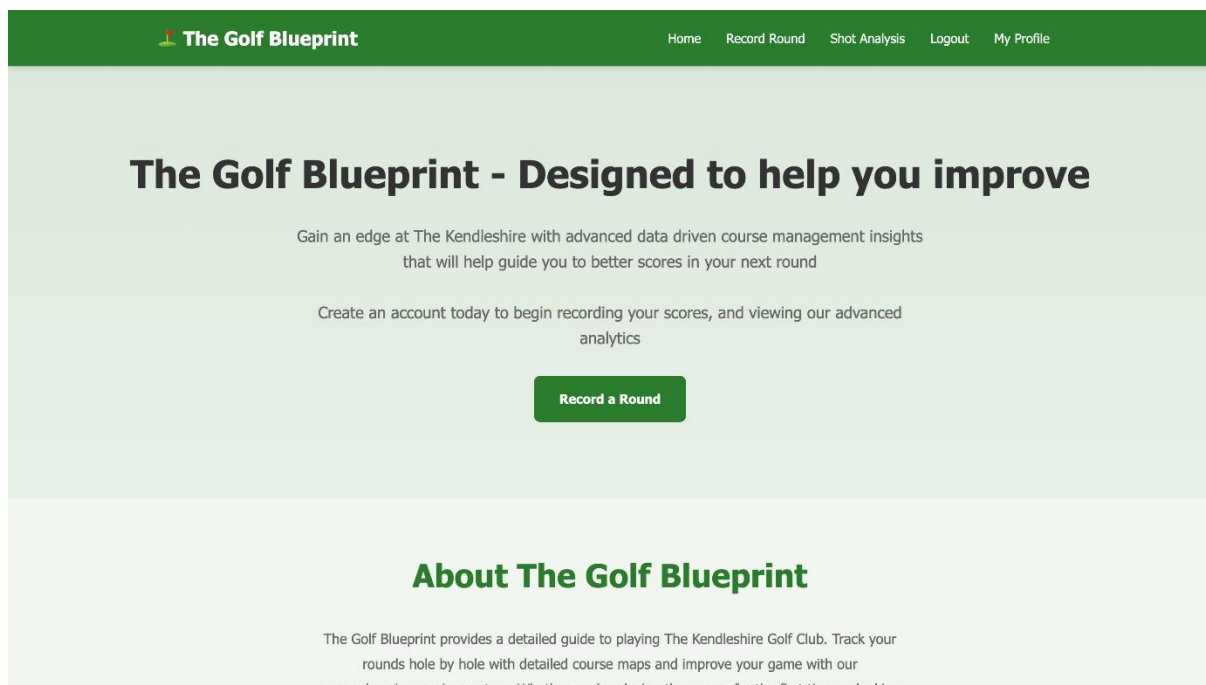


Figure 6.4: Homepage implementation showing the welcome message, call-to-action button for recording a round and navigation bar

6.1.3 User Login Page

The user login page was an extremely important feature of the system, for allowing users to keep track of their individual progress. A secure way of logging in and out of The Golf Blueprint was implemented using a simple page, which can be seen in Figure 6.5.

The screenshot shows the login page for 'The Golf Blueprint'. At the top, there is a green header bar with the site's logo and name on the left, and 'Home' and 'Login' links on the right. The main content area has a light green background. In the center, there is a white card with a green border. The card has a title 'Login' in green, underlined. Below the title, there are two input fields: 'Email' and 'Password'. The 'Email' field is a blue-outlined text box. The 'Password' field is a blue-outlined text box with dots for masking. Below these fields is a green button with the text 'Login'. At the bottom of the card, there is a link that says 'Don't have an account? Register here'.

Figure 6.5: Screenshot showcasing the login page for The Golf Blueprint

As well as implementing this simple login page, it was just as important to incorporate a registration page to allow new users to sign up. It was important to incorporate password masking into this design, so that when the password boxes are typed into, no text is visible.

The screenshot shows the registration page for 'The Golf Blueprint'. At the top, there is a green header bar with the site's logo and name on the left, and 'Home' and 'Login' links on the right. The main content area has a light green background. In the center, there is a white card with a green border. The card has a title 'Register' in green, underlined. Below the title, there are four input fields: 'Username', 'Email', 'Password', and 'Confirm Password'. The 'Username' field is a white-outlined text box. The 'Email' field is a white-outlined text box. The 'Password' field is a white-outlined text box with dots for masking. The 'Confirm Password' field is a white-outlined text box. Below these fields is a green button with the text 'Register'. At the bottom of the card, there is a link that says 'Already have an account? Login here'.

