TrackMate Report

Unit Code: CM10025
Group Name: TrackMate
Group Number: 7

Table of Authors

Group Member	Username	Degree Course
Nicholas Legh-Smith	nls40	BSc Computer Science and Mathematics (yr. 1)
Amarveer Johal	asj71	MComp Computer Science and Mathematics (yr. 1)
Taha Ahmed	tma42	BSc Computer Science and Mathematics (yr. 1)
Rishabh Madhok	rm2870	BSc Computer Science (yr. 1)
Samsul Hoque	sh3295	BSc Computer Science and Mathematics (yr. 1)
Oscar Horsfall	owh31	BSc Computer Science (yr. 1)
Tanay Patel	thp39	BSc Computer Science (yr. 1)
Luka Matthew Ian Cooke	lc2880	BSc Computer Science (yr. 1)
Aaryan Lal	al2698	MComp Computer Science (yr. 1)
Adil Rasheed	ar3128	BSc Computer Science and AI (year 1)

TrackMate Report

Nicholas Legh-Smith, Amarveer Johal, Taha Ahmed, Rishabh Madhok, Samsul Hoque, Oscar Horsfall, Tanay Patel, Luka Matthew Ian Cooke, Aaryan Lal, Adil Rasheed

Abstract

TrackMate is a gym tracking app designed to help users, particularly students, track their workouts and monitor progress over time. The main problem it addresses is the difficulty in consistently tracking gym performance and staying motivated. The app allows users to log detailed workout data such as sets, reps, and weight lifted while offering visual progress tracking through graphs and infographics. It also enables users to set and track fitness goals, create personalised workouts, and monitor their fitness journey through a simple, user-friendly interface. The app's key features include workout logging, goal setting, progress visualisation, and motivation through streaks and milestones. The two sprints focused on implementing core features such as a profile tab, goal system, basic workout logging, and simple graphs. Sprint 1 refined the UI, improved data visualisation, and enhanced the goal system. Sprint 2 introduced features like exercise customisation and advanced goal tracking, with an emphasis on user engagement. For further evolution, the app could add multi-device synchronisation, integration with wearables, social features, and gamification. It could also benefit from having a login system so that multiple users are capable of using the app.

Table of Contents

Introduction

Account of the problem and problem area (Personal Informatics)

Our product TrackMate is a comprehensive solution to a growing problem in an even faster growing industry that is Personal Informatics. The aim of our project is to produce an app that caters to all information needs of gym-goers that care about tracking and monitoring their performance and activity in the gym, in order to make progress and meet their fitness goals. Fitness and more specifically gym activity has shown to correlate to significant boosts in mental wellbeing, confidence, feeling of inclusion and academic performance (ukActive Research Institute, BUCS, Sport England, 2019). 70% of students achieving first classes attend the gym and moreover, 90% of students attend gym according to the article. Therefore this highlights the benefit of fitness as well as the mass market available for a product like ours. Therefore, the aim of our project is to cater to such users and provide them with a tool that can aid them to achieve their fitness goals and stay consistent to reap the host of benefits stated.

The personal informatics sector is a rapidly growing industry. With more and more people getting involved in fitness, users are keen to find new ways to monitor and track their progress to help reach their individual goals. To understand the importance of personal informatics on fitness we researched an article (Li, Medynskiy, Froehlich, Larsen, 2012). This article provided a holistic guide into personal informatics and a roadmap to building an effective one. From the article we understood that PI systems support behavioral change and self control - issues that we are keen to solve, interestingly it explained that implementing behavioral theories into the design that promote long term engagement will lead to the greatest likelihood of behavioral change.We then looked to research the effects that infographics are having from a statistical perspective (Kersten-Van Dijk, Westerink, Beute, IJsselsteijn, 2017). 88% of participants gained some form of self insight, 38% reported behavioral change. The guidance from the study suggested that to increase these numbers, all stages of behavioral change should be supported not just maintenance, in addition the app can provide contextual information and goal based feedback as opposed to raw data. These data points gave us plenty of valuable insights into the efficacy of infographics and how to optimise their effect.We then researched the efficacy of current PI softwares in a study (Rapp, Cena, 2016) it showed that for users with no PI experience, tracking became burdensome and user activity dropped later in the trial, additionally the apps available offered dull infographics and little insights into the raw data. Therefore, this research gave us key information as to the problem area and how our product can fulfill the problem (producing valuable infographics) better than the competition.

What makes a PI software effective for users?

To identify the key features that make PI softwares effective we researched 3 different leading apps in the market: Strong, Heavy, Gymshark-Workout.

Firstly, we researched Strong. The key features that make Strong appealing to users are: the profile, workout log, macros log(diet), custom exercises and workouts tab (Dr Muscle, 2025). It

uses a minimalistic design with an intuitive UI making it easier to get started and make use of the full feature package.

Secondly, we researched Hevy. The key features that make Hevy appealing to users are: workout planner, an exercise timer, graph history, weight tracking (Garett Reid, 2025). The USP of the app is advanced tracking data dating back into history as well as a number of engaging infographics to capture your progress.

Thirdly, Gymshark Workout. The key features that make Gymshark Workout appealing to users are: single workouts, short term programs, speciality programs - designed by specialist trainers (Garett Reid, 2025). Specialized programs provide personalization to fulfill users' individual needs thereby branching from a generic to specialized service platform.

The key ideas behind our PI software

For our PI software we have decided to implement a profile tab, a goal tab, a log workout tab a database and a graphic UI. TrackMate will enable users to set up their own accounts and log information about their individual workouts as well as setting and editing their goals and profile details e.g. weight. The UI will provide engaging infographics to display the users progress tracked over time. All workouts will be logged and stored in the database which can be recalled for future reference. We aimed to implement a more minimalist front end design that is both intuitive and easy to use (thereby reducing initial usage hurdles) and also simple to implement under the time constraints we have for the task (2 sprints). We aimed to use the database to collate a host of bioinformatic evidence over a span of time (storing historical data) to then provide valuable insights and infographics to display to the user in the various tabs on the app, in order to improve future performance and aid the user to meet fitness goals..

Agile Software Process Plan and Management

Planning
For this project, we followed the SCRUM framework for agile software development. We began with a few meetings to conceptualise the project as a group, between which we conducted market research, interviewed potential users and used surveys to inform our choice of project and requirements. We decided on a gym tracking app with the following product backlog.

Trackmate Product Backlog:

- Multiple input streams (Such as time spent at gym, sleep data, calories. Could potentially use fitbit API)
- Allow long term tracking of data
- Allow the user to view their progress in graphs
- Have some sort of streak based motivation system
- Allow users to set goals
- Allow users to edit goals
- Allow for comparing different kinds of PI data together

- Profile to log weight and gender etc
- Clean UI

Our plan was to run two sprints, lasting around two weeks each. We began each sprint with a scrum meeting where we would allocate tasks between members of the group based on the requirements agreed upon in previous meetings. We kept tasks small in order to maintain simplicity, and we estimated the size of each task as small (S), medium (M) or large (L) to ensure a fair distribution of the workload. At the end of the sprint we would review the current functionality of the app, modifying our requirements if necessary which allowed us to embrace change, and discuss how it could be improved in the next sprint. In this scrum meeting we would ensure any defects were triaged so they could be prioritised in the next sprint.
Sometimes, we would meet in the middle of a sprint too in order to help each other or discuss how best to implement features, i.e. what software packages we should use or how we want the user interface to look. These mid-sprint scrum meetings were particularly useful for communication among the team and allowed more collaboration, in accordance with the “people not process” principle of agile development.
We initially planned to focus on the backend, such as data collection and storage, in the first sprint. Then we would work on analysis and visualisation of the data in our second sprint. However, after deciding to use PySide6 with Qt for the user interface, we found that Qt encourages a form of asynchronous programming where the structure of the program depends heavily on the user interface (with functionality linked to widgets such as buttons and menus). Because of this we decided that in the first sprint we would concentrate mostly on creating a plain but functional user interface, and then improve the aesthetics as well as implementing more of the backend in the second sprint.

Sprint 1
Backlog:
- Allow long term tracking of data
- Multiple input streams
- Allow users to set goals
- Clean UI

Tasks:
- Create a database that can store bodyweight and also exercises completed (including the weight) over time. - L
- Make the homepage more aesthetic (we designed the homepage together in the meeting, leaving the different tabs that it links to to be developed in the sprint). - S
- Create the “Goals” tab where the user can set themselves goals. - M
- Create the “Profile” tab displaying age, name, gender and bodyweight, as well as allowing the user to input these fields. - M
- Create the “LogWorkout” tab where the user can enter the exercises and weight they’ve been doing. - M

Review:

Everyone completed their tasks successfully. However, the UI was unattractive within the tabs. We agreed to prioritise fixing this for the next sprint.

Sprint 2
Backlog:

- Allow the user to view their progress in graphs
- Allow users to edit goals
- Allow for comparing different kinds of PI data together
- Clean UI

Tasks:

- Improve UI aesthetics - M
- Add a feature for users to remove goals once they've been met - S
- Create the "Progress" tab where data is displayed in a graph, and the user can select what exercises (or their bodyweight) they want displayed in the graph in order to compare their progress - L
- Additional more rigorous testing of code across the whole project - L

Review:
Overall success. Current version of the app is functional and looks nice. Things we'd consider improving if we were going to do another sprint would include using fitbit API, automating the management of goals and adding a streak based motivation system.

