



HIPPODROME

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
AWARD OF THE DEGREE OF

B.TECH. (COMPUTER ENGINEERING)

TO

RK UNIVERSITY, RAJKOT

SUBMITTED BY

Name of Student
Parmar Parth H.

Enrollment No.
20SOECE13022

UNDER THE GUIDANCE OF

Internal Guide

(Assistant) Prof. Bhoomi Dangar
RK University Rajkot

External Guide
Mr. Alpesh Ratho
CEO/ MD
Codizious Technologies
Rajkot, Gujarat.

April 2023



SCHOOL OF ENGINEERING, RK UNIVERSITY, RAJKOT

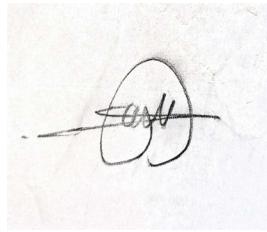
DECLARATION

I hereby certify that I am the sole author of this project work and that neither any part of this project work nor the whole of the project work has been submitted for a degree to any other University or Institution. I certify that, to the best of my knowledge, my project work does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my project document, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I declare that this is a true copy of my project work, including any final revisions, as approved by my project review committee.

Parmar Parth H. (20SOECE13022)

Date: 01/06/2023

Place: Rajkot



Signature: _____



CERTIFICATE

OF EXPERIENCE



PROUDLY PRESENTED TO

Parth H Parmar

We're delighted to confirm your successful completion of a six month Android Developer internship at Codizious Technologies. During your time with us, you've honed skills in Java, Kotlin, and Android SDK, and have contributed to multiple app development projects. Your hands-on experience in UI/UX design and API integrations has been noteworthy.

Mr. Alpesh Rathod
For, Codizious Technologies.
HR Manager



📞 +91 81605 19719
✉️ info@codizious.com
📍 609, Shivalay Complex, Mavdi Circle, Nr.
Ashirwad Hospital, Mavdi Main Rd,
150ft. Ring Rd, Rajkot - 360004

CERTIFICATE

This is to certify that the work which is being presented in the Project Report entitled "**HIPPODROME**", in partial fulfillment of the requirement for the award of the degree of **B.Tech. (Computer Engineering)** and submitted to the School of Engineering, RK University, is an authentic record of my/our own work carried out during a period from **January 2022 to June 2023**.

The matter presented in this Project Report has not been submitted by me/us for the award of any other degree elsewhere.

Student Name: Parmar Parth H. (20SOECE13022)

This is to certify that the above statement made by the student is correct to the best of my knowledge.

Internal Guide	External Guid	Head of Department
Assistant) Prof. Bhoomi Dangar School of Engineering,RK University , Rajkot	Mr. Alpesh Rathod CEO/ MD Codizious Technologies, Rajkot, Gujarat	Dr. Nirav Bhatt CE / IT / BCA / MCA School of Engineering , RK University, Rajkot



SCHOOL OF ENGINEERING, RK UNIVERSITY, RAJKOT

ACKNOWLEDGEMENT

Dear (Assistant) Prof. Bhoomi Dangar and Mr. Alpesh Rathod (CEO/ MD),

I am writing this acknowledgement to express my sincere gratitude for the invaluable guidance and support you have provided me during my internship with Android application development. Your insights, expertise, and willingness to share your knowledge have been instrumental in shaping my understanding of the industry and refining my skills.

To (Assistant) Prof. Bhoomi Dangar, I am grateful for the opportunities you provided me with to learn and grow, both academically and professionally. You were always available to answer my questions, and your feedback has been invaluable in helping me to improve my work. I appreciate the patience, understanding, and encouragement that you have provided me during my time of 8th Semester.

To Mr. Alpesh Rathod (CEO/ MD), I am grateful for the opportunity to have worked with such a dedicated and hardworking team. Your insights and knowledge of the industry have been invaluable in helping me to understand the company's operations and objectives. I appreciate the time you took to mentor me and answer my questions, and I am grateful for the feedback you provided me with throughout the internship.

I would also like to extend my gratitude to the rest of the team at Codizious for their support and collaborative work environment. The positive culture at the company has made me feel welcome and encouraged me to contribute to the team.

Lastly, I would like to thank all the people who have contributed to the success of Hippodrome Application. I feel fortunate to have had the opportunity to work with such a dynamic and forward-thinking team, and I am grateful for the experience.

Once again, thank you for your invaluable guidance and mentorship during my internship. I am confident that the skills and knowledge I have acquired during my time with the school and the company will serve me well in my future endeavors.

Sincerely, Parth

Parmar H.

Abstract/synopsis

Health and fitness tracking apps are mobile applications that help users to track various aspects of their health and fitness, such as activity, food intake, sleep, and biometric data. These apps allow users to input and store this data, providing them with insights into their health and fitness patterns. They can also set and track progress towards personal health and fitness goals, such as weight loss or muscle gain. The apps generate reports and analytics on health data, to help users understand their progress and identify areas for improvement. They also allow users to integrate with other devices and apps to provide a more comprehensive view of their health and fitness. Health and fitness tracking apps typically have user-friendly interfaces and accessibility options to make the app easy to use for all users. They also allow users to update their personal information, such as their profile and goals. Health and fitness tracking apps can be an effective tool for individuals looking to improve their health and fitness, by providing them with the necessary data and insights to make informed decisions and set achievable goals.

List Of Figures

Fig. – 4.1 Covid related results

Fig. – 4.1.1 Fitness App. Results

Fig. – 4.6 Use case Diagram

Fig. – 4.7.2 E-R Diagram

Fig. – 4.7.3 Class Diagram

Fig. – 4.7.4 System Activity

Fig. – 4.7.5 Object Diagram

Fig. – 4.8.1 DFD 0-1 Diagram

Fig. – 5.2.1 Flow chart Diagram

Fig. – 5.3.1 Interface Design

Fig. – 8.0 Screen Shots

List Of Tables

1. User Profile
2. Workout Log
3. Exercise Library
4. Meal Tracker
5. Water Intake Tracker
6. Weight Tracker
7. Body Measurements Tracker
8. Progress Graphs/Charts
9. Fitness Challenges
10. Personalized Workouts
11. Virtual Coach
12. Community Forum
13. Goal Setting
14. Calorie Counter
15. Macro Tracker
16. Heart Rate Monitor
17. Sleep Tracker
18. Meditation Timer
19. Reminder Notifications
20. Health Tips and Articles.

Table of Content

CANDIDATE'S DECLARATION	I
INDUSTRIAL CERTIFICATE	II
INSTITUTE CERTIFICATE	III
ACKNOWLEDGEMENT	IV
ABSTRACT	V
LIST OF FIGURES	VI
LIST OF TABLES	VII
TABLE OF CONTENT	VIII-X
COMPANY PROFILE	XI
CHAPTER 1	
1.1 Project Summary	01
1.2 Purpose	01
1.3 Project Scope	02
1.4 Technology and Literature Review of Past Work/System	03
CHAPTER 2	
2.1 Project Planning and Scheduling	
2.1.1 Project Development Approach and Justification	05
2.1.2 Project Plan	06
2.1.3 Schedule Representation	07
2.2 Risk Management	
2.2.1 Risk Identification	08

2.2.2 Risk Analysis	09
2.2.3 Risk Planning	10
CHAPTER 3	
3.1 User Characteristics	11
3.2 Hardware and Software Requirements	11
3.3 Constraints	12
CHAPTER 4	
4.1 Study of Current System	14
4.2 Problem and Weaknesses of Current System	16
4.3 Requirements of New System	17
4.4 Feasibility Study	18
4.5 Requirement Validation	19
4.6 Function of System	
4.6.1 Use Case Diagram	20
4.7 Data Modeling	
4.7.1 Data Dictionary	21
4.7.2 E-R Diagram	22
4.7.3 Class Diagram	23
4.7.4 System Activity	24
4.7.5 Object Interaction Diagram	25
4.8 Functional and Behavioral Modeling	
4.8.1 Data Flow Diagram (0 and 1 level)	25
4.8.2 Process Specification and Decision Table	26
4.9 Main Modules of New System	27
4.10 Selection of Hardware and Software and Justification	28

CHAPTER 5	
5.1 System Design	
5.1.1 Mapping objects/classes to tables	29
5.1.2 Tables and Relationship	30
5.1.3 Logical Description of Data	31
5.2 System Procedural Design	
5.2.1 Flow chart or activity design	32
5.3 Input/Output and Interface Design	
5.3.1 Samples of Forms and Interface	33
5.3.2 Access Control and Security	34
CHAPTER 6	
6.1 Implementation Environment	36
6.2 Program/Modules Specification	36
6.3 Security Features	38
6.4 Coding Standards	39
6.5 Sample Coding	40
CHAPTER 7	
7.1 Testing Plan	43
7.2 Testing Strategy	44
7.3 Testing Methods	45
7.4 Test Cases	46
CHAPTER 8	47
CHAPTER 9	49
CHAPTER 10	51
REFERENCES	52

Company Profile

WE ARE DEVELOPING INCREDIBLE **WEB** AND **MOBILE SOLUTIONS**.