Figure 6.6: Screenshot showcasing the registration page of The Golf Blueprint

In order to comply with the requirements for the design of the system, it was also necessary to implement error handling, to enforce that users of the system must be logged in to use features such as recording a round. I wanted to ensure that the error message fitted in with the rest of the system, but still was bold enough to alert the user that there has been an error. Another important feature of the error message design was to give the user the option to easily reach the login page from the error message, by putting a 'Go to Login' button on the error message box

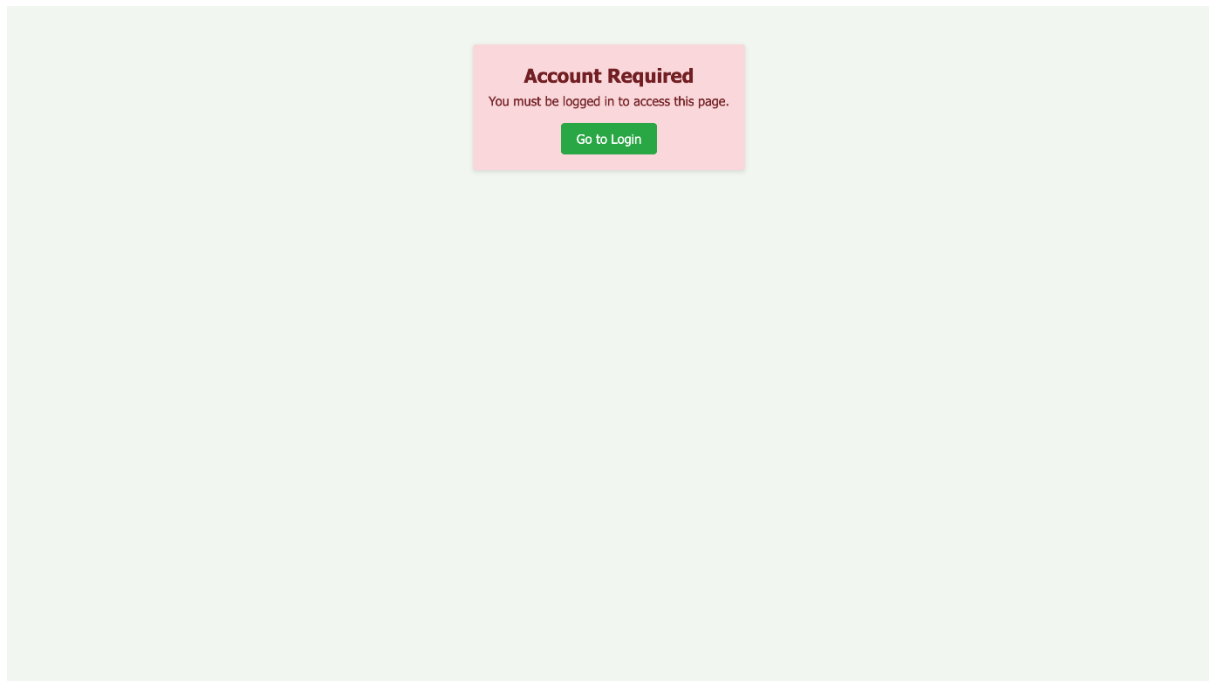


Figure 6.7: Screenshot showcasing the error generated by a user attempting to record a round, without having logged in

6.1.4 User Profile Pages

A key feature of the application was implementing the user profile pages, that allow users to view their created account, and all of the associated account information. This includes their detailed account details, their statistics of number of rounds played, and their best round and their round history, in order to provide the users with a way of tracking their golfing progress, as set out in the requirements.