Software Requirement Specification

Introduction & Requirements Elicitation

When we first started thinking about what this gym app would do, we began by looking at existing apps like Hevy, Gymshark, and Strong to understand user needs and expectations.

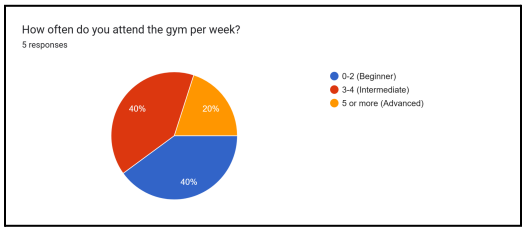
We found many online reviews with users making points like:

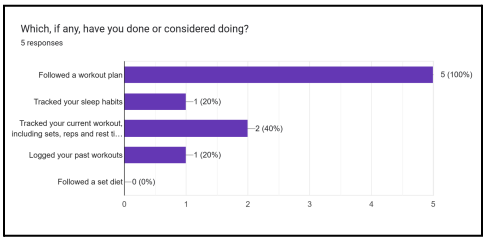
- They hated when simple features were locked behind paywalls
- They wanted cleaner layouts with less distractions
- They liked seeing their progress clearly (e.g graphs)
- They wanted custom workouts, not just preloaded ones

What People Told Us

In total, we interviewed five students, conducted a survey (from beginner to athlete level; shown below) and conducted market competitor reports [see: insert references]. Main concerns were:

- Progress tracking is key: Users wanted to see how they're doing week by week - alluding to numbers or visual graphs (Interview 1, 3)
- Cluttered apps suck: Complicated layouts or apps packed with ads made people give up. Simplicity was key. (Interview 4)
- Custom workouts matter: Users hated being stuck with preloaded exercises - Customisation was desired. (Interview 1, 4)
- No weight-loss obsession: Users were put off by apps that only focused on losing weight. They wanted a focus on feeling better or building strength (Interview 5)





How This Shaped Our Features

Based on what people said, we made a complete feature list (which is expanded on in 4.1):

- Users can set goals and track progress toward them
- The app gives motivation, like streak counters and weekly targets
- Everything should be easy to access from the main screen — no digging through menus
- Users can add custom workouts, including drop sets and special exercises
- No paywalls. Basic features will all be free.

We dropped some ideas, like a sleep tracker, because most people felt it was unnecessary. All these features went into our Sprint 1 backlog, and we planned to update them as we go through sprints. We're using an Agile Scrum approach, so this whole requirements list could (and did) change based on what worked and what didn't.

Domain and User Viewpoint

System Domain: Personal Informatics for Fitness

Our app is being built for the Personal Informatics (PI) domain.
"Personal Informatics basically means apps that help people collect, track, and reflect on personal data to improve their lives" (CM12005 Agile slides, [10]).

In our case, the data being tracked is information like:

- Gym workouts (exercises, sets, reps, weight lifted)
- Progress over time (graphs and stats)

- Fitness goals (e.g., workout streaks, lifting targets)
- Personal Information (Weight, Age)

The goal of a PI system is to help users reflect and see real improvement over time (Li et al., 2010 [12]). Our app shows their progress through simple visuals like graphs and badges, motivating users without constant, annoying reminders.

Target User Viewpoints

Each data stream offers motivational insights, with gym tracking being key. Logging workouts feels satisfying, reviewing progress is rewarding, and tracking sets reveals what works best. Monitoring personal data shows weight changes and goal progress, making long-term achievements especially fulfilling.

Overall, this app supports gym-goers who want to measure their progress and customize how they view and track their fitness journey.

Use Cases

Standard Use Case - Casual Gym-Goer

Actor: A student who aims to go to the gym once or twice a week, sometimes missing weeks.

Goal: Stay motivated to keep going to the gym while tracking their progress.

System Use:

- Will log their workout after each session
- Updates their personal information such as weight regularly
- Will setup fitness related goals for additional motivation
- Will occasionally look at their long term graphs to be inspired by their overall progress

Purpose:

The main purpose for this actor is to help inspire them to continue going to the gym. While they do not need precise information or logging, they benefit greatly from being encouraged by celebrating various milestones and graphs, increasing their likelihood of returning to the gym.

Secondary Use Case - Gym Enthusiast

Actor: A student who goes to the gym multiple times a week and monitors performance closely.

Goal: Have detailed workout tracking to optimise training.

System Use:

- Log customised workouts with specific exercises, sets and reps

- Use the progress charts to review muscle group coverage and optimise future workouts
- Set specific goals such as lift targets

Purpose:

The main purpose for a dedicated gym user is to enhance their workouts and ensure they have the correct information to make decisions about the best exercises and workouts they can do.

Tertiary Use Case - Weight Tracking

Actor: A student who focuses on weight gain or weight loss goals, and does not go to the gym

Goal: Stay motivated towards their health goals while tracking body weight

System Use:

- Regularly input weight
- Set and complete weight based goals (e.g. lose 5 kg)
- View progress graphs of their weight over time

Purpose:

While not the primary audience, the features (such as goal tracking and progress graphs) implemented for gym users make the app viable for those wishing to monitor their weight. Similarly, those struggling with weight benefit from the motivation that the app provides.

It must be noted that the app priority is still on gym use, this is just a bonus potential user case.

Functional & Non-Functional Requirements

We used interviews, survey results, and competitor analysis to shape our requirements list. As we're following Agile Scrum, we built an evolving Product Backlog and picked what to build during each Sprint Planning session based on user needs and what we could actually deliver.

We've grouped requirements, assigned priorities and estimated the time it takes to complete each function. All functional requirements are testable, so we can check if they were met.

Functional Requirements (FRs)

ID	Requirement	Description	Priority	Estimated Time To Complete	Source
FR1.1	Workout Logging	Users must be able to log exercises with sets, reps, and weights	High	High	Interview 1,2,4
FR1.2	Custom Workouts	Users can add their own workouts	High	High	Interview 1
FR1.3	Progress Visualisation	Users can view progress via graphs	High	High	Interview 1,3

FR1.4	Goal Tracking	Users can create, edit and complete goals	High	Low	Interview 2,5
FR1.5	Personal Info Tracking	Users can input personal data	Medium	Medium	Gymshark App Comparison
FR1.6	Access from home	All key features must be accessible from home screen	Medium	Low	Interview 4
FR1.7	Templates	Users can save a workout template	Low	High	Team Discussion
FR1.8	Exercise Notes	Users can attach notes to their workouts	Low	High	Team Discussion

FR Evolution Through Sprints

Because we followed an Agile Scrum methodology, the list of Functional Requirements wasn't final at the start. Instead, we built an initial Product Backlog based on interviews and research, then selected features for each Sprint Backlog depending on time, priority, and feasibility.

For example:

- In Sprint 1, we focused on the core features: Workout Logging (FR1.1), Custom Workouts (FR1.2), and Progress Visualisation (FR1.3) - these were rated as highest priority and were mentioned in almost every interview.
- Some features were refined or re-scoped. For instance, Access from Home (FR1.6) started as a general UI guideline, but turned into a concrete requirement to keep all major functions within one or two taps from the home screen.
- We also dropped or postponed ideas like exercise notes.

This constant re-evaluation is exactly what Scrum encourages — adapting the system design to real feedback and what's technically doable each sprint.

Non-Functional Requirements (NFRs)

ID	Requirement	Description	Priority	Estimated Time To Complete	Source
NFR1.1	Agile Process	Development must follow Scrum with regular sprints and reviews	High	High	Spec
NFR1.2	Accessibility	UI must be simple and accessible, with clear fonts and no clutter	High	Low	Interview 4
NFR1.3	No Paywalls	Users can view progress via graphs	High	High	Interview 2
NFR1.4	Visual Design	UI should follow a consistent visual style with minimal distractions	Medium	Medium	Interview 4
NFR1.5	Offline Use	Users can use application without access to internet	Medium	Low	Hevy App Feature Comparison
NFR1.6	Performance	Logging a workout should take <10	Low	Low	Team Discussion

		seconds from home screen			
NFR1.7	Privacy	Data (e.g., weight, goals) must be stored securely and not shared	Low	High	Interview 5

NFR Evolution Through Sprints

We defined a rough list of NFRs early on, but as development progressed, we adjusted these to better match what people actually needed or expected.

For example:

- NFR1.5 (No Paywalls) became a high priority after multiple people said they deleted apps when basic features were locked behind subscriptions (Interview 2). We initially considered having a small upgrade tier but scrapped it during Sprint Planning to avoid demotivating users.
- We added NFR1.6 (Performance) mid-way through Sprint 2 after testing showed that some actions (like logging a workout) felt slow — we set a rule that key actions should take less than 10 seconds from the home screen.

Overall, our non-functional requirements evolved alongside the app - not just being added at the start, but constantly reviewed in retrospectives and updated based on what actually helped users enjoy the experience more.

Summary

These requirements guided our Sprint Backlog selections and would continue to evolve as we get more feedback and test prototypes. Functional requirements are aimed at being tied to user stories (see Section 2.3), and each one was backed up with testing evidence as we go (see later section) - in line with test-driven development, which is a key Agile practice (CM12005 slides, [10]).