OUR CONSTANT STRIVE FOR PERFECTION ENSURES GREAT END-RESULTS.

WE BELIEVE IN PERFECTION AND THAT'S THE FUNDAMENTAL RULE WE PASS TO EVERY TEAM MEMBER WHO JOINS A PROJECT AT **CODIZIOUS TECHNOLOGIES**.



CHAPTERS

1.0 Introduction

1.1 Project Summary

A health and fitness app is a software application that aims to help users track and improve their health and fitness. The app can track things like exercise, calories consumed and burned, weight, and sleep. It can also provide users with information and resources to help them achieve their health and fitness goals. The app will be available on mobile devices and will be user-friendly. The app will have a variety of features such as personalized workout plans, meal plans, and reminders to encourage regular physical activity. It will also allow users to connect with others who have similar health and fitness goals, such as through social networks or support groups. It will integrate with wearables like fitness trackers to provide more detailed and accurate data about users' physical activity. The app will help users monitor and track their sleep patterns, diet, water intake and medication schedule. It will also offer features like tracking of menstrual cycle for women. The app will provide progress tracking and analytics to help users see how they are doing over time and adjust as needed. The goal of the project is to develop an app that will help people maintain a healthy lifestyle, lose weight, and improve their overall fitness level.

1.2 Purpose: Goals & Objectives

The goals and objectives of a health and fitness app can vary depending on the specific app and the intended audience. Some common goals and objectives include:

- Helping users track and monitor their physical activity, including steps taken, calories burned, and distance travelled.
- Providing users with information and resources to help them set and achieve their health and fitness goals, such as losing weight, gaining muscle, or improving overall fitness.

- Offering personalized workout plans, meal plans, and other guidance to help users make healthy lifestyle choices.
- Encouraging users to engage in regular physical activity by providing reminders, notifications, and other incentives.
- Allowing users to connect with others who have similar health and fitness goals, such as through social networks or support groups.
- Integrating with wearables like fitness trackers to provide more detailed and accurate data about users' physical activity.
- Helping users monitor and track their sleep patterns, diet and water intake.
- Providing users with the ability to set reminders for medication and track their medication schedule.
- Offering features like tracking of menstrual cycle for women.
- Offering progress tracking and analytics to help users see how they are doing over time and adjust as needed.

The *objective* of our health and fitness tracking app is to provide users with a comprehensive and user-friendly platform for monitoring and improving their overall health and fitness. The app will enable users to track a wide range of health-related metrics, including physical activity, food intake and biometric data. Additionally, the app will provide users with goal setting and progress-tracking features using graph, as well as personalized insights based on their data. The goal is to empower users to make informed and data-driven decisions about their health and fitness, and to support them in achieving their personal health and fitness goals all for free.

1.3 Scope

The scope of a health and fitness app can vary depending on the specific app and the intended audience. However, some common elements that may be included in the scope of a health and fitness app include:

- Tracking and monitoring of physical activity: The app will allow users to track and monitor their physical activity, including steps taken, calories burned, and distance travelled.
- Personalized workout plans and meal plans: The app will provide users with personalized workout plans and meal plans to help them achieve their health and fitness goals.
- Reminders and notifications: The app will include reminders and notifications to encourage regular physical activity and healthy lifestyle choices.
- Social network and support groups: The app will allow users to connect with others who have similar health and fitness goals, such as through social networks or support groups.
- Progress tracking and analytics: The app will provide progress tracking and analytics to help users see how they are doing over time and adjust as needed.[SEP]
- User-friendly interface: The app will be designed with a user-friendly interface that is easy to navigate.
- Mobile-friendly: The app will be available for mobile devices and will be compatible with Android platforms.
- Data security: The app will have robust data security measures in place to protect users' personal information and data.

1.4 Technology and Literature Review of Past Work/System

There are many health and fitness apps currently available on the market, which use a variety of technologies to help users track and improve their health and fitness. Some common technologies used in these apps include:

- GPS tracking: Many apps use GPS tracking to allow users to track their outdoor workouts, such as runs and bike rides.
- Mobile sensors: Many apps use the sensors built into smartphones, such as the accelerometer, to track users' physical activity and provide them with more accurate data.
- Wearable integration: Some apps integrate with wearables like fitness trackers and smartwatches to provide users with more detailed data about their physical activity.
- Social networks: Some apps allow users to connect with others who have similar health and fitness goals, such as through social networks or support groups.
- Machine Learning: Some apps use machine learning algorithms to personalize workout plans and meal plans for each user based on their fitness level, goals, and preferences.
- Cloud-based storage: Many apps use cloud-based storage to allow users to access their data from any device and to store large amounts of data.

Literature review studies, have shown that health and fitness apps can be effective in helping users achieve their health and fitness goals, and can also help people to make healthier lifestyle choices. Studies have also found that apps that use social networks and support groups can be especially effective in promoting engagement and adherence to healthy lifestyle changes. Additionally, apps that integrate with wearables and use machine learning algorithms, have been found to provide more accurate data and to be better at personalizing workout plans and meal plans for each user.

Some examples of popular health and fitness apps include MyFitnessPal, Fitbit, and Edmundo. These apps have been widely used and have received positive feedback from users for their ability to help users track and improve their health and fitness.

However, it's important to note that not all health and fitness apps are created equal, and some may not be as effective or user-friendly as others. It's recommended to research different options and read reviews before deciding on an app to use.

2.0 Project Management

2.1 Project Planning and scheduling

2.1.1 Project Development Approach (Process Paradigm) and Justification

The process paradigm, or development approach, used for a health and fitness app will depend on the specific needs and requirements of the project. However, there are several common process paradigms that are often used for software development projects, including:

- Waterfall: The Waterfall model is a linear, sequential approach to software development. It involves a strict, step-by-step process, where each phase of the project must be completed before moving on to the next. This model is best suited for projects with well-defined and fixed requirements, and where the final product can be clearly defined at the beginning of the project.
- Agile: The Agile model is a flexible, iterative approach to software development. It involves regularly releasing small, working versions of the app, and then incorporating feedback and making changes as needed. This model is best suited for projects with constantly evolving requirements, and where the final product is not clearly defined at the beginning of the project.
- Scrum: Scrum is an Agile development framework that is often used for software development projects. It involves a product owner, development team, and scrum master working together to deliver a working product incrementally and in short cycles (sprints).

The Agile or Scrum development approach is most suitable for the health and fitness app development, as it allows for flexibility and constant adaptation to changing requirements

and feedback from users. This approach also allows for regular releases of the app, which can be tested and refined based on user feedback.

Additionally, Agile and Scrum methodologies focus on collaboration and communication between team members and stakeholders, which is essential for a successful health and fitness app development. It also allows for a more efficient use of resources, and it is more suited to software development projects that are complex and subject to change.

In summary, Agile or Scrum development approach is the best choice for the health and fitness app development, due to its flexibility and adaptability, user-centred approach, and focus on collaboration and communication between team members and stakeholders.

2.1.2 Project Plan including Milestones, Deliverables, Roles,

Responsibilities and Dependencies

A project plan for a health and fitness app should include milestones, deliverables, roles, responsibilities, and dependencies.

Milestones:

- Project initiation: Define project objectives, scope, and requirements, and create a project charter.
- App design and prototyping: Design the layout and functionality of the app and create a working prototype.
- Development and testing: Develop and test the app, incorporating any necessary changes or updates.
- Beta testing: Release a beta version of the app for testing and feedback from a select group of users.
- App launch: Launch the app in the app store.
- Post-launch support: Provide ongoing support and maintenance for the app.