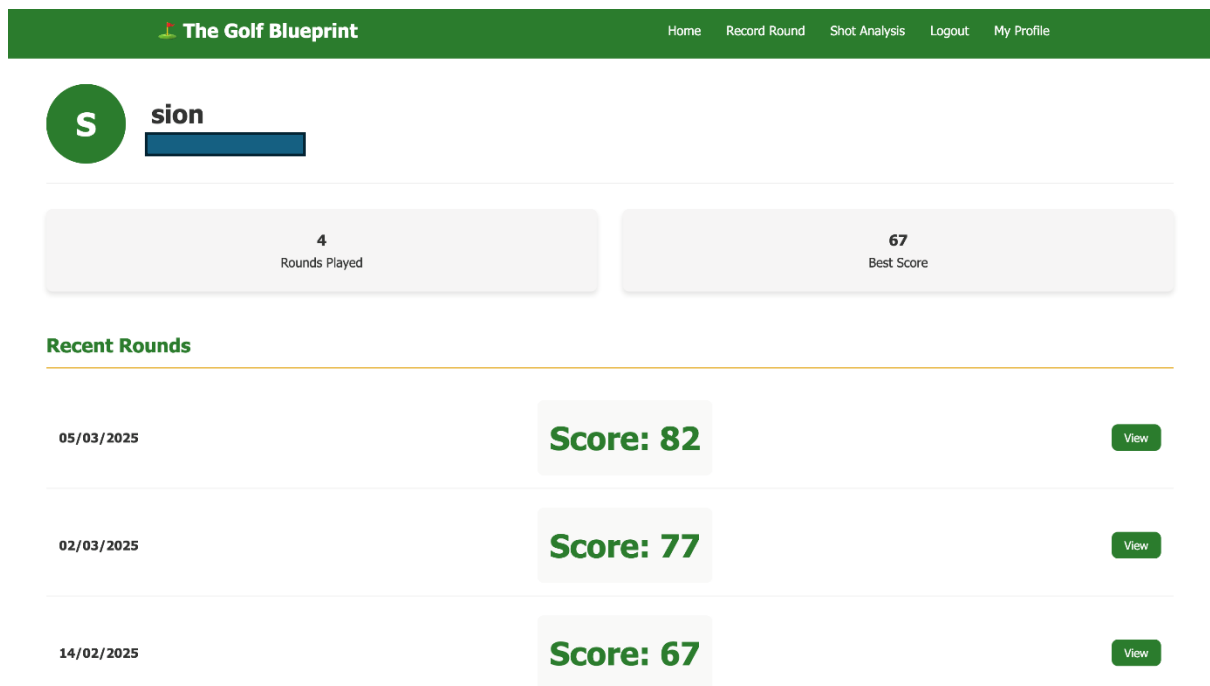


Figure 6.8: Screenshot showcasing the implementation of the user profile page

This user profile page would allow the users to see their overall round history, but in order for the users to better track their progress and round history, the ability to view individual rounds of golf that have been inputted into the database was also successfully implemented. This simple and familiar layout of a golf scorecard will allow the users to revisit their previous rounds in greater detail and shows them statistics such as their number of pars, birdies, bogeys and double+ bogeys.

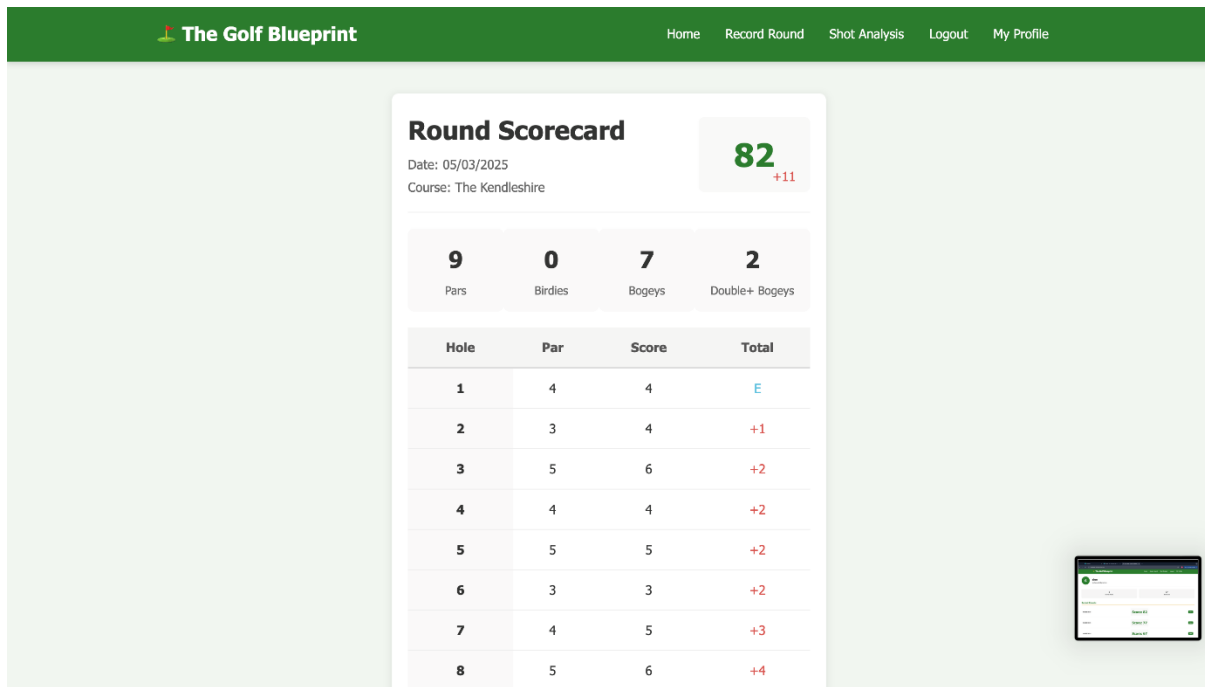


Figure 6.9: Screenshot showcasing the individual round scorecard page of the system

6.1.5 Shot Data Collection System

For a user to enter their shot data into the system, they first must select the date that the round was played on. This will then be attached to the round when it is stored in the database.

The shot tracking system was implemented using interactive SVG graphics/images that allow users to accurately mark shot locations from their rounds. Since the SVG image has been split up into detailed zones, all the user has to do is click on the map where their shot landed, and the details of the shot will be stored into the database. There are intuitive instructions on the page which walks the user through the steps for recording a round accurately. When a user hovers over the map, the different zones for that specific hole will be highlighted, giving the user a greater insight into how the data will be stored for that shot.