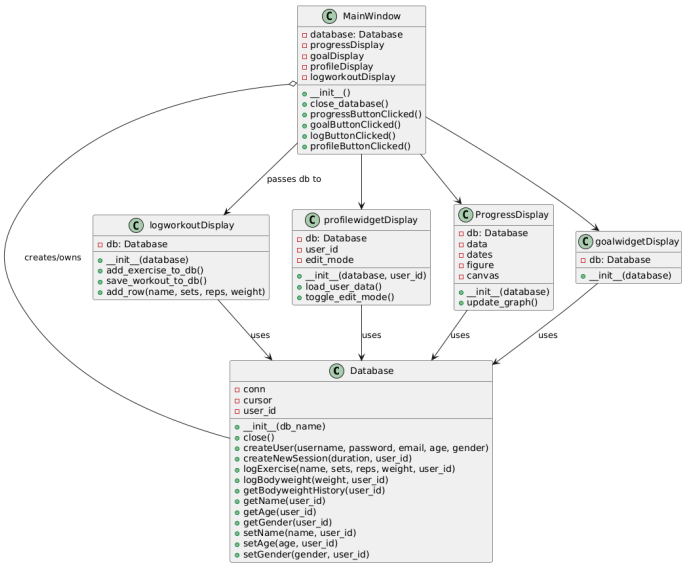
Design

Class Overview

This section outlines the design approach we used in developing our gym tracking app as well as the rationale behind it, specifically in terms of how we would meet the requirements as outlined in the project brief.

From the start, our design philosophy was centred around simplicity and ease of use. Most existing apps are bloated with unnecessary features to hook people in whilst keeping the basic necessities locked behind paywalls. Based on interviews and research, we found that recreational gym-goers specifically aren't keen on this. As a result, we aimed to develop a clean and intuitive system that prioritises accessibility without sacrificing meaningful tracking features.

As soon as the user opens the app, they are greeted by a minimalist looking home screen featuring four bright and clearly labelled buttons: **Progress, Goals, Log Workouts and Profile.** This design reflects our core aim at reducing unnecessary complexity and making navigation easy for all.



Above you can see a UML diagram, used to model the design approach we used in our project to show the relationships between key components.The **MainWindow** class acts as the central controller, featuring four methods that correspond to the main navigation buttons: **Progress, Goals, Log Workouts and Profile.** Each of these methods opens up a different corresponding display class, allowing for modularity and a very intuitive layout. The **MainWindow** also initialises the **Database** and then passes this into each class to handle consistent reading and writing. Our design also results in a clean architecture that's easy to maintain and test as each component is largely self contained. It also supports scalability as new features are easily addable as this would only require creating a new display class and linking it to the main window with another button.

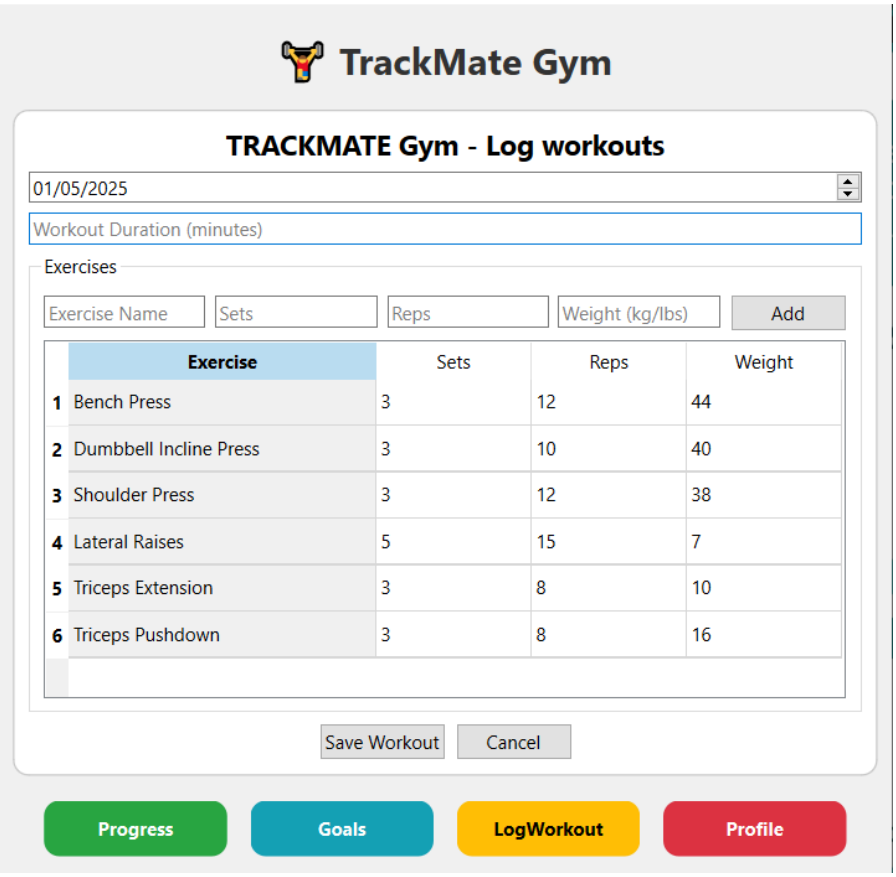
Class breakdown

Main Window

Acts as the central controller for the app. Initialises the database and aggregates (combines into one entity) logworkoutDisplay, profilewidgetDisplay, progressDisplay and goalwidgetDisplay. Also aggregates the database by creating a single instance and passing it to the other classes. This helps maintain consistent data handling throughout.

Log Workout Display
Handles the input of past workouts, allowing a user to input all exercises at once, before writing them to the database to ensure mistakes are less likely. Database is handled using the methods save_workout_to_db() and add_row().

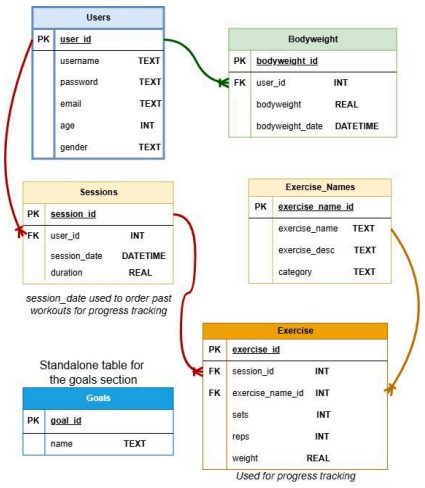
Profile Widget Display
Manages the user profile information, allowing data to be stored locally in the database for when the app is closed and reopened. The methods allow you to edit the data so you can track bodyweight over time in the progress tab



Database

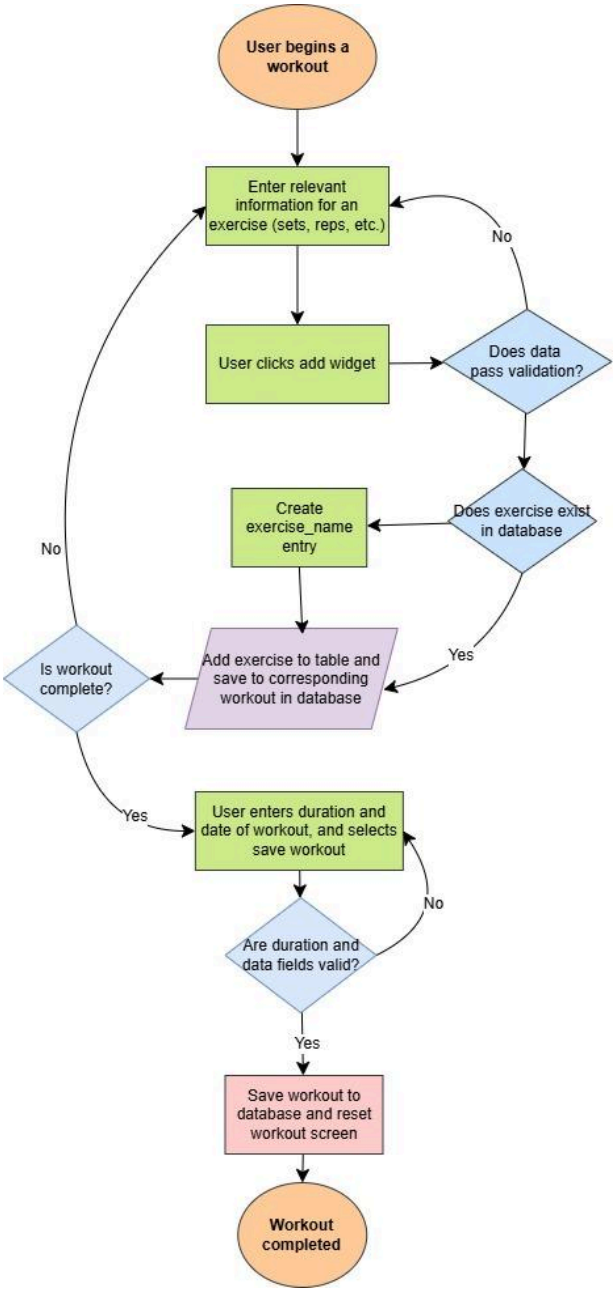
A key design principle when setting out our timeline was scalability and extensibility. Though we set long-term visions during the prototyping stage, we couldn't realistically be sure that we would meet all our goals. As such, it was vital we had a forward-minded approach that would enable us to expand the feature set without requiring major rewrites.
For example, we had heavily considered implementing support for multiple users, or perhaps adding features for tracking food habits and nutrition.