Deliverables:

- Project charter

- App prototype
- Final version of the app
- User manual
- Technical documentation
- Test cases and test results

Roles and Responsibilities:

- Project manager: Responsible for overall project management, including project planning, scheduling, and monitoring.
- App designer: Responsible for designing the layout and functionality of the app.
- Developers: Responsible for developing and testing the app.
- Quality assurance: Responsible for testing the app and ensuring that it meets the required quality standards.
- Beta testers: Responsible for testing the beta version of the app and providing feedback.
- Product owner: Responsible for defining and prioritizing the app features and user stories

Dependencies:

- App design and prototyping is dependent on the project initiation phase.
- Development and testing is dependent on the app design and prototyping phase.
- Beta testing is dependent on the development and testing phase.
- App launch is dependent on the beta testing phase.
- Post-launch support is dependent on the app launch phase.

2.1.3 Schedule Representation

A schedule representation for a health and fitness app can take many forms, but some common examples include:

- Gantt Chart: A Gantt chart is a type of bar chart that represents the schedule of a project. It shows the start and end dates of each task, as well as the dependencies between tasks. Gantt charts can be used to visualize the project schedule, and to identify any potential schedule conflicts or delays.
- Network Diagram: A network diagram is a visual representation of the project schedule that shows the dependencies between tasks. It uses arrows to show the relationship between tasks and can be used to identify the critical path of the project (the sequence of tasks that determine the project's completion date).
- PERT Chart: A PERT chart is like a network diagram, but it also includes the expected duration of each task. This can be useful for identifying tasks that are taking longer than expected, or for identifying tasks that are at risk of delaying the project.
- Kanban Board: A Kanban board is a visual representation of the workflow of a project. It shows the status of each task and can be used to identify bottlenecks, visualize progress, and see where resources are being spent.
- Timeline: A timeline is a simple representation of the project schedule, showing the start and end dates of each task. It can be useful for communicating the project schedule to stakeholders.

2.2 Risk Management

2.2.1 Risk Identification (it is concerned with discovering possible risk to the project)

Some common risks that may be identified for a health and fitness app include:

- Technical risks: These include risks related to the technology used to develop the app, such as compatibility issues, bugs, or security vulnerabilities.
- Design risks: These include risks related to the design of the app, such as usability issues or user experience problems.

- Legal risks: These include risks related to compliance with laws and regulations related to health and fitness apps.
- Market risks: These include risks related to the market for the app, such as lack of demand or competition from other apps.
- Resource risks: These include risks related to the availability and quality of resources, such as personnel, equipment or funding.
- Time risks: These include risks related to delays or schedule overruns, which can cause the project to be delivered late or over budget.
- Data privacy and security risks: These include risks related to the data privacy and security of the users.

2.2.2 Risk Analysis (each identified risk is considered in turn and a judgment made about the probability and the seriousness of the risk)

For a health and fitness app specifically:

- Technical risks such as compatibility issues, bugs, or security vulnerabilities would fall under the High probability and High impact category.
- Design risks such as usability issues or user experience problems might fall under the High probability and low impact category.
- Legal risks such as non-compliance with laws and regulations related to health and fitness apps would fall under the High probability and High impact category.
- Market risks such as lack of demand or competition from other apps would fall under the High probability and High impact category.
- Resource risks such as personnel, equipment or funding issues would fall under the High probability and High impact category.
- Time risks such as delays, or schedule overruns would fall under the High probability and High impact category.

- Data privacy and security risks would fall under the High probability and High impact category.

2.2.3 Risk Planning (Identify strategies to manage the risk)

For a health and fitness app specifically:

- Technical risks such as compatibility issues, bugs, or security vulnerabilities can be mitigated by conducting thorough testing and security audits, using secure coding practices and keeping the software updated.
- Design risks such as usability issues or user experience problems can be mitigated by conducting user research and usability testing during the design phase.
- Legal risks such as non-compliance with laws and regulations related to health and fitness apps can be mitigated by consulting with legal experts, keeping up to date on laws and regulations and having a compliance plan in place.
- Market risks such as lack of demand or competition from other apps can be mitigated by conducting market research, identifying target users and developing a marketing strategy.
- Resource risks such as personnel, equipment or funding issues can be mitigated by having backup resources or plans in place, or by outsourcing certain tasks to vendors.
- Time risks such as delays or schedule overruns can be mitigated by developing a detailed project schedule, managing dependencies between tasks and monitoring progress regularly.
- Data privacy and security risks can be mitigated by implementing strict data security and privacy policies, conducting regular security audits and providing data encryption.

3.0 System Requirements Study

3.1 User Characteristics (Type of users who is dealing with the system)

For a health and fitness app, some possible user characteristics are:

- Age: The app could be designed for a specific age group, such as young adults or seniors.
- Gender: The app could be designed to cater to a specific gender, such as women or men or others.
- Income: The app could be designed for people with different income levels, such as low-income or high-income individuals.
- Education level: The app could be designed for people with different levels of education, such as those who have completed high school or those who have a college degree.
- Fitness level: The app could be designed for people with different fitness levels, such as beginners or experienced athletes.
- Health conditions: The app could be designed for people with specific health conditions such as diabetes, hypertension, or arthritis.
- Occupation: The app could be designed for people with different occupations such as office workers, farmers, or construction workers.
- Geographical location: The app could be designed for people living in different geographical locations, such as urban or rural areas.

3.2 Hardware and Software Requirements (minimum requirements to run your system)

Hardware Requirements:

- Processor: A modern processor with at least 1.5 GHz clock speed.

- Memory: At least 2 GB of RAM.
- Storage: At least 1 GB of free storage space.
- Display: A display with a resolution of at least 480 x 800 pixels.
- Operating System: A recent version of iOS or Android.
- Connectivity: Internet connectivity through Wi-Fi or cellular data.

Software Requirements:

- Operating System: A recent version of iOS or Android.
- SDK: Software development kit for the chosen platform (iOS or Android).
- Programming Language: Swift (iOS) or Kotlin/java (Android)
- Database: A database management system (DBMS) such as SQLite, MySQL, or MongoDB.
- Libraries and Frameworks: Libraries and frameworks such as Retrofit, OkHttp, and RxJava to develop and manage network communication, and libraries such as Glide, Picasso or Fresco to handle images.
- Push Notification Services: Firebase Cloud Messaging (FCM) or OneSignal
- Map Services: Google Maps or Apple Maps
- Analytics: Google Analytics or Firebase Analytics

3.3 Constraints

(It includes Regulatory Policies, Hardware Limitations, Interfaces to Other Applications, Parallel Operations, Higher Order Language Requirements, Reliability Requirements, Criticality of the Application, Safety and Security Consideration, Assumptions and Dependencies or any other constraints related to your system)

- Regulatory policies: Health and fitness apps are subject to various laws and regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) in the India, which govern the handling of personal health information (PHI). The app must comply with these regulations, which can impact development and operation.
- Hardware limitations: The app must be able to run on a wide range of devices, including smartphones and tablets, with different screen sizes and resolutions. This can impact the design and functionality of the app.
- Interfaces to other applications: The app may need to integrate with other applications, such as social media platforms, wearables, or health tracking devices. This can impact development and operation.
- Parallel operations: The app may need to be able to perform multiple operations simultaneously, such as tracking fitness data and displaying workout instructions. This can impact the design and functionality of the app.
- Reliability requirements: The app must be reliable and provide accurate information. This can impact development and testing.
- Criticality of the application: The app must be critical in terms of functionality and data security. It must not compromise user's data privacy and security.
- Safety and security considerations: The app must be designed to protect user data and prevent unauthorized access. This can impact development and operation.
- Assumptions and dependencies: The app may make assumptions about the availability of certain hardware or software or depend on other applications or services. This can impact development and operation.

4.0 System Analysis

4.1 Study of Current System

A study of the current system for a health and fitness app involves analysing existing apps in the market to understand their features, functionality, user interface, and overall performance. This can help to identify gaps in the market and areas where a new app could improve upon or add value to the existing options.