This page also allows the user to enter further details about that specific hole, such as the number of putts, and adding any penalty strokes if required.

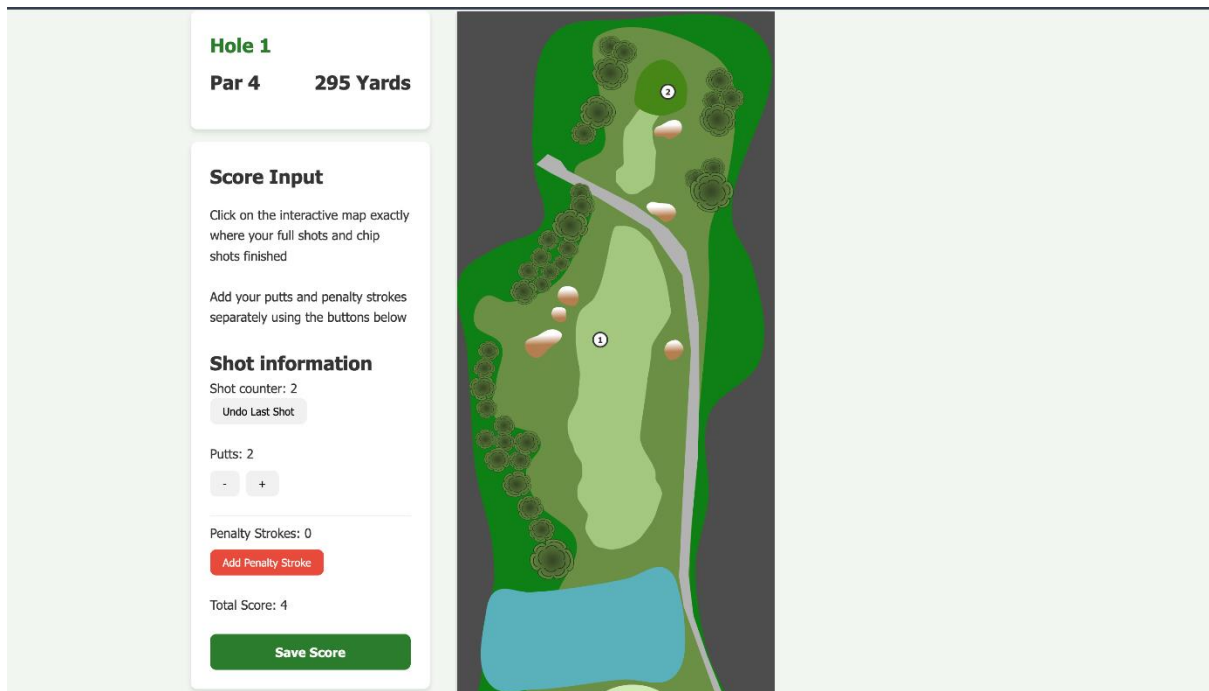


Figure 6.10: Screenshot showcasing the shot placement screen

Once the user has clicked on the map with the necessary shot locations and has entered in the correct amount of putts or penalty strokes, the user can see where they have clicked and in what order by looking at the white circles containing numbers on the image. Once the user is satisfied that the data is correct, they can click the save score button, which will provide them with a confirmation message that their score was saved, and after a short delay it will transition them to the data entry for the next hole.