This design philosophy is especially reflected in the database structure. We ensured that core database design principles that are typically expected in a professional environment were followed. These included atomicity, normalization, referential integrity, and the avoidance of data redundancy. Tables were carefully structured with foreign keys and constraints to ensure consistent and logical relationships between entities such as users, sessions, and exercises.



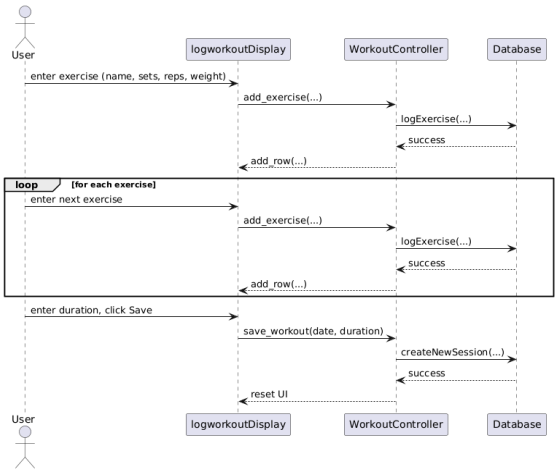
As a result of our modular approach, whereby a subgroup was assigned to each task, encapsulation was vital in ensuring seamless collaboration across components. The clearest example lies in the database module, which serves as the backbone of the program. All direct access to the SQLite connection and cursor — that is, self.conn and self.cursor — are kept private within the class. Furthermore, internal helper functions such as _getExerciseNameByID (prefixed with an underscore to indicate they are for internal use) are hidden from the frontend team. This prevents unintended usage or tampering from outside the database module.

Instead, external components interact with the database exclusively through clearly defined public methods such as logExercise(), getBodyweightHistory(), and createNewSession(). These methods are thoroughly validated to ensure the integrity of the database. For example, if a user tries to log an exercise with missing or non-numeric values, the input is rejected before reaching the SQL layer, reducing the risk of corrupt or inconsistent data. This also offers some protection against attacks such as SQL injections. We recognised that handling sensitive information carries a great deal of risk, and we made efforts to ensure the user's information would remain secure. In particular, passwords are stored in the database as a hash using the SHA-256 algorithm. Errors and exceptions are also handled internally to avoid crashing the frontend, ensuring a more robust and fault-tolerant experience. The save_workout_to_db() method for example tries to convert the input to a float and silently return should it fail.



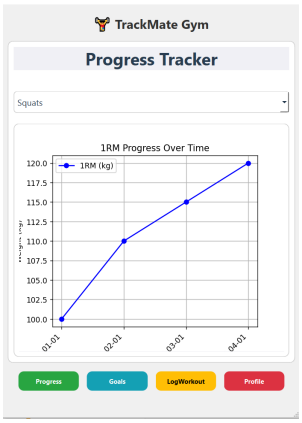
Logging workouts

The fundamental feature of the program is logging workouts (or sessions). Again, it embodies many of our aforementioned design principles. From a usability standpoint, we implemented defensive programming by clearing form fields after valid submissions, and rejecting invalid inputs (e.g. non-numeric durations) alongside. This not only improves user experience but also enforces data correctness across multiple layers.



Furthermore, the modular approach to logging – where exercises are added one at a time and committed to a session upon saving – opens a clear pathway for extendability. For instance, we could later support more advanced features such as RPE tracking, drop sets, and the deletion of past sessions without violating the existing structure. By designing with future flexibility in mind, the log workout display serves as a scalable and reliable interface between the user and the core database.
On the right is a sequence diagram demonstrating the interaction between the different layers when the user logs a workout.

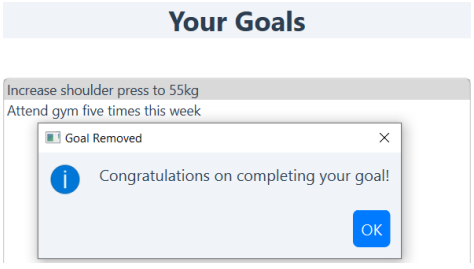
Progress Tracking



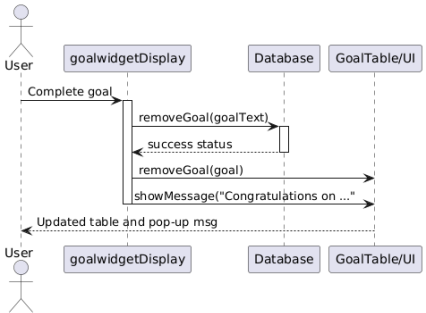
The bulk of our pioneering features are found in the progress tracking. As discussed in our initial research, many gym goers become frustrated by a lack of physical progress, and that leads to a considerable number giving up on the fitness journey.

We aimed to tackle this issue by serving the user with deeper insights like 1 rep max trends across all exercises and highlighting correlations with bodyweight. The correlation display is a pioneering feature of TrackMat and is our key differentiator from the competitors. As discussed in section “What People Told Us”, users were “put off by apps only focused on losing weight”. As weight is a fundamental metric for fitness, we sought to combine it with exercise charts to deliver a more nuanced and motivating view of their progress. This addresses our functional requirements of tracking multiple metrics, but also our non-functional requirements for greater user satisfaction.

Goals



Our most recently implemented feature is the goals tab. In order to reach our motivational requirements, we decided that such a feature would be the most useful and practical for our demographic. The user has the option to add new goals. Upon completing a goal, a pop up message will be displayed to provide immediate positive reinforcement and boost user engagement.
In its current form, it is entirely self-contained in that it manages its own UI, handles user input, and queries the database. This choice



was made as it allowed for a swift and smooth development – vital given the strict time constraints.
In line with our design principles, it has the infrastructure to facilitate many future endeavours, including adding a priority system, progress bars, and even eventually adding integration with the progress display graphs. Given the modular construction of the program, these enhancements can be added without impacting the rest of the program.

Software testing

Our testing consisted of two stages, one being done after sprint one and the other after sprint two. Testing ran in parallel to the development of the code, allowing us to find these problems early and fix them appropriately .

As we were a remote team, managing this project was difficult. However, we were able to use GitHub desktop to coordinate, run, analyse and respond to any test results that we got.

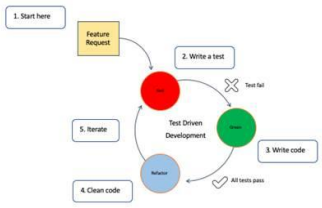
As each sprint began, we would set out clear goals in the meeting that we wanted to incorporate for that sprint, such as ensuring that all the buttons work or that a graph is outputted. By the time the development of code had started for each sprint, the tests were set out clearly, meaning that we were quicker and more productive.

We followed a test-driven development (TDD) approach which meant that coding, testing and refactoring were all tightly interwoven [1]. The TDD cycle consists of:

1. **Set clear tests.** These will be failed at the beginning of development as it has not yet been implemented.

2. **Develop the code to make the test pass.** The code will be extremely simple and just enough to make the test pass.

3. **Refactor.** As the test has now been passed, the code is refactored so that you make it the best version possible. This will include things like improving naming, maximising efficiency and optimising logic.



[2]

TDD worked well for us as it minimised bugs when we bought the whole project together and it also improved our collaboration as we were able to see what everyone was testing and add to each other's requirements.

Testing plans

Sprint 1

Feature	Requirement	Method	Result	Evidence Reference
A database is created	Data retrieval	Storing values in the database	Pass	N/A
Progress graph with dummy data	Graph generation	Check that the correct graphs have been plotted	Pass	2
Log workouts	User input	Check if the user can input something	Pass	3
Profile tab	User input and user interaction	Check if the user can click on the tab and input personal data	Pass	4
UI functionality	User interaction	Ensure buttons work correctly by having someone interact with it	Pass	N/A

As you can see, for sprint 1, we focused more on the easier development and making sure that the backbone of the design was implemented well.

Sprint 2

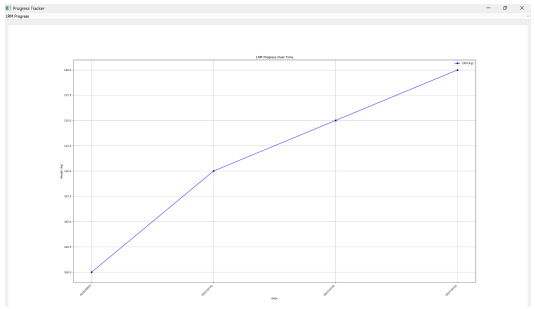
Feature	Requirement	Method	Result	Evidence Reference
Database code refactored to have getters and setters	Data retrieval	Testing each method to get data and set data in the database	Pass	1, 2
Progress graph uses database data	Displaying graph	Cross check database data with graph data	Pass	2
Workouts are saved to database	Data input	Check whether the database has gotten the correct data	Pass	N/A

UI functionality	All the sections must link and the app must run smoothly	Have a real user test it out	Pass	2

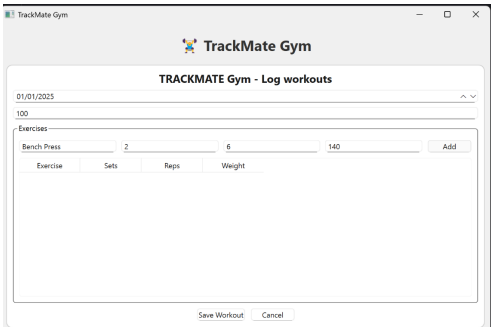
Evidence of testing

Sprint 1

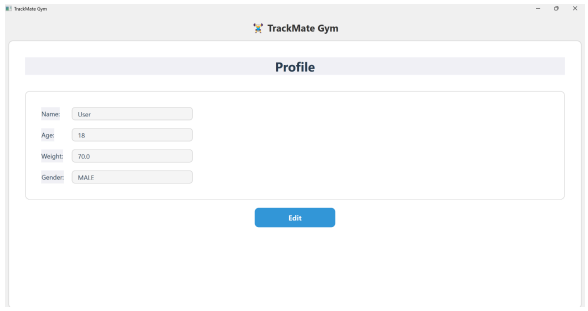
2.