COVID-19 fitness app market impact: an increase of 48.8% between 2019 and 2020

Pandemic Impact	Post COVID Outlook
The fitness app market increased by 48.8% from 2019 to 2020.	The market is estimated to witness a y-o-y growth of 17.0% to 17.6% in the next 5 years
COVID-19 has had a significant impact on the fitness app market as many startups have observed high growth in the past 7 months. For instance, HealthifyMe touched USD 1.35 million in 2019, and the Fitter app has seen a 30.0% growth in its revenue, with more than 650,000 downloads. After HealthifyMe, the GOQii app is being used the most.	The COVID-19 pandemic led to consumer transition with the adoption of virtual fitness avenues over traditional gyms. The COVID-19 pandemic is providing growth opportunities for workout apps due to lockdown restrictions, therefore, high use of fitness apps for bodyweight training is being observed worldwide.
Social distancing measures to slow down COVID-19 transmission have led to the closure of various gyms and health clubs, thereby adversely affecting the young adults and athletes, with many of them being unable to meet their fitness goals.	Virtual app vendors are focusing on introducing innovative solutions and expanding their geographical reach through various strategies, such as product launches, partnerships, approvals, and collaborations. Moreover, they are offering free premium access to users amid the pandemic and support in maintaining their health and fitness at home.

Fig ~ 4.1 Covid related result

Fitness App Market Report Scope

Report Attribute	Details
Market size value in 2022	USD 1.3 billion
Revenue forecast in 2030	USD 4.8 billion
Growth Rate	CAGR of 17.6% from 2022 to 2030
Base year for estimation	2021
Historical data	2016 - 2020
Forecast period	2022 - 2030
Quantitative units	Revenue in USD million and CAGR from 2022 to 2030
Report coverage	Revenue forecast, company ranking, competitive landscape, growth factors, and trends
Segments covered	Type, platform, device, region
Regional scope	North America; Europe; Asia Pacific; Latin America; MEA
Country scope	U.S.; Canada; U.K.; Germany; France; Spain; Italy; Japan; China; Australia; India; Brazil; Mexico; South Africa
Key companies profiled	Adidas; Appster; Fitbit, Inc.; FitnessKeeper; Azumio, Inc.; MyFitnessPal Inc.; Nike; Noom; Under Armour, Inc.; Appliko; Aaptiv; Appinventiv
Customization scope	Free report customization (equivalent up to 8 analysts working days) with purchase. Addition or alteration to country, regional & segment scope.
Pricing and purchase options	Avail customized purchase options to meet your exact research needs. Explore purchase options

Fig ~ 4.1.1 Fitness app result

Overall, conducting a study of the current system for a health and fitness app can provide valuable insights into the market and help to inform the development of a new app that is well-suited to the needs of its target audience.

4.2 Problem and Weaknesses of Current System

The problem and weaknesses of the current system for health and fitness apps can vary depending on the app being analysed. However, some common issues that may be identified through a study of existing systems include:

- Limited customization options: Many existing apps may have pre-set workout plans or tracking options, which may not be suitable for all users.
- Inaccurate tracking: Many apps use sensors in smartphones to track activity and movements, which can lead to inaccurate data.
- Lack of integration: Some apps may not integrate with other devices or apps, such as wearables or food tracking apps, which can make it difficult for users to get a complete picture of their health and fitness.
- Limited social features: Some apps may not have social features, such as the ability to share progress or compete with friends, which can make it difficult for users to stay motivated.
- Complex user interface: Some apps may have a complex user interface, which can make it difficult for users to navigate or understand how to use the app.
- Lack of personalization: Some apps may not consider individual users' needs and preferences, such as fitness level, injuries, or medical conditions.
- Inadequate Data Security: Most of the health and fitness apps are not sufficient to keep the user's data secure which leads to the data breaches, hacking and leaking of personal information.
- Limited reach: Some apps may not be available on all platforms, such as iOS and Android, which can limit the potential user base.

4.3 Requirements of New System (Proposed System)

(Mention all functional and non-functional including user and system requirements)

Functional Requirements:

- Customizable workout plans: Users should be able to create their own workout plans based on their fitness level, goals, and preferences.
- Accurate tracking: The app should use accurate tracking methods, such as wearables or manual input, to ensure that data is as accurate as possible.
- Integration with other devices and apps: The app should integrate with other devices, such as wearables or food tracking apps, to provide a comprehensive view of users' health and fitness.
- Social features: The app should include social features, such as the ability to share progress or compete with friends, to help users stay motivated.
- User-friendly interface: The app should have a user-friendly interface that is easy to navigate and understand.
- Personalization: The app should consider individual users' needs and preferences, such as fitness level, injuries, or medical conditions.
- Data Security: The app should have robust security features to protect the user's personal and health information from data breaches and hacking.

Non-Functional Requirements:

- Performance: The app should have a fast response time and be able to handle a high number of users.
- Scalability: The app should be able to handle an increase in users and data.
- Reliability: The app should be reliable and have minimal downtime.
- Usability: The app should be easy to use and understand for users of all ages and fitness levels.

- Accessibility: The app should be accessible to users with disabilities, such as large text and easy-to-follow instructions.
- Compatibility: The app should be compatible with multiple platforms, such as iOS and Android.
- Security: The app should have robust security features to protect user's data and personal information.

4.4 Feasibility Study

(In this section, does feasibility analysis by finding answers of the questions

like Does the system contribute to the overall objectives of the organization?

Can the system be implemented using the current technology and within the given cost and schedule constraints? Can the system be integrated with other systems which are already in place? etc. ;)

- Technical feasibility: Can the proposed system be implemented using current technology and within the given cost and schedule constraints?
- Operational feasibility: Will the proposed system support the overall objectives of the organization and contribute to the business goals?
- Economic feasibility: Will the proposed system provide a positive return on investment? Will the benefits of the system outweigh the costs?
- Schedule feasibility: Can the proposed system be developed and implemented within the given schedule?
- Legal and regulatory feasibility: Does the proposed system comply with all relevant laws and regulations?
- Integration feasibility: Can the proposed system be integrated with other existing systems and platforms?

- Maintenance and Support Feasibility: Will the proposed system be maintainable and supportable with the current resources?
- Security Feasibility: Will the proposed system be able to meet the security requirements and standards

4.5 Requirements Validation (is concerned with showing that the requirements define the system which the customer wants)

- User acceptance testing: User acceptance testing (UAT) is a testing process in which end-users are involved in evaluating the system to ensure it meets their needs and expectations.
- Inspection: Requirements inspection is a formal process where a team of experts and stakeholders review the requirements to ensure they are complete, consistent, and accurate.
- Prototyping: Creating a prototype of the proposed system allows stakeholders and end-users to see and interact with the system, providing valuable feedback on the requirements.
- Walkthrough: A walkthrough is a meeting where stakeholders and developers review the requirements together, discussing any questions or concerns that arise.
- Quality Assurance: Quality assurance (QA) is a process that ensures the developed product meets the requirements and expectations of the customer.