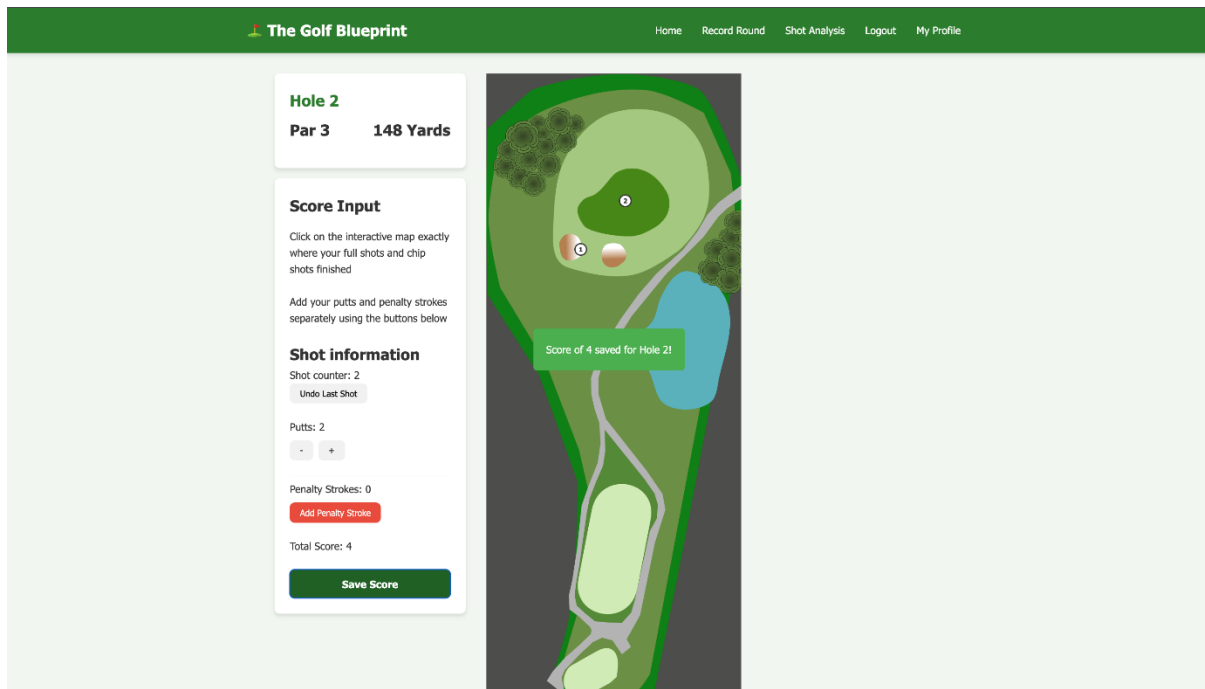


Figure 6.11: Screenshot showcasing the confirmation of score entry once a user has clicked on 'Save Score'

6.1.6 Data Heatmap Visualisation and Analytics

The heatmap visualisation system has been correctly implemented successfully. The system is able to analyse data from the SQL database and create hole specific heatmaps which provide detailed insight into course management considerations, which satisfies the requirements for the system.

The implementation uses colour coding to represent the average score per zone, with green for lower scores, yellow for neutral scores, and red for areas associated with higher scoring. This intuitive design allows users to quickly gain an understanding of which areas of each specific hole they should either actively to utilise or avoid entirely in order to help their golf scoring.

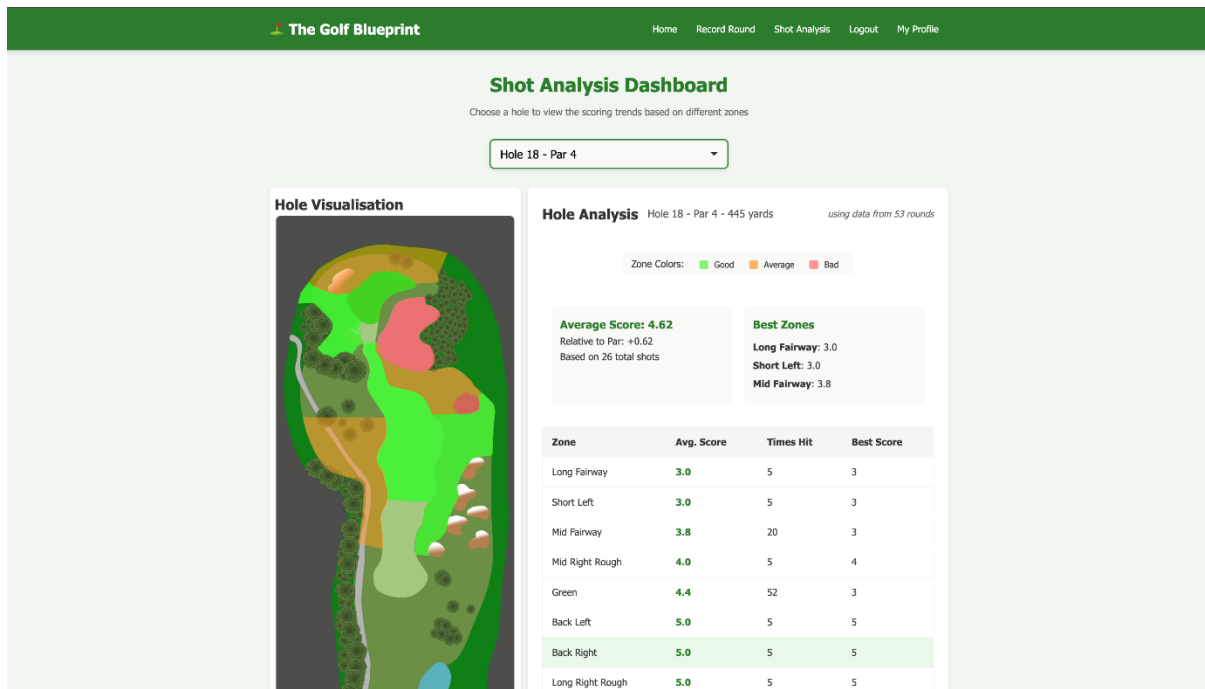


Figure 6.12: Screenshot showcasing the hole/shot visualisation page for the 18th hole

6.2 System Testing

6.2.1 Test Approach

Testing for the system followed a comprehensive approach to ensure that the application has met all of the specified requirements and delivered a high-quality user experience.