3.

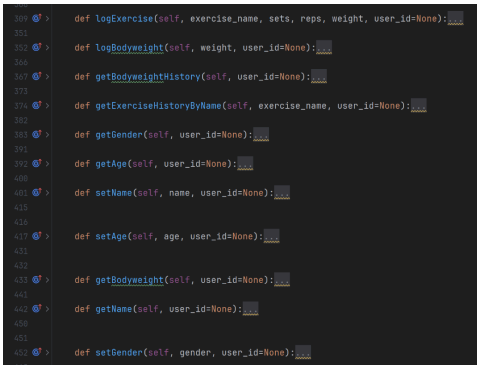


4.

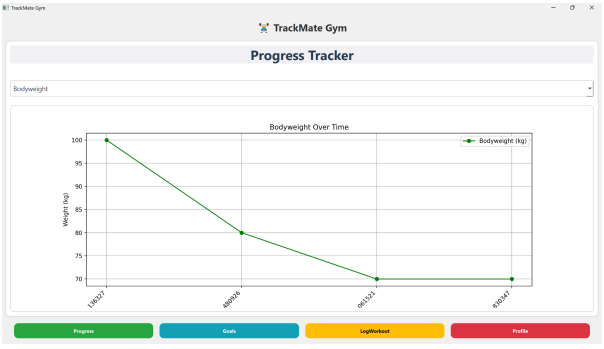


Sprint 2

1.



2.



Reflection and Conclusion

Critique of the Requirements Specification

Our requirements specification was grounded in both competitive analysis and user-centred research, which gave us a well-informed and realistic foundation for development. However, there were strengths and areas that could be improved.

Strengths

- Requirements based upon collected evidence: we began by researching popular fitness apps (e.g. Hevy, Strong, Gymshark) and gathering feedback from Reddit communities and real students through interviews and surveys. This gave us a strong understanding of common flaws in gym applications (e.g. paywalls, cluttered UIs) and high-value features (e.g. custom workouts, visual progress tracking).
- Prioritising user input: the final features we prioritised (custom workouts, graph-based progress tracking, goal setting) directly responded to user feedback. This alignment ensured our application sprint addressed actual needs and not just assumptions.
- Selecting features based on priority: We deliberately dropped lower-priority features like sleep tracking due to privacy concerns and lack of user demand. This disciplined scoping helped us stay focused and deliver usable features in Sprint 1.
- Agile methodology: Our requirements were expressed in a form that supported Agile workflows i.e. user-centric and iterative so that initial requirements aren't fixed, but are treated as a starting point for further refinement across the sprints.

Weaknesses

- Accounting for edge cases: some user needs, such as how to handle missed workouts or adjusting workout logs, weren't considered until they emerged during development. Including more diverse cases such as injury recovery and irregular users could have broadened our understanding of real-world scenarios.
- Delayed Feedback Loops: although we engaged users at the start, we could have maintained a tighter feedback loop during sprints. For example, quick user demos mid-sprint might have caught design or usability issues earlier.
- Vague non-functional requirement definitions: we didn't clearly define expectations around performance (e.g. how fast data should load) or security (e.g. privacy of personal statistics). These should have been explicitly stated and tested.

Future Improvements

- Adopt clear user stories: future projects should start each feature as a user story (e.g., As a gym-goer, I want to track custom workouts so I can optimise my routine) with measurable success criteria. This will make both testing and progress tracking smoother.
- Schedule regular user check-ins: more frequent, informal user demos or prototype reviews (even if it is just UI drafts) can help refine priorities mid-sprint and avoid costly changes (with respect to time) late in development.

Critique of the Software Design

Our design evolved over the sprints, reflecting feedback and changes in user needs. Some strengths and weaknesses were evident:

Strengths

- Modular design: we used a component-based architecture that allowed us to update data collection and visualization modules independently.
- Scalability: the system is set up to incorporate new data types easily without changing core functionality.

Weaknesses

- Inconsistent UI/UX Design: due to parallel development, user interface components lacked a unified look in early sprints - this can affect user experience and causes refactoring in later stages of the development.
- Overengineering in early sprints: we initially designed for more complex features (e.g., predictive analytics) that weren't possible within our sprint timeline.
- Lack of visual appeal: since our development process focused mainly on pure functionality of features (which it achieved), the UI lacked the modern, engaging feel that first-time users would expect from fitness apps.

Future Improvements

- Begin with prototype design layout, and evolve them based on user feedback.
- Collaborate with a UI designer or adopt a design system to improve visual consistency and user engagement.
- Align functionality decisions with sprint goals — focus on simplicity early on and scale complexity later.

Critique of Testing Approach

Testing was integrated into our Agile lifecycle, but effectiveness varied across stages:

Strengths

- Test-driven development (TDD): we adopted TDD for core modules, which made sure crucial features were reliable (improving code quality) and allowed potential issues to be spotted early in development.
- Usability testing sessions after sprints: each sprint concluded with a round of usability testing and bug tracking, which enabled us to quickly identify issues and gather real feedback on the current version of the app.

Weaknesses

- Limited testing approaches: our testing covered core app logic but didn't expand to further testing techniques such as integration testing (interaction between different components) or UI testing (behaviour of the user interface).
- Insufficient user testing early on: initial user feedback came late in the process, which limited our ability to iterate on usability concerns.

Future Improvements

- Expand automated testing to UI layer and add integration testing.
- Schedule regular usability testing with potential users during each sprint.
- Create more comprehensive test data which involves further edge cases.

Conclusion

This project demonstrated the value of following an Agile SDLC in a rapidly evolving context like Personal Informatics. Our iterative approach allowed us to refine our concept through user input and adjust priorities in real time. However, key lessons include the importance of clear, testable requirements, the need for consistency in design, and the critical role of comprehensive testing

at all layers. Applying these insights in future projects would lead to more efficient development and a higher-quality end product.

Evidence of Agility in the Software Process

1. Sprint-Based, Iterative Planning

From the very first meeting , the team adopted an Agile structure built around two sprints. Each sprint had clearly defined goals and deliverables. Sprint 1 focused on developing the core UI and backend, while Sprint 2 was scoped to build upon this foundation with additional features. Sprint checkpoints and planning sessions allowed for adjustments as the project progressed.Key deliverables achieved in Sprint 1 included:

- A database structure to support data logging
- A clean, functional UI using PyQT5
- A workout logging system
- Graphing tools to visualize progress
- A self-managed goal-setting system
- A basic profile page

Sprint 2 began after a review of Sprint 1's outcomes, confirming that core goals had been met. The team planned to build upon the existing framework by refining existing modules and adding new features. For example, the goals module was expanded to support new motivational prompts, and graph functionality was improved to allow more meaningful comparisons of user data.

New task assignments were issued: the database team worked on optimizing data flow for additional inputs while The UI team continued refining the layout. The feature teams focused on making the app more user-friendly and responsive. As with Sprint 1, feedback and check-ins allowed us to adapt in real time, identify bottlenecks, and ensure all features aligned with user needs and project requirements.

2. Continuous Collaboration and Regular Scrums

The group met frequently to review progress, adapt plans, and reassign tasks. Each meeting served as a Scrum-style check-in, enabling ongoing visibility into the team’s progress and promoting accountability. The use of GitHub for code collaboration further supported Agile practices by enabling asynchronous contributions and version control

3. Role Flexibility and Cross-Functional Work

Responsibilities were divided based on individual strengths, but roles evolved as needed. For instance, UI tasks, database integration, and feature development were reassigned across meetings to maintain balanced workloads and reflect shifting priorities. This reflects Agile’s principle of self-organizing along with cross-functional teams

4. Technical Adaptability and Incremental Delivery

The team demonstrated technical agility by switching tools when better options emerged—such as replacing PyGy with PyQT5 after initial tests, and selecting SQLite over MySQL for simpler implementation. Each technical decision was made based on feasibility, team familiarity, and sprint efficiency

Evidence of Evolving Requirements
1.User-Driven Design: Initial brainstorming was followed by interviews and questionnaires, which revealed user needs such as motivational systems, customizable goals, and progress visualization. These features became central to the app's functionality over time.

2.Incremental Feature Development: By Meeting 5, the team had broken the app into core functional areas—goals, progress, workout logging, and profile—which aligned with the most common user requests. This modular design not only reflects good Agile scoping but shows how features evolved from general ideas to concrete tasks.

3.Refinement Through Prototyping and Discussion: Technical plans such as database integration and UI structure became clearer as the group began building and testing. For instance, the decision to forgo multi-user support in the first version reflects a practical understanding of scope and technical limitations.