4.6 Functions of System

4.6.1 Use Cases, event trace or scenario (Use Case Diagram)

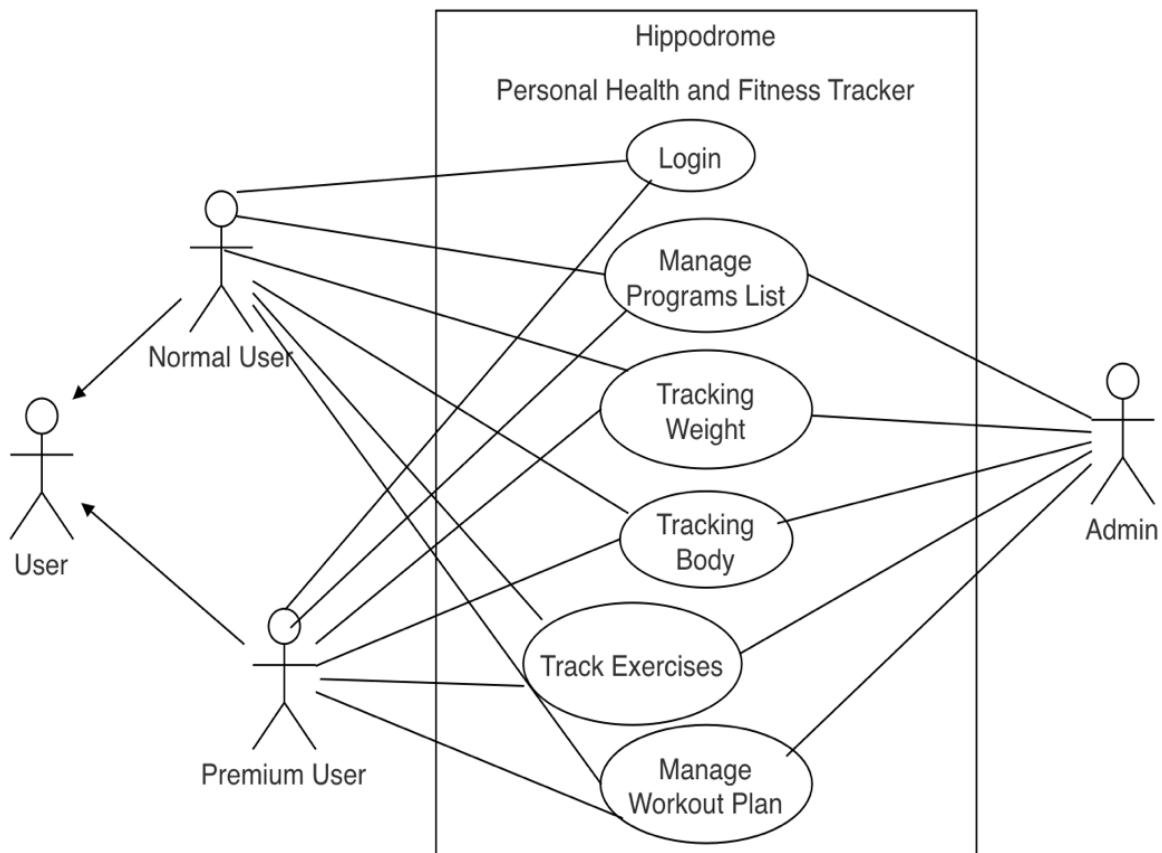


Fig ~ 4.6 Use case diagram

4.7 Data Modeling

4.7.1 Data Dictionary

A data dictionary for a health and fitness app could include the following information:

- 1) User Profile:
 - a) User ID: unique identifier for each user
 - b) Age: user's age
 - c) Gender: user's gender
 - d) Height: user's height
 - e) Weight: user's weight
 - f) Goal: user's fitness goal (e.g. weight loss, muscle gain, etc.)
- 2) Exercise Data:
 - a) Exercise ID: unique identifier for each exercise
 - b) Name: name of the exercise
 - c) Type: type of exercise (e.g. cardio, strength training, etc.)
 - d) Duration: duration of the exercise in minutes
 - e) Caloric burn: number of calories burned during the exercise
 - f) Sets/Reps: sets and reps for strength training exercises
 - g) Distance: distance for cardio exercises
- 3) Nutrition Data:
 - a) Food ID: unique identifier for each food item
 - b) Name: name of the food item
 - c) Calories: number of calories in the food item
 - d) Protein: amount of protein in the food item (grams)
 - e) Carbohydrates: amount of carbohydrates in the food item (grams)
 - f) Fat: amount of fat in the food item (grams)
 - g) Fiber: amount of fiber in the food item (grams)
 - h) Serving size: serving size of the food item
- 4) Activity Data:
 - a) Activity ID: unique identifier for each activity
 - b) Type: type of activity (e.g. steps, sleep, etc.)
 - c) Duration: duration of the activity in minutes
 - d) Date/Time: date and time of the activity
- 5) Progress Data:
 - a) Progress ID: unique identifier for each progress
 - b) User ID: User ID of the user
 - c) Weight: user's weight
 - d) Body Fat Percentage: user's body fat percentage
 - e) Measurement Date: date of the measurement
 - f) Notes: any notes or observations the user wants to add.