A total of 31 test cases were developed across 4 components: User Authentication (9 tests), Profile Access (5 tests), Hole/Shot Analysis (4 tests) and Round Recording (13 tests). These represent the key functionality of the system and ensure alignment with the objectives of the system/project.

6.2.2 User Authentication

The User Authentication was tested covering the essential functionality of user registration, login, logout and validation:

Test ID	Component	Test Description	Test Steps	Expected Result	Actual Result	PASS/FAIL
1	User Authentication	Test User Setup	1. Click login 2. Click 'Register' 3. Enter user details 4. Click 'Register' 5. Verify user creation in database	User details are created and stored in SQL database	As Expected	PASS
2	User Authentication	Test User Login	1. Click 'Login' 2. Enter user details 3. Click 'Login' 4. Verify login in console	User is logged in - confirmed by console	As Expected	PASS
3	User Authentication	Test User Logout	1. Ensure user is logged in 2. Click 'Logout' 3. Check user is logged out with the system message	System provides message that user has logged out. 'Logout' changes to 'Log in'	As Expected	PASS
4	User Authentication	Test Invalid Credentials	1. Click 'Login' 2. Enter invalid credentials 3. Click 'Login' 4. Check error message is shown	System provided error message that email or password is invalid	As Expected	PASS
5	User Authentication	Test Invalid Email	1. Click 'Register' 2. Enter an email that doesn't include an '@' 3. Click 'Register'	System provides error message for the email box requiring an '@'	As Expected	PASS
6	User Authentication	Test Single Use Email Addresses	1. Click 'Register' 2. Enter an email address that is already in use 3. Click 'Register'	System provides an error stating the email is already in use	As Expected	PASS
7	User Authentication	Test Password Matching	1. Click 'Register' 2. Enter valid email address 3. Enter password 4. Enter different password in 'Confirm Password' box 5. Click 'Register'	System provides an error saying that the passwords do not match	As Expected	PASS
8	User Authentication	Test Login Requirement to Record Round	1. Ensure user is logged out 2. Click 'Record a Round'	System provides an error saying that the user must be logged in to record a round	As Expected	PASS
9	User Authentication	Test Logging Out Takes User to Homepage	1. Click 'Logout' 2. Ensure user is taken to homepage	When clicking logout, no matter what page the user is on, they are taken to the homepage	As Expected	PASS

Figure 6.13: Table showcasing the user authentication test cases

These tests verified that the authentication system provided secure access to the application while offering appropriate user feedback for invalid operations. The test results confirm that users can successfully register, login, and logout, and that the system appropriately handles error cases.

6.2.3 Profile Access

The Profile Access component was tested with 5 test cases to ensure user profiles displayed accurate information and provided access to historical data:

Test ID	Component	Test Description	Test Steps	Expected Result	Actual Result	PASS/FAIL
1	Profile Access	Test Loading User Profile	1. Ensure Status is Logged In 2. Click 'Profile'	The system should load the correct profile for the logged in user	As Expected	PASS
2	Profile Access	Test Data Accuracy	1. Click 'Profile' 2. Ensure Username and Email Address are Correct	The system should automatically load the signed in user's username and email address	As Expected	PASS
3	Profile Access	Test Statistic Accuracy	1. Click 'Profile' 2. Ensure Rounds Played and Best Score Statistics are Correct	The system should automatically load the rounds played and best score statistics, accurately	As Expected	PASS
4	Profile Access	Test Viewing a Historic Round	1. Click 'Profile' 2. Click 'View' on an Old Round	When clicking view, the system should load up the scorecard for the historic round	As Expected	PASS
5	Profile Access	Test Historic Round Details are Correct	1. Click 'View' on an Old Round	When viewing an old round, the data that is displayed, such as round location, round date and scoring details should all match that of the original round	As Expected	PASS

Figure 6.14: Table showcasing the profile access test cases

These tests verified that the user profile functions correctly displayed personalised information and allowed users to access their historical round data. The test results confirm that users can track their progress over time, addressing the progress tracking objective stated in the requirements.

6.2.4 Hole/Shot Analysis

The Hole/Shot Analysis component was tested with 4 test cases to ensure the heatmap visualisation correctly displayed strategic insights:

Test ID	Component	Test Description	Test Steps	Expected Result	Actual Result	PASS/FAIL
1	Hole/Shot Analysis	Test Loading Shot Analysis Page	1. Click 'Shot Analysis'	The shot analysis page will load, and display the data for the first hole by default	As Expected	PASS
2	Hole/Shot Analysis	Test Changing Hole	1. Click the Hole Drop Down Menu 2. Select Hole to View	The system will change the hole view to whatever hole is clicked by the user	As Expected	PASS
3	Hole/Shot Analysis	Test Heatmaps Working	1. Click on a Hole to View the Heatmap 2. Check the Heatmap Colour Matches the Average Scores for That Zone	When viewing a hole's zone heatmaps, the higher score zones should be red, and lower score zones should be green	As Expected	PASS
4	Hole/Shot Analysis	Test Hovering Over Heatmap	1. Click on a Hole to View the Heatmap 2. Hover Over a Zone	When hovering over a zone, it should show the user the name of the zone, and the average score	As Expected	PASS

Figure 6.15: Table showcasing the hole/shot analysis test cases

These tests verified that the shot analysis functionality correctly displayed heatmap visualisation. The successful implementation of the color-coded zones (red for higher scores, green for lower scores) directly addresses the primary aim of improving course management through visual insights.