4.Planning for Future Iterations: The team consistently planned for upcoming work based on current sprint outcomes, using Agile's emphasis on iterative progress. Discussions about what features to include in Sprint 2 were already underway as Sprint 1 neared completion

References

Rapp, A. and Cena, F. (2016). *Personal informatics for everyday life: How users without prior self-tracking experience engage with personal data. International Journal of Human-Computer Studies,* [online] 94, pp.1–17. Available from: https://www.researchgate.net/publication/303635819_Personal_Informatics_for_Everyday_Life_How_Users_without_Prior_Self-Tracking_Experience_Engage_with_Personal_Data [Accessed 30 April 2025]

Li, I., Medynskiy, Y., Froehlich, J., and Larsen, J.E. (2012). *Personal Informatics in Practice: Improving Quality of Life Through Data*. [online]. Available from: https://makeabilitylab.cs.washington.edu/media/publications/Personal_Informatics_in_Practice_Improving_Quality_of_Life_Through_Data_C9S1whc.pdf [Accessed 30 April 2025]

Kersten-van Dijk, E.T., Westerink, J.H.D.M., Beute, F. and IJsselsteijn, W.A. (2017). *Personal Informatics, Self-Insight, and Behaviour Change: A Critical Review of Current Literature. Human–Computer Interaction*, 32(5-6), pp.268–296. Available from: https://www.tandfonline.com/doi/full/10.1080/07370024.2016.1276456 [Accessed 30 April 2025]

Dr. Muscle (2023). *Strong Workout App Review: Is It Worth It?* [online] Dr. Muscle. Available from: https://dr-muscle.com/strong-workout-app-review/ [Accessed 30 April 2025]

Reid, G. (2023). *HEVY Workout App: Independent Review*. [online] Dr. Muscle. Available from: https://dr-muscle.com/hevy-workout-app-review/ [Accessed 30 April 2025]

Reid, G. (2023). *Gymshark Workout App: Honest Review & Alternative.* [online] Dr. Muscle. Available from: https://dr-muscle.com/gymshark-workout-app-review/ [Accessed 30 April 2025]

BUCS (2020) *British Active Students Survey: Higher Education 2019/2020 Report* [online]. Available from: https://www.bucs.org.uk/static/f352c43b-621d-44ae-8a44cfeb2196771c/BASS-2019-HE.pdf [Accessed 30 April 2025]

Agile Alliance (2018). *What is Test Driven Development (TDD)?* [online] Agile Alliance. Available at: https://www.agilealliance.org/glossary/tdd/ [Accessed 30 April 2025]

Steinfeld, G. (2020). *5 steps of test-driven development. [online] IBM Developer*. Available at: https://developer.ibm.com/articles/5-steps-of-test-driven-development/ [Accessed 30 April 2025]

Appendices

Group Contribution Form

Group Member	Email	Contribution	Comment
Nicholas Legh-Smith	nls40@bath.ac.uk	10	Took on a leading role in creating the project and coded many parts of it, also helping other users to get familiar with the software we were using.
Amarveer Johal	asj71@bath.ac.uk	8	Good group member who fully participated.
Taha Ahmed	tma42@bath.	8	VERY good group member who

	ac.uk		fully participated.
Rishabh Madhok	rm2870@bath.ac.uk	8	Good group member who fully participated.
Samsul Hoque	sh3295@bath.ac.uk	8	Excellent group member who fully participated.
Oscar Horsfall	owh31@bath.ac.uk	10	Took a leading role in organising the group, organising meetings and taking notes. Also did a significant amount of coding for the project.
Tanay Patel	thp39@bath.ac.uk	8	Good group member who fully participated.
Luka Matthew Ian Cooke	lc2880@bath.ac.uk	8	Good group member who fully participated.
Aaryan Lal	al2698@bath.ac.uk	8	Good group member who fully participated.
Adil Rasheed	ar3128@bath.ac.uk	8	Good group member who fully participated.

Meeting Notes

Date: 19/03/2025

Meeting 1 (Intro & Brainstorm)

Attendees: Luka Cooke, Oscar Horsfall, Adil Rasheed, Tanay Patel, Nicholas Leigh-Smith, Samsul Hoque, Rishabh Madhok, Taha Ahmed, Aaryan Lal, Amarveer Johal

Agenda

Topic	Details
Read assignment to gain understanding	We went through the coursework and got a clear understanding of what was expected - following an Agile approach, working in sprints, and

Topic	Details
	focusing on both the development process and the final product. We looked through the mark scheme to prioritise our efforts.
Discussed project ideas (brainstorm)	Amar came up with an idea in the gym after realizing they lacked a way to properly track their progress beyond basic fitness apps. We debated between **TrackMate**, which would track fitness, diet, and productivity, and a study tracker focused on revision habits. In the end, we felt **TrackMate** was more versatile and aligned better with the project requirements.
Structure & timeline of project	After reading the specification, we made it clear to follow Agile with two sprints, each lasting around 2-3 weeks. The first sprint would focus on setting up data collection and storage, while the second would refine analysis and visualization. Weekly Scrum meetings were set to keep track of progress, make adjustments, and get tutor feedback.
Discussed member strengths & role assignments for the week	We went over everyone's strengths and discussed how we could utilize them later on. Since we needed real user insights, we assigned each other to take on interviews and market research, while others started setting up the technical framework. We assigned tasks to be completed by our next meeting on Friday, making sure everyone had something to contribute before we checked in again.

Action items

- ☑ ~~Interviews with target audience: Luka, Amar & Oscar~~
- ☑ ~~Questionnaire (for more quantitative data): Taha, Samsul, Aaryan~~
- ☑ ~~Market research into fitness software: Rishabh, Adil, Nick, Tanay~~
- ☑ ~~Meeting Summary: Luka~~

Next Steps & Goals (for meeting on 21/03/2025)

- Check in on everyone's progress with their tasks
- Go over initial thoughts on features and direction
- Gather any early findings from research and interviews
- See if there are any challenges so far and how to approach them
- Start thinking about the technical setup and requirements
- Plan next steps and assign new tasks

Date: 21/03/2025

Meeting 2 (Follow-up on market research)

Attendees: Luka Cooke, Oscar Horsfall, Adil Rasheed, Tanay Patel, Nicholas Legh-Smith, Samsul Hoque, Rishabh Madhok, Taha Ahmed, Aaryan Lal, Amarveer Johal

Agenda

Topic	Details
Follow-up on last meetings tasks	Since it has only been two days since the last meeting, the tasks are ongoing. Some interviews have been had and a questionnaire has been drafted to be filled out by other students. The project analysis should be finished so that we can define project requirements next week.
Asked Dr. Hayes about development choices	Dr. Hayes confirmed that Python could be used for the project provided it is Object-Oriented. He also said that the software does not need to be too complex, and data storage could be done in files.
Thought about initial requirements for the gym app	We discussed the requirements for the app that came up in interviews and the problems that we need to solve, such as systems to help motivate users, and ways of displaying data over

	time. Discussed finding some sources/studies to find common problems with gym apps.

Action items

- ☑ ~~Interviews with target audience: Luka, Amar & Oscar~~
- ☑ ~~Questionnaire (for more quantitative data): Taha, Samsul, Aaryan~~
- ☑ ~~Market research into fitness software: Rishabh, Adil, Nick, Tanay~~
- ☑ ~~Decide project requirements~~

Next Steps & Goals (for meeting on 24/03/2025)

- Check in on everyone's progress with their tasks
- Go over initial thoughts on features and direction
- Gather any early findings from research and interviews
- See if there are any challenges so far and how to approach them
- Start thinking about the technical setup and requirements
- Plan next steps and assign new tasks
- Specify the requirements for the project

Date: 24/03/2025

Meeting 3 (Planning for the start of the sprint)

Attendees: Luka Cooke, Oscar Horsfall, Adil Rasheed, Tanay Patel, Nicholas Leigh-Smith, Samsul Hoque, Rishabh Madhok, Taha Ahmed, Aaryan Lal, Amarveer Johal

Agenda

Topic	Details

Follow-up on last meetings tasks	Project Analyses, Interviews and Questionnaires have been mostly completed, we started to expand upon our initial requirements for the gym app.
Started planning for designing and creating the app	Decided to use python and the GUI PyGy to create the app, and a GitHub will be set up to help people collaborate on the app. Discussed that people should become familiar with PyGy and GitHub before the next meeting on friday. Discussed that on friday the layout of the app should be decided so we can start allocating tasks to be coded.
Expanded upon initial requirements	We discussed the requirements for the app that came up in interviews, questionnaires and analyses, coming up with many requirements such as Settings and changing goals, having streak based motivation systems, letting users see their progress with graphs, etc.

