Work Breakdown Structure (WBS):

1.0. Project Planning and Management		
I.D	Task	Description
1.1	Define Team.	Assign roles, start communication channels and create initial resources.
1.2	Project Start Meeting.	Establish the scope, objectives and WBS.
1.3	Set up the Agile Infrastructure.	Initialise the task tracker (Trello) and version control (GitHub).
1.4	Sprint and Review Planning.	Hold bi-weekly sprint meetings and weekly team review meetings.
1.5	Risk Management.	Identify potential risks and fixes.

2.0. Requirements Analysis		
I.D	Task	Description
2.1	Functional Requirements.	Define the specific features, such as exercise inputs and calorie calculator.
2.2	Non-Functional Requirements.	Define performance, security, scalability and accessibility.
2.3	AI and Data Requirements.	Specify the data points and required outputs for the AI.
2.4	Android Specifications.	Detail the device compatibility requirements.

3.0. Design and Architecture		
I.D	Task	Description
3.1	UX/UI Wireframing.	Create the low-fidelity wireframes.
3.2	Initial Testing.	Test the low-fidelity wireframes with volunteers.
3.3	Visual Design.	Develop the high-fidelity design.
3.4	Database Schema Design.	Design the data structure for the user profiles, meals, exercises and fitness plans.
3.5	Main Application Architecture.	Define the Android development stack and overall structure.

4.0. Development and Implementation		
I.D	Task	Description
4.1	Environment Setup.	Create the backend, CI/CD pipelines and development environment.
4.2	User Profiles and Authentication.	Build a secure registration, login and profile management.
4.3	Tracking Development Feature.	Implement all user input and calculation features.
	Exercise and Nutritional Tracker.	Build the screens for logging workouts and meals.
	Calorie and Data Calculation.	Implement the backend logic for the calories trackers.
4.4	AI Development.	Integrate the AI features.
	AI Fitness Plan.	Create a feature that takes the user data and makes a fitness plan accordingly.
	AI Chatbot Development.	Build the AI Chatbot for fitness queries.
4.5	Data Analytics Dashboard.	Implement the dashboard that shows user progress.
4.6	Notifications and Reminders.	Create notifications for fitness reminders and motivation notifications.

5.0. Testing		
I.D	Task	Description
5.1	Integration Testing.	Test backend and frontend integration, including individual functionality.
5.2	Feature Testing.	Confirm all main features meet the defined requirements.
5.3	Performance Testing.	Test the app’s response to a heavy data load.
5.4	User Testing.	Have target users test the final app and gather feedback.
5.6	Bug Fixes.	Address all identified bugs from the testing.

Resource Plan:

1.0. Project Planning and Management			
I.D	Human Resource	Financial Resource	Equipment Resource
1.1	Team Members	N/A	Outlook Emails
1.2	Team Members	N/A	Discord
1.3	Information Manager	N/A	Trello, Github
1.4	Team Members	N/A	Discord, Google Docs
1.5	Risk Manager	N/A	Google Docs

2.0. Requirement Analysis			
I.D	Human Resource	Financial Resource	Equipment Resource
2.1	UX/UI Designer Other Team Members	N/A	Google Docs
2.2	Team Members	N/A	Google Docs
2.3	Backend Designer AI Designer	N/A	Google Docs
2.4	Team Members	N/A	Google Docs

3.0. Design and Architecture			
I.D	Human Resource	Financial Resource	Equipment Resource
3.1	UX/UI Designer	N/A	OneNote
3.2	Volunteers	N/A	Forms
3.3	UX/UI Designer	N/A	Android Studio
3.4	Backend Designer AI Designer	N/A	Database Modeling
3.5	Team Members	N/A	Android Studio

4.0. Development and Implementation			
I.D	Human Resource	Financial Resource	Equipment Resource

4.1	Backend Designer	N/A	Android Studio
4.2	Security Designer	N/A	Android Studio, Firebase
4.3	Team Members	N/A	Android Studio, Backend Frameworks
4.4	Team Members	N/A	Android Studio, API
4.5	Team Members	N/A	Android Studio
4.6	Team Members	N/A	Android Studio, Firebase

5.0. Testing			
I.D	Human Resource	Financial Resource	Equipment Resource
5.1	Team Members	N/A	Completed App, Android Studio
5.2	Team Members	N/A	Completed app, Android Studio, Test Scenarios
5.3	Team Members	N/A	Load Testing Tool
5.4	Team Members, Volunteers	N/A	Completed App

Quality Management:

The quality standards will be assessed under four main components. Firstly, the functional accuracy of the project must be completely accurate. All of the important features, such as the calorie counters, must be completely accurate and not output errors. Secondly, the app must be reliable and fast. The screens should load quickly and never crash, even with multiple users. Thirdly, the app should be accessible to a range of users; no matter what background they have. The app should be inclusive and have settings that help it become accessible. Finally, data must be handled robustly and securely. The app must keep all data private and safe; this will be achieved through testing with any major weaknesses being attended to immediately.

Quality checks will take place in our weekly meetings. In these meetings, everyone will talk about what they have completed and it will give time for other team members to check or question any quality issues with the completed task. The developer will run a quality test on the small part of code first to check it works on its own. Once checked, the code will be integrated into the main part of code and tested to see if it still works.

Once the code is completed, two more tests will take place for the quality checks. The code will undergo a speed/load test; this test will check how the app handles multiple users and whether it will affect the speed and use of the app overall. The final test will be the real-world test, where people outside of the team will use the app and fill in a questionnaire pointing out any major changes or bugs that need to be fixed before launch.

Monitoring and Evaluation:

The team keeps the project on track by holding multiple different types of meetings. Weekly meetings are held for role assignments, quick overview of progress and any questions that may need to be answered. Every two weeks, a sprint meeting is held to review the working new features of the app. This allows for the necessary feedback to be given for quick fixes and recovery.

The team will try their best to follow the sprints and timeline in order to hit the project's main targets at the correct time. By completing these targets at the correct time, the app will be completed in time for the deadline without issue. Having specific target deadlines makes the developers work harder to hit goals and allows for extensions if necessary with plenty of time before the overall deadline. All team members will evaluate their own code as well as having another team member evaluate their code too, as fresh eyes can sometimes spot things others cannot. These evaluations will be kept up until the end of the project.