6.2.5 Round Recording

The Round Recording component was tested with 13 test cases to ensure accurate and reliable shot data collection:

Test ID	Component	Test Description	Test Steps	Expected Result	Actual Result	PASS/FAIL
1	Round Recording	Test Entering Round Details	1. Click 'Record a Round' 2. Enter Date for the round 3. Click 'Begin Scoring'	The chosen date will be reflected in the database	As Expected	PASS
2	Round Recording	Test Not Putting a Date	1. Click 'Record a Round' 2. Do not Enter a Date 3. Click 'Begin Scoring'	The system will prompt the user to enter a date	As Expected	PASS
3	Round Recording	Test Hole 1 Loading	1. Click 'Begin Scoring'	The first hole's page should load up automatically	As Expected	PASS
4	Round Recording	Test Placing Shots	1. Click on SVG Image of Hole Where Shots Landed	The SVG Image should showcase a white circle where the shot was placed. Total score should	As Expected	PASS
5	Round Recording	Test Adding Putts	1. Click the Plus Button Under Putts	The number of putts and the total score should increase by 1	As Expected	PASS
6	Round Recording	Test Adding Penalty Stroke	1. Click 'Add Penalty Stroke'	The system should give a message saying 'Penalty Stroke Added'. Total score should increase by 1	As Expected	PASS
7	Round Recording	Test Removing a Shot	1. Click 'Undo Last Shot'	The last shot should be undone. Whether it is a putt or a full shot, or penalty shot	As Expected	PASS
8	Round Recording	Test Removing a Putt	1. Click the Minus Button Under Putts	The number of putts and the total score should decrease by 1	As Expected	PASS
9	Round Recording	Test Saving Score	1. Enter Scoring for Hole 2. Click 'Save Score'	System should show a message saying that the score was saved. The next hole should load up after a short delay	As Expected	PASS
10	Round Recording	Test Unrealistic Score	1. Try to Submit a Score of 0	System should show a message saying that a score of 0 is not possible	As Expected	PASS
11	Round Recording	Test Round Summary	1. Submit Score for 18th hole	The system should automatically load up the round summary	As Expected	PASS
12	Round Recording	Test Editing a Hole	1. Click 'Edit'	When clicking edit, the system should take the user back to the score input for that hole	As Expected	PASS
13	Round Recording	Test Round Submission	1. Enter Full Round 2. Submit Round	When clicking submit, the system should show a message confirming the submission. The round should appear in the database	As Expected	PASS

Figure 6.16: Table showcasing the round recording test cases

These tests verified that users could accurately record their round data, including shot locations, putts, and penalty strokes. The successful implementation of this component ensures that the system collects reliable data to support the heatmap visualisations and other analytical features.

6.3 Requirements Traceability Matrix

The Requirements Traceability Matrix (RTM) provides a structured and detailed example of the mapping between the functional requirements and their implementation and subsequent testing through the project. It demonstrates how/if each functional requirement has been addressed, the relevant software file, the testing ID to prove it has been addressed and whether it was a pass or a fail.

Requirement ID	Priority	Software Module	Test Case ID	Status
FR1	Must	register.html	Authentication 1	Pass
FR2	Must	login.html	Authentication 2	Pass
FR3	Must	golf_blueprint.sql	Authentication 1	Pass
FR4	Should	-	Unimplemented	Fail
FR5	Should	-	Unimplemented	Fail
FR6	Could	-	Unimplemented	Fail

Figure 6.17: Table showcasing the RTM for the functional requirements 1-6

Requirement ID	Priority	Software Module	Test Case ID	Status
FR7	Must	hole1.html	Round Recording 3	Pass
FR8	Must	hole1.html	Round Recording 4	Pass
FR9	Must	round-details.html	Round Recording 1	Pass
FR10	Must	hole1.html	Round Recording 5	Pass
FR11	Must	round-summary.html	Round Recording 13	Pass
FR12	Must	hole1.html	Round Recording 10	Pass
FR13	Should	-	Unimplemented	Fail
FR14	Should	round-summary.html	Round Recording 12	Pass
FR15	Could	hole1.html	Round Recording 13	Pass

Figure 6.17: Table showcasing the RTM for the functional requirements 7-15

Requirement ID	Priority	Software Module	Test Case ID	Status
FR16	Must	shot-analysis.html	Hole/Shot 3	Pass
FR17	Must	shot-analysis.html	Hole/Shot 3	Pass
FR18	Must	shot-analysis.html	Hole/Shot 3	Pass
FR19	Should	shot-analysis.html	Hole/Shot 4	Pass
FR20	Could	-	Unimplemented	Fail