Action items

- ☑ ~~Set up GitHub for everyone~~
- ☑ ~~Make sure team member are comfortable with PyGy and GitHub~~
- ☑ ~~Confirm layout of the app~~
- ☑ ~~Ensure the project includes 6 sources (three in analysis)~~
- ☑ ~~Finalise project requirements~~

Next Steps & Goals (for meeting on 28/03/2025)

- Check in on everyone's progress with their tasks
- Decide on project requirements to be used in the first sprint
- Start work on the sprint
- See if there are any challenges so far and how to approach them
- Plan next steps and assign new tasks
- Specify the requirements for the project

Date: 28/03/2025

Meeting 4 (UI layout & Sprint Start)

Attendees: Luka Cooke, Oscar Horsfall, Nicholas Leigh-Smith, Samsul Hoque, Rishabh Madhok, Aaryan Lal, Amarveer Johal

Agenda

Topic	Details
Explored UI implementation with PyQT5	We started exploring PyQT5 to create the UI for our application. We looked at tutorials and documentation to get a basic understanding of designing layouts and adding widgets. Initial tests were successful, and we feel confident about proceeding with this approach.
Discussed database implementation ideas, debated usage of mySQL and SQLlite	Debated and discussed potential database implementations, decided that SQLlite would be the most optimal approach, firstly because of team member familiarity but also because it is much simpler to set up and maintain for a large team of programmers.
Task delegation for sprint	We began working on the frontend as part of our first sprint. The goal is to create a basic, functional UI and ensure that it communicates with the backend smoothly. We are focusing on adding essential features and validating the initial design.
Started integrating frontend in sprint	We went over everyone's strengths and discussed how we could utilize them later on. Since we needed real user insights, we assigned each other to take on interviews and market research, while others started setting up the technical framework. We assigned tasks to be completed by our next meeting on Friday, making

Topic	Details
	sure everyone had something to contribute before we checked in again.

Action items

- ☑ ~~Refine UI Design: Oscar, Nick~~
- ☑ ~~Set Up Data Flow for Initial Prototype~~
- ☑ ~~Writeup a finalised list of key features that match user requirements + a structured design diagram to demonstrate our vision~~
- ☑ ~~Create a mock drawing for UI.~~

Next Steps & Goals (for meeting on 21/03/2025)

- Finalize the basic UI with PyQT5 and begin connecting it to the backend.
- Conduct initial testing to ensure frontend and backend interaction works as expected (As pers Dr. Hayes
- Have a workable model available for next meeting.

Date: 1/04/2025

Meeting 5 (Sprint Checkup)

Attendees: Luka Cooke, Oscar Horsfall, Nicholas Leigh-Smith, Samsul Hoque, Rishabh Madhok, Aaryan Lal, Amarveer Johal

Agenda

Topic	Details
Decided the requirements for Sprint 1	Selected a number of requirements that should be finished after the first sprint
Decided on a UI	Decided the UI would be made of 4 main tabs which could be delegated, along with delegation of the backend

Topic	Details
Task delegation for sprint	Delegated the tasks for the sprint between everyone, ensuring people has similar amounts of work.
Discussed how database would be integrated with frontend	Decided how the database would work, how it could be interacted with from the fronted, and that having multiple users isnt necessary for a phone app.

Action items

- ☑ ~~Database Creation and Management - Rishabh, Nick, Aaryan and Samsul~~
- ☑ ~~UI Management - Luka and Taha~~
- ☑ ~~Goals Tab - Tanay and Oscar~~
- ☑ ~~Profile Tab - Adil and Amar~~

Next Steps & Goals

- Ensure the app is working with a clean UI
- Allow the user to log data and view it
- Ensure people have made progress or finished their tasks
- Decide on features to be added for second sprint once this is completed

Date: 13/04/2025

Meeting 6 (End of Sprint 1)

Attendees: Luka Cooke, Oscar Horsfall, Nicholas Leigh-Smith, Samsul Hoque, Rishabh Madhok, Aaryan Lal, Amarveer Johal

Agenda

Topic	Details
Checked that sprint 1 has been fully completed	Sprint one had been completed to a good degree
Decided on a start date and length for sprint 2	Sprint two will be from friday to friday in the following week
Checked that the current direction of the project was still satisfactory	The project still works for the original guidelines of a motivation based app
Test some of the features implemented in sprint 1	Make sure that sprint 1 is fully functional

Action items

- ☑ ~~Database Creation and Management - Rishabh, Nick, Aaryan and Samsul~~
- ☑ ~~UI Management - Luka and Taha~~
- ☑ ~~Goals Tab - Tanay and Oscar~~
- ☑ ~~Profile Tab - Adil and Amar~~

Next Steps & Goals

- Decide on features for the second sprint
- Start the second sprint
- Add the named features

Date: 19/04/2025

Meeting 7 (Sprint 2 Start)

Attendees: Luka Cooke, Oscar Horsfall, Nicholas Leigh-Smith, Samsul Hoque, Rishabh Madhok, Aaryan Lal, Amarveer Johal

Agenda

Topic	Details
Decided the requirements for Sprint 2	Selected a number of requirements that should be finished after the second sprint sprint
Ensured that the features added for sprint 2 would result in a completed app	Decided the UI would be made of 4 main tabs which could be delegated, along with delegation of the backend
Task delegation for sprint 2	Delegated the tasks for the sprint between everyone, ensuring everyone has similar amounts of work.

Action items

- ☑ ~~UI Management - Luka and Aaryan~~
- ☑ ~~Progress Tab - Nick, Rishabh, Oscar and Amar~~
- ☑ ~~Logging Workouts - Taha, Samsul, Adil and Tanay~~

Next Steps & Goals

- Ensure the app is working with a clean UI
- Allow the user to log data and view it
- Ensure people have made progress or finished their tasks

Date: 25/04/2025

Meeting 8 (Sprint 2 End)

Attendees: Luka Cooke, Oscar Horsfall, Nicholas Leigh-Smith, Samsul Hoque, Rishabh Madhok, Aaryan Lal, Amarveer Johal

Agenda

Topic	Details
Ensure that sprint 2 is completed	Look at sprint 2 features and see if they’re all completed
Internally test sprint 2 features	Ensure the features are robust and don’t have bugs
Get an outside testers perspective on the app	Make sure the app is useable for our target audience

Action items

- ☑ ~~Test sprint 2 features~~
- ☑ ~~Make sure app is good for users~~
- ☑ ~~Ensure all features have been added~~

Interview Transcripts

Interview 1 (20yr old Male - Very Physically Active)

How often do you go to the gym?

Five times a week

Do you use any Gym companion apps and if so, which?

I use Hevy

What is the best feature of the app?

Having graphs to track my progress and being able to compare my workouts to other people I know

Are there any problems that the app has solved for you?

It saves me from forgetting my workouts and makes it easier to progress in reps and sets.

What would you say is the biggest obstacle to regularly exercising is?

Finding time to go

Do you think streak-based systems help to motivate you?

Yes, to make sure I am consistent to going

Are there any features of your app that you think are detrimental?

You have to pay for the more advanced features and also I don't like how you can't include drop sets in the inputs for sets and reps.

Are there any features that you would like to be added to your app?

Have it so you can include drop sets and have some motivation things like showing the total weight you've lifted and equate that to real world objects.

Interview 2 (19yr old Male - Moderately Physically Active)

How often do you go to the gym?

Three times a week

Do you use any Gym companion apps and if so, which?

No but I have been considering it

What are you looking for in the app?

Basic things like tracking sets, reps, weights, calories, etc.

Why aren't you using an app at the moment?

You have to pay for most of them nowadays and the free versions aren't very good. Plus I am not very consistent with inputting my data.

What would you say is the biggest obstacle to regularly exercising is?

Motivation

Do you think streak-based systems help to motivate you?

Yes, to make sure I am consistent to going

Are there any specific features you would like to see in this app?

I would like to see weekly goals and things like badges and levels as an award system. Also I would like a social system to see what other users are up to on the app. Finally I would like to be able to store photo progress on here too.

What would you not want to see from this app?

I don't want a clunky interface or have to pay for anything.

Interview 3 (20yr old Male - Athlete)

How often do you go to the gym?

Two times a week

Do you use any Gym companion apps and if so, which?

I use MyFitnessPal

What is the best feature of the app?

The graph of your weight and it also shows your individual macros for what you eat.

Are there any problems that the app has solved for you?

It helps me monitor my diet and also helps me keep track of my sets and reps

What would you say is the biggest obstacle to regularly exercising is?

Finding time to go

Do you think streak-based systems help to motivate you?

Yes, to make sure I regularly train.

Are there any features of your app that you think are detrimental?

The design doesn’t look good and the app doesn’t offer any motivation to train.

Are there any features that you would like to be added to your app?

Push notifications, tracking goals, GIFs to demonstrate each exercise and maybe a favourite button to favourite exercises.

Interview 4 (19yr old Male - Very Physically Active)

Q: *What motivates you to keep going to the gym regularly?*

A: I like the sense of routine it gives me. Plus, seeing progress over time really keeps me going.

Q: *Do you track your workouts or progress in any way?*

A: Yeah, I jot things down in a notes app, but it's a bit messy.

Q: *Have you tried any gym-related apps in the past?*

A: I've tried a few, like Fitbod and Strong, but I didn't stick with them long.

Q: *Why do you think you didn't stick with those apps?*

A: Some were too complicated or had too many ads. Others just didn't feel personal to my goals.

Q: *What features would make you actually want to use a gym app long-term?*

A: Simplicity is key. I'd want clear progress tracking, maybe some kind of personalised plan, and reminders that aren't annoying.

Q: *Would you be interested in any social or community aspects within an app?*

A: Definitely. Being able to share progress or challenges with friends would help keep me accountable.

Q: ***What's something you wish gym apps did better?***

A: Adapting to how I'm feeling. Like if I'm low energy one day, it could recommend a lighter workout instead of making me feel guilty.