4.7.2 E-R diagrams

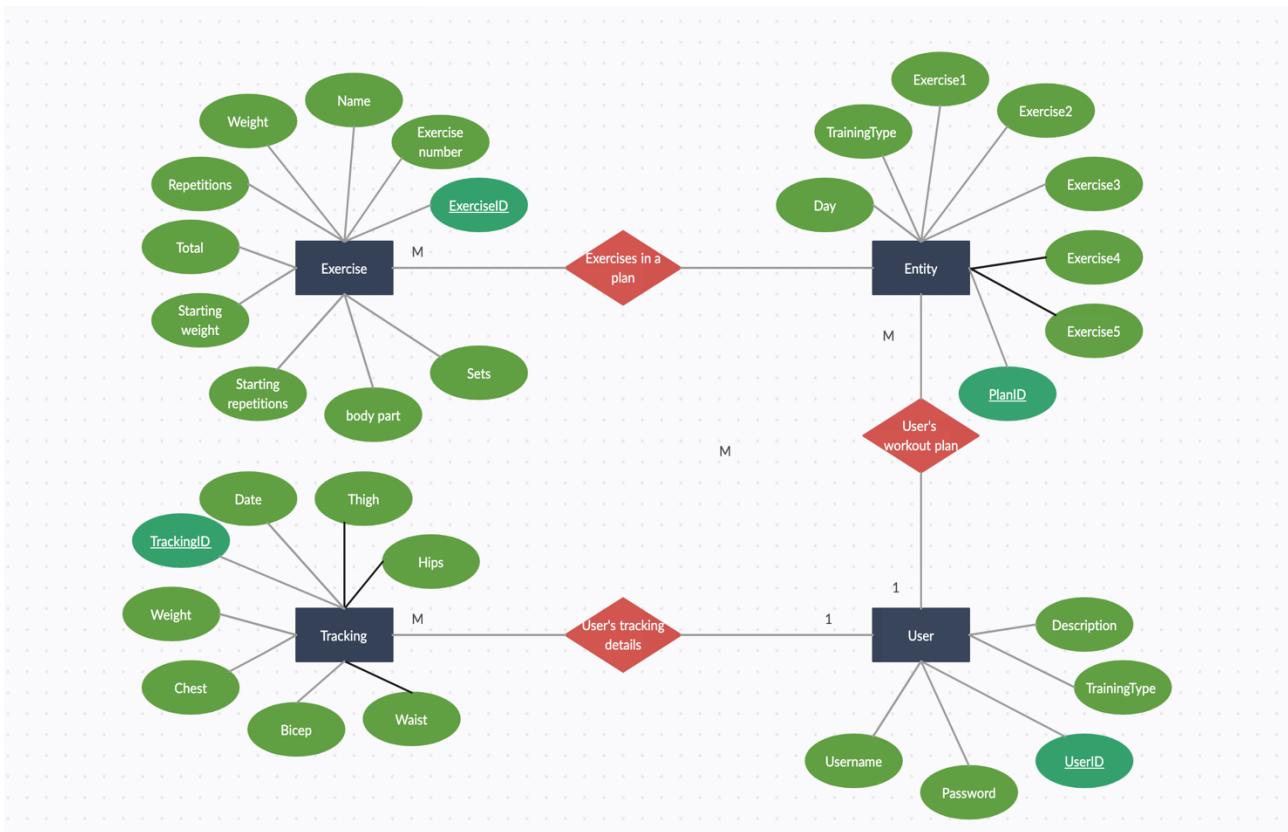


Fig ~ 4.7.2 E-R diagram

4.7.3 Class Diagram

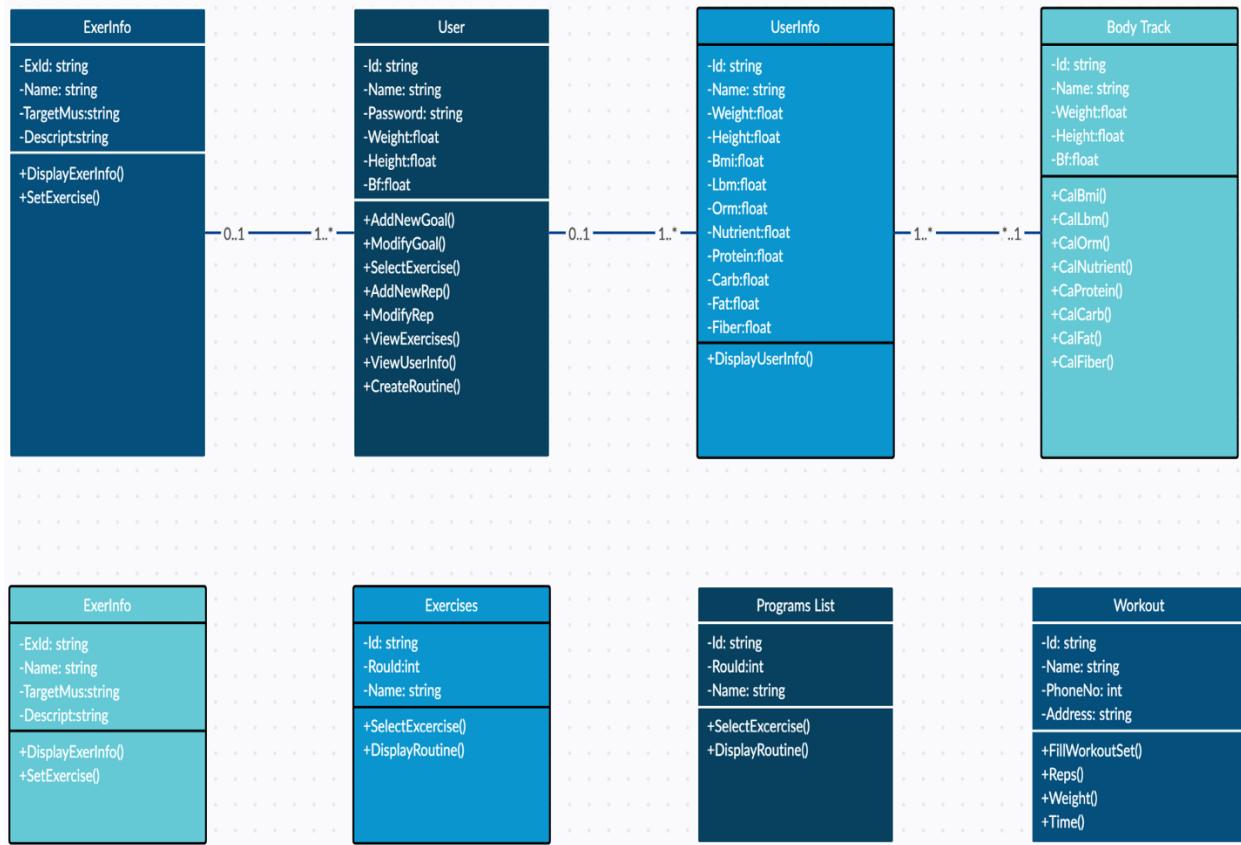


Fig ~ 4.7.3 Class diagram

4.7.4 System Activity

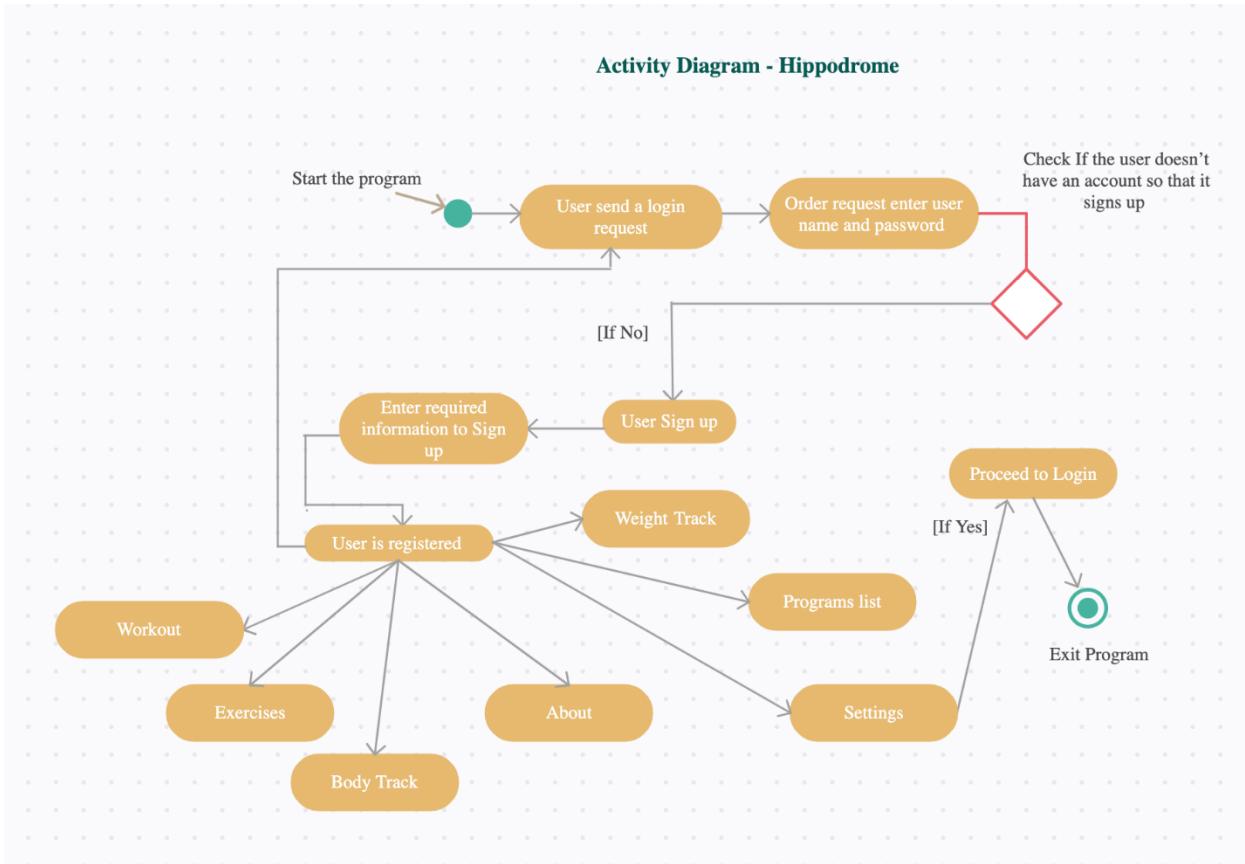


Fig ~ 4.7.4 System Activity

4.7.5 Object interaction Diagram

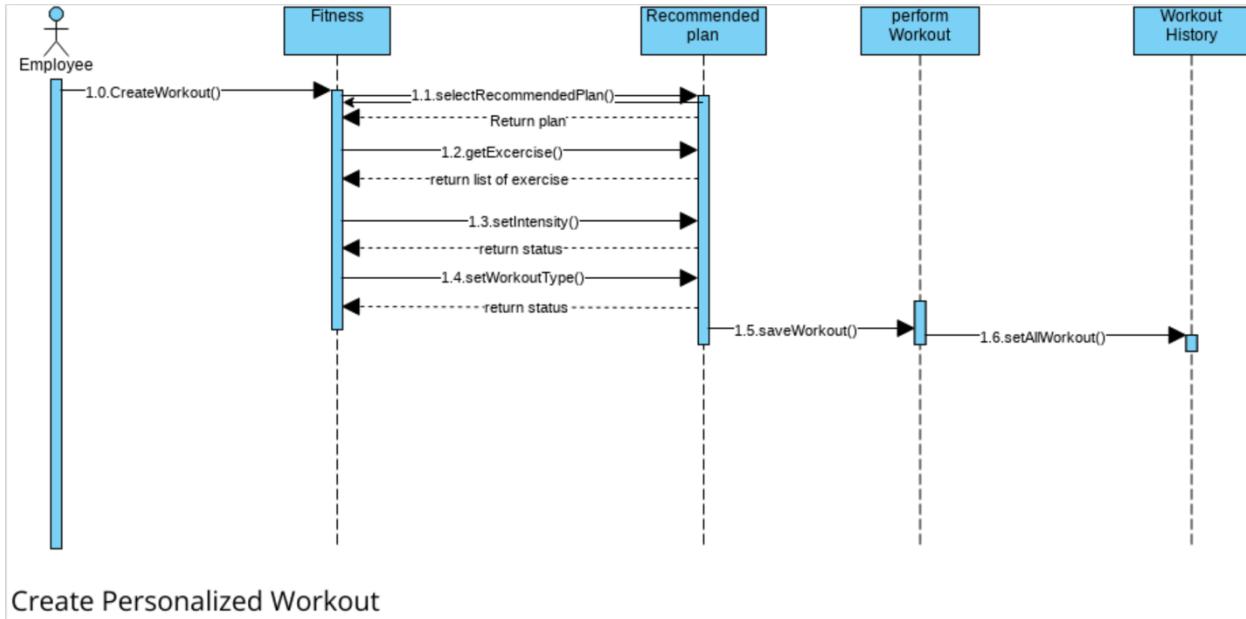
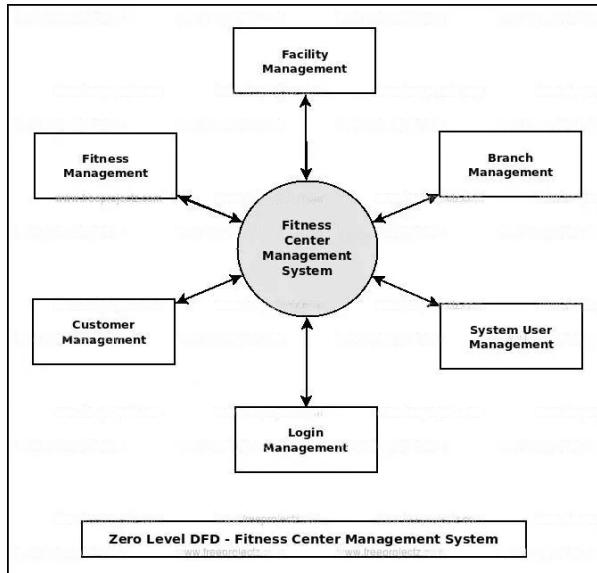


Fig ~ 4.7.5 Object Diagram

4.8 Functional and Behavioral Modeling

4.8.1 Data Flow Diagram (0 and 1 level)



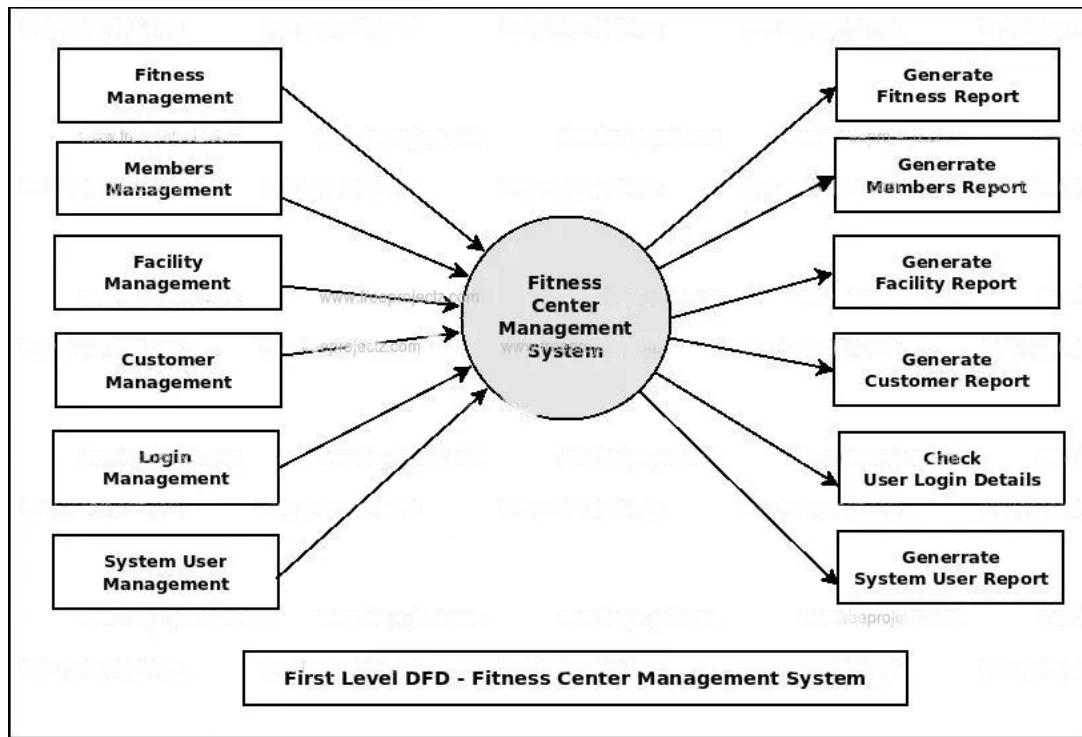


Fig ~ 4.8.1 DFD

4.8.2 Process Specification and Decision Table

Process Specification:

A process specification is a document that describes the steps, inputs, outputs, and decision points of a particular process within the health and fitness app.

- For example, the process of creating a new workout plan would have steps such as selecting the type of workout, choosing the number of days per week, and selecting the specific exercises for each day. The inputs would include the user's fitness goals and any limitations or injuries, and the output would be a personalized workout plan.

Decision Table:

A decision table is a tool used to represent the logic of a complex decision-making process in a clear and organized way.

- For example, a decision table for the process of creating a workout plan could have inputs such as the user's fitness goals, number of days per week, and any limitations or injuries. The outputs would be the specific exercises and number of sets and reps for each day, and the decision points would be based on the inputs to determine which exercises and workout plan would be most appropriate for the user.

4.9 Main Modules of New System

The main modules of a new system will vary depending on the specific requirements and goals of the project. However, some commonly included modules may include:

- User Management: This module allows users to create and manage their accounts, including login and registration, password reset, and profile management.
- Data Management: This module allows users to input, store, and retrieve data, such as customer information, inventory, sales, and financial transactions.
- Inventory Management: This module allows users to track inventory levels, receive and process orders, and manage suppliers and vendors.
- Financial Management: This module allows users to track financial transactions, manage accounts payable and receivable, and generate financial reports.
- Reporting and Analytics: This module allows users to view and analyse data, generate reports, and gain insights into the performance of the system.
- Project Management: This module allows users to plan, track, and manage projects, including tasks, timelines, and resources.
- Workflow Management: This module allows users to automate and streamline business processes, such as approvals, document management, and customer service.
- Communication and Collaboration: This module allows users to communicate and collaborate with other users, share information and files, and manage team projects.
- Security and privacy: This module ensures that the user's data is secure and private.

4.10 Selection of Hardware and Software and Justification

The selection of hardware and software for a health and fitness app is an important step in the development process, as it can have a significant impact on the overall performance and functionality of the system. The following are some factors to consider when selecting hardware and software:

- Platform compatibility: The hardware and software should be compatible with the target platform, whether it is iOS, Android, or web based.
- Hardware requirements: The hardware should be powerful enough to run the app smoothly and efficiently and should meet the minimum requirements for the chosen software.
- Software requirements: The software should be able to support the features and functionality required for the app and should be compatible with the chosen hardware.
- Cost: The cost of the hardware and software should be considered and should be within the budget.
- Security: The hardware and software should meet the security requirements and standards.
- Scalability: The hardware and software should be scalable to accommodate the future growth of the app.
- Support and maintenance: The hardware and software should be supported and maintainable with the current resources.
- User experience: The hardware and software should provide an optimal user experience.

5.0 System Design

5.1 Database Design/Data Structure Design

5.1.1 Mapping objects/classes to tables (if non-OOP languages)

In a health and fitness app, the database design and data structure design will be a crucial aspect of the system. Mapping objects or classes to tables in a non-OOP language can help to organize the data in a logical and efficient way.

For example, some of the main classes that would be mapped to tables in the database would include:

- User: This class would be mapped to the users table in the database. The properties of the class, such as name, email, password, age, gender, height, weight, etc. would be mapped to the columns of the users table. The methods of the class, such as create, read, update, delete, would be mapped to the corresponding CRUD operations that can be performed on the users table.
- Exercise: This class would be mapped to the exercises table in the database. The properties of the class, such as name, description, muscle group, equipment, etc. would be mapped to the columns of the exercises table. The methods of the class, such as create, read, update, delete, would be mapped to the corresponding CRUD operations that can be performed on the exercises table.
- Workout: This class would be mapped to the workouts table in the database. The properties of the class, such as name, description, user_id, etc. would be mapped to the columns of the workouts table. The methods of the class, such as create, read, update, delete, would be mapped to the corresponding CRUD operations that can be performed on the workouts table.
- Workout_exercise: This class would be mapped to the workout_exercise table in the database. The properties of the class, such as workout_id, exercise_id, sets, reps, weight, etc. would be mapped to the columns of the workout_exercise table. The methods of the class, such as create, read, update, delete, would be mapped to the

corresponding CRUD operations that can be performed on the workout_exercise table.

5.1.2 Tables and Relationship

•Users:

- 1.User ID (Primary Key)
- 2.Name
- 3.Email
- 4.Password
- 5.Date of Birth
- 6.Gender

•Profile:

- 1.Profile ID (Primary Key)
- 2.User ID (Foreign Key)
- 3.Height
- 4.Weight
- 5.Age
- 6.Body Mass Index (BMI)
- 7.Sex
- 8.Goal (e.g., weight loss, muscle gain)

•Goals:

- 1.Goals ID (Primary Key)
- 2.User ID (Foreign Key)
- 3.Date
- 4.Goal type (e.g., weight loss, muscle gain)
- 5.Target value
- 6.Progress

•Reports:

- 1.Reports ID (Primary Key)

- 2.User ID (Foreign Key)
- 3.Graph Data
- 4.Report type (e.g., weekly, monthly)
- 5.Report data (e.g., weight change, calories burned)

•Integration:

- 1.Integration ID (Primary Key)
- 2.User ID (Foreign Key)
- 3.Date
- 4.Integrated device/app name
- 5.Data synced

•User Interface:

- 1.UI ID (Primary Key)
- 2.User ID (Foreign Key)
- 3.Language preference
- 4.Theme preference
- 5.Font size
- 6.Accessibility settings

•Security:

- 1.Security ID (Primary Key)
- 2.User ID (Foreign Key)
- 3.Date
- 4.Login credentials
- 5.Encryption keys
- 6.Firewall settings

5.1.3 Logical Description of Data

The Logical Description of Data for a health and fitness app would involve outlining the data that the app will collect and how it will be organized and stored. This includes

information such as user profiles, workout plans, exercise logs, nutrition information, and any other relevant data. The logical description of data also includes how the data will be accessed and modified by the app, as well as any constraints or rules that must be followed when working with the data. It should also include any data validation rules that need to be in place to ensure data integrity and accuracy. Additionally, it should also include the data backup and recovery procedures to ensure that the data is secure and can be easily retrieved in case of any failure.

5.2 System Procedural Design

5.2.1 Flow chart or activity design

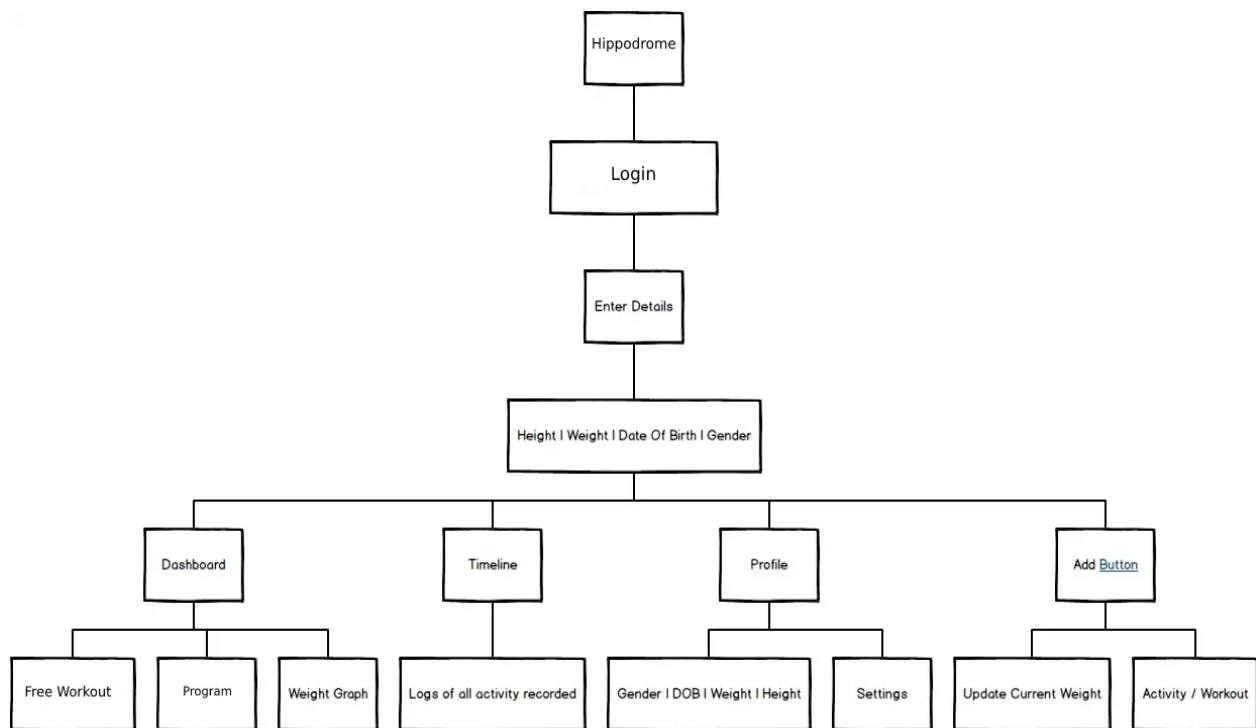
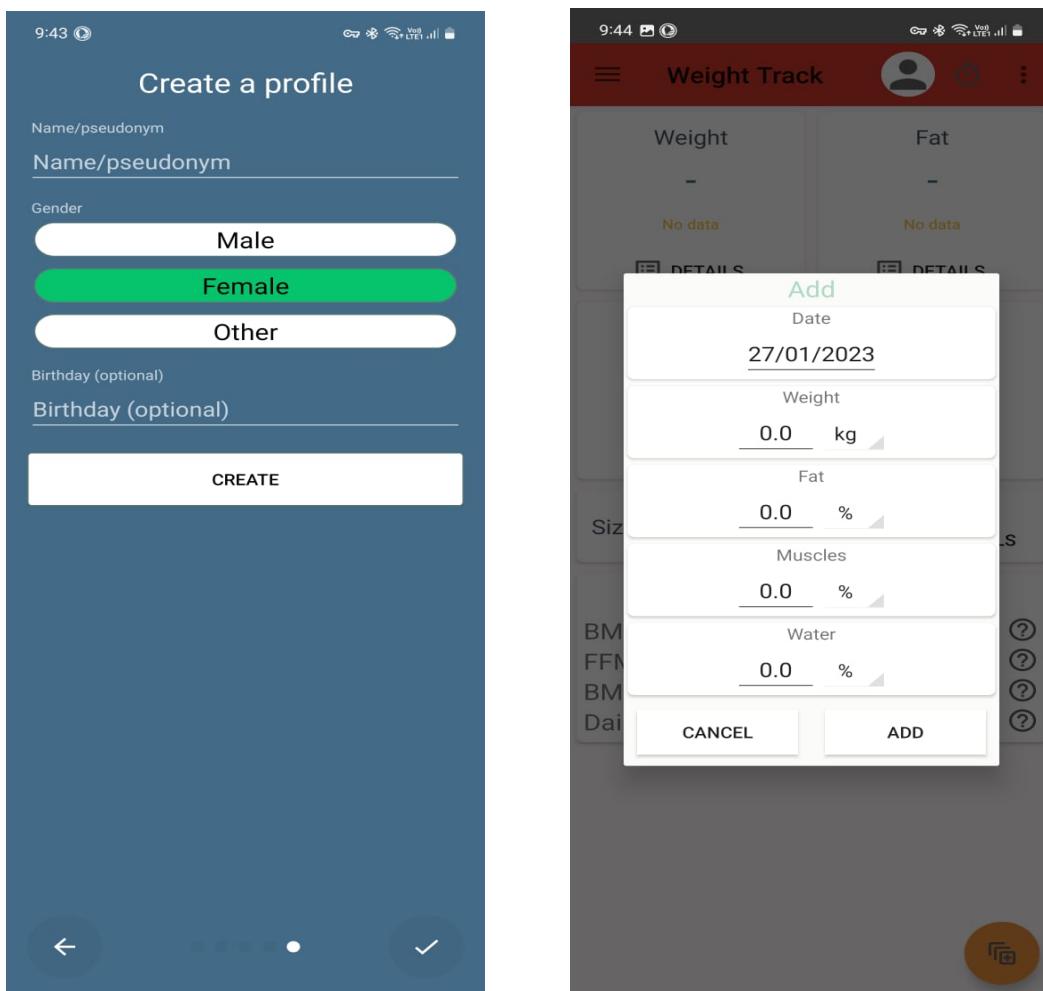


Fig ~ 5.2.1 Flow chart diagram

5.3 Input/Output and Interface Design

5.3.1 Samples of Forms and Interface

Samples of forms in a health and fitness app could include registration forms for new users, login forms for existing users, forms for inputting personal information such as weight, height, and fitness goals, forms for tracking progress such as exercise logs, and forms for setting reminders and notifications. These forms may be designed to be user-friendly and easy to navigate, with clear instructions and prompts for inputting information. They may also include validation checks to ensure that the information entered is accurate and consistent with the user's goals and preferences.