Figure 6.17: Table showcasing the RTM for the functional requirements 16-20

Requirement ID	Priority	Software Module	Test Case ID	Status
FR21	Must	profile.html	Profile Access 3	Pass
FR22	Must	scorecard.html	Profile Access 4	Pass
FR23	Must	scorecard.html	Profile Access 5	Pass
FR24	Should	-	Unimplemented	Fail
FR25	Could	-	Unimplemented	Fail

Figure 6.17: Table showcasing the RTM for the functional requirements 21-25

Requirement ID	Priority	Software Module	Test Case ID	Status
FR26	Must	golf_blueprint.sql	Authentication 1	Pass
FR27	Must	1.svg	Hole/Shot 1	Pass
FR28	Could	-	Unimplemented	Pass

Figure 6.17: Table showcasing the RTM for the functional requirements 26-28

The Requirements Traceability Matrix demonstrates that The Golf Blueprint has successfully met its primary objective of providing a data-driven golf course management tool through the complete implementation of all "Must Have" requirements. The strategic prioritisation of development efforts ensured that the core functionality necessary for collecting shot data, visualising strategic insights through heatmaps, and tracking player progress was delivered successfully.

While some secondary and enhancement features remain for future implementation, these do not impact the system's ability to deliver on its fundamental value proposition. The 74.1%

overall implementation rate represents an appropriate balance between feature completeness and development timeframe constraints

7. Conclusion

(TO BE INSERTED LATER)

The Golf Blueprint differentiates itself by focusing specifically on course management at The Kendleshire Golf Club, utilising crowd-sourced data to create a comprehensive understanding of each hole's strategic challenges. Unlike generic golf GPS apps that simply provide distances, or personal tracking systems that focus on individual performance metrics, this project aims to capture and visualise the collective experience of players at The Kendleshire.

This local focus allows for the identification of specific course management patterns that generic golf analytics solutions cannot provide.

References

- Ahmad, M.O., Markkula, J. and Oivo, M. (2013) *Kanban in software development: A systematic literature review*. In: *2013 39th Euromicro Conference on Software Engineering and Advanced Applications* [online] 2013 39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA). Santander: IEEE, pp. 9–16. Available from: <https://ieeexplore.ieee.org/document/6619482/> [Accessed 27 January 2025].
- Betzler, N.F., Monk, S.A., Wallace, E.S. and Otto, S.R. (2012) Effects of golf shaft stiffness on strain, clubhead presentation and wrist kinematics. *Sports Biomechanics* [online]. 11 (2), pp. 223–238.
- Broadie, M. (2014) *Every Shot Counts: Using the Revolutionary Strokes Gained Approach to Improve Your Golf Performance and Strategy*. Penguin.
- Brush, K. (2023) *What Is the MoSCoW Method? TechTarget*. March 2023 [online]. Available from: <https://www.techtarget.com/searchsoftwarequality/definition/MoSCoW-method>.
- Ehrlich, J.A. and Kamimoto, N. (2024) *Analyzing professional golf performance and earnings: insights from strokes gained analysis*. *Managerial Finance* [online].
- ISO (2024) *ISO/IEC 38500:2024_02 : Information technology - governance of IT for the organization*.
- Kute, S.S. and Thorat, S.D. (2014) A Review on Various Software Development Life Cycle (SDLC) Models. 3 (7).
- Lei, H., Ganjezadeh, F., Jayachandran, P.K. and Ozcan, P. (2017) A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing* [online]. 43, pp. 59–67.
- MacKenzie, D. (2023) *Course Management Lessons from The PGA Tour Instruction For The Mental Game of Golf*. 30 August 2023 [online]. Available from: <https://golfstateofmind.com/course-management-lessons-from-the-pga-tour/>.
- Plummer, B. (2024) *What Is Strokes Gained? Golf Monthly Magazine*. 19 March 2024 [online]. Available from: <https://www.golfmonthly.com/features/the-game/what-is-strokes-gained-185460>.
- Pressman, R.S. (2001) ‘Software engineering: a practitioner’s approach’ McGraw-Hill series in computer science software engineering and databases. 5. ed., 20. anniversary ed. Boston, Mass: McGraw Hill.

Sjøberg, D.I.K. (2018) *An empirical study of WIP in kanban teams*. In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* [online]ESEM '18: ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. Oulu Finland: ACM, pp. 1–8. Available from: <https://dl.acm.org/doi/10.1145/3239235.3239238> [Accessed 27 January 2025].

The R&A (2024) *Golf participation continues to enjoy growth globally R&A*. 25 September 2024 [online]. Available from: <https://www.randa.org/en/articles/golf-participation-continues-to-enjoy-growth-globally> [Accessed 27 January 2025].