Q: ***Would you find it helpful if an app helped track nutrition alongside workouts?***

A: For sure. It's hard to keep track of everything, so having it all in one place would make things easier.

Interview 5 (18yr old Female - Low Physical Activity)

Q: ***Would you describe yourself as someone who goes to the gym regularly?***

A: Not really. I've had phases where I went for a few weeks, but I always fall off. I want to get into it properly, though.

Q: ***What's held you back from going more often?***

A: Motivation, mainly. Also, not knowing what to do once I'm there. The gym can feel a bit intimidating, especially when it's busy.

Q: ***Have you tried using any gym or fitness apps before?***

A: Yeah, I downloaded a few, but they usually expect you to already know what you're doing. I end up feeling overwhelmed and give up.

Q: ***What kind of features would actually help you get started?***

A: Honestly? Simple routines for beginners, clear instructions, maybe even videos that show how to use the machines. And I'd love something that eases me in, not just a random full-on workout.

Q: ***Would you use any kind of motivational system, like streaks or progress badges?***

A: I think so, as long as it doesn't feel too pressured. Something like small, achievable goals would be nice — like "go twice this week" or "try a new exercise."

Q: *Do you feel current fitness apps speak to women like you?*

A: Not really. A lot of them are either super intense or all about getting skinny. I just want to feel better in myself, not be yelled at by a trainer avatar.

Q: *Would you be interested in connecting with others through the app?*

A: Maybe. If it was a supportive space, like a beginners' group or people with similar goals, that could help me stay on track.

Q: *Is there anything else that might help you stick to a fitness habit?*

A: Something that fits into my life. Like reminders that aren't annoying, and a way to log how I'm feeling — not just physical stats. Some days I need rest, not guilt.

Key Takeaways – Interviews (1-5) Summary

What Users Want

- **Progress Tracking with Graphs**
  Users love seeing visual data like total weight lifted or workout progress over time (Interview 1, 3).

- **Motivation Features**
  Streaks, small weekly goals, badges, and levels were mentioned as helpful by both active and less active users (Interviews 1, 2, 5).

- **Beginner-Friendly Workouts**
  Clear, simple routines — especially for beginners — are in demand. Video demos or tips would make users feel more confident (Interview 5).

- **Custom Workout Logs**
  Users want to track sets, reps, and weights easily. Being able to customise inputs (like drop sets) would improve experience (Interview 1).

- **Simple & Clean Design**
  A clunky interface puts people off. Users prefer apps that are easy to navigate without

distractions or unnecessary clutter (Interview 4).

What to Avoid

- **Paywalls**
  Users dislike having to pay for core features (Interviews 1, 2).
- **Overly Complex Interfaces**
  Too many ads or confusing layouts made some users stop using apps altogether (Interview 4).
- **Too Much Focus on Weight Loss** Not using the app to get skinny/gain weight —feel healthier should be the priority. (Interview 5)

Competitor Analysis

Comparison of Hevy, MyFitnessPal, Strong and a notepad and pen.

Feature	*MyFitnessPal*	*Hevy*	*Strong*	*Notepad*
Calorie Tracking	Very Good/Excellent*	None	None	Basic
Workout Tracking	Average	Excellent	Very Good/Excellent*	Basic
UI	Average	Very Good	Good	User dependant
Progress Tracking	Good	Good	Good	Basic
Social	Yes	Yes	No	No
Ease of Use	Moderate	Easy	Easy	Easy

* - Paid version

Missing Features

- Mental Health tracker
  - Mood before/after workouts#
  - Meditation
- Making it like a game
  - Levelling up based on weight/consistency
  - Challenges

    - Competitions with friends
- Recommendations for recovery

Hevy - Market Research

Key features

Hevy tracks several important workout metrics:

- Exercise history and progress tracking
- Volume progression (total weight lifted)
- Personal records (PRs) for different exercises
- Workout duration and frequency
- Rest times between sets
- Exercise completion rates
- Weight, sets, and reps for each exercise

Pros of Hevy

1. **User-Friendly Interface**
    - Clean, intuitive design that's easy to navigate
    - Quick exercise logging with minimal taps
2. **Comprehensive Exercise Database**
    - Large library of pre-loaded exercises with proper form descriptions
    - Ability to create custom exercises
3. **Detailed Progress Tracking**
    - Visual graphs showing progress over time
    - Weight, volume, and PR tracking
    - Historical workout data easily accessible
4. **Social Community Features**
    - Option to follow friends and view their workouts
    - Community support and motivation
    - Ability to share workouts to social media
5. **Workout Templates**
    - Save and reuse custom workout routines
    - Access to community-created templates
6. **Free Core Functionality**
    - Most essential features available without subscription
    - Premium features reasonably priced
7. **Offline Functionality**
    - Works without internet connection
    - Syncs when connectivity is restored

Cons of Hevy

1. **Limited Programming Features**
    - No built-in progressive overload scheduling
    - Lacks periodization planning tools
    - No automatic weight progression recommendations
2. **Restricted Analytics in Free Version**
    - Advanced analytics require premium subscription
    - Limited historical data visualization
3. **No Integrated Nutrition Tracking**
    - No calorie or macro tracking capabilities
    - No integration with nutrition apps
4. **Limited Cardio Tracking**
    - Primarily focused on strength training
    - Basic cardio logging compared to dedicated cardio apps
5. **No Coaching Features**
    - Lacks functionality for trainers to program for clients
    - No form check or technique feedback
6. **Occasional Sync Issues**
    - Some users report data synchronization problems
    - Potential for workout data loss if not properly synced
7. **Limited Equipment Filtering**
    - Could improve exercise suggestions based on available equipment
    - Gym versus home workout differentiation could be enhanced

Market research case study: Gymshark Training App

Aim and appeal

This app aims to create an all-inclusive personalised, customizable fitness training hub where users can build and find workouts that will suit their personal needs and fast track them towards their fitness goals. It appeals to all demographics and ages as it caters to fitness which can appeal to all.

Key features

· Collects personal information upon setup of the account: age, lifting weight unit, name, birthdate.

· Daily habits list that can be selected from 3 areas of focus: training, health, mindset – feature includes a daily reminder push notification functionality.

- Progress charts used to track progress on PBs in the gym over time along with challenges to motivate you to work harder to reach your goals.
- Rewards scheme.
- Personal workout builder and workout plan builder customised from a selection of premade workouts.
- A selection of workouts and workout plans by popular creators and fitness coaches which can be filtered by your goal, duration, workout schedule and equipment available.

Strengths

- Extensive Workout Library: The app provides access to over 450 free workouts and programmed training plans curated by industry experts, catering to various fitness levels and goals.
- User-Friendly Design: Its intuitive interface allows users to navigate easily through workouts and programs, making it accessible for both beginners and seasoned fitness enthusiasts.
- Influencer-Led Programs: Leveraging Gymshark's strong ties with fitness influencers, the app features routines and classes curated by popular fitness personalities, adding motivation and variety to workouts.
- Free Access: Unlike many fitness apps that require subscriptions, the Gymshark Training App offers its extensive content entirely free of charge, providing significant value to users.
- Rest Timer Functionality: The app includes a rest timer to guide users through their workouts, ensuring appropriate rest intervals between sets.

Weaknesses

- Lack of Personalization: The app does not offer personalized workout plans or long-term progression tracking, limiting its effectiveness for users seeking tailored fitness programs.
- Overwhelming Choices: The vast array of workouts and programs can be overwhelming, especially for beginners, making it challenging to select the most suitable routine.
- Limited Progress Tracking: The app's progress tracking features are underdeveloped, lacking detailed metrics and analytics that many users desire to monitor their fitness journey effectively.
- Customization Limitations: Users face restrictions in customizing workouts, such as adding drop sets or supersets, which may hinder those seeking more advanced training modifications.
- Occasional Technical Issues: Some users have reported technical problems, such as the app signing them out unexpectedly and losing saved workouts, affecting the overall user experience.

- Inadequate Exercise Library: While extensive, the exercise library may not cover all exercises users wish to include, and there's no current option to add personal exercises with custom media.
- Insufficient Progress Visualization: The app lacks comprehensive progress visualization tools, such as graphs or streak trackers, which could enhance motivation and goal setting for users.

Description:
Strong is a popular fitness application designed to help users track their workouts, monitor progress, and achieve their fitness goals. It offers a user-friendly interface that allows individuals to log exercises, sets, and repetitions efficiently.

Key Features:
- Exercise Logging: Users can quickly input exercises, sets, reps, and weights, making workout tracking straightforward.
- Customisable Routines: The app enables the creation and saving of personalized workout routines for easy access during future sessions.
- Progress Tracking: Visual representations of progress over time, such as graphs and statistics, help users stay motivated and informed about their improvements.
- Exercise Library: A comprehensive library with descriptions and animations assists users in performing exercises correctly.

Advantages:
- User-Friendly Design: The intuitive interface ensures that users spend less time navigating the app and more time focusing on their workouts.
- Comprehensive Tracking: Detailed logs and progress visuals provide valuable insights into one's fitness journey.
- Customization: The ability to tailor workouts caters to individual preferences and goals.

Disadvantages:
- Limited Free Features: Some advanced functionalities may require a subscription, which could be a barrier for users seeking free options.
- No Guided Workouts: Unlike some competitors, Strong does not offer guided workout sessions or instructional videos within the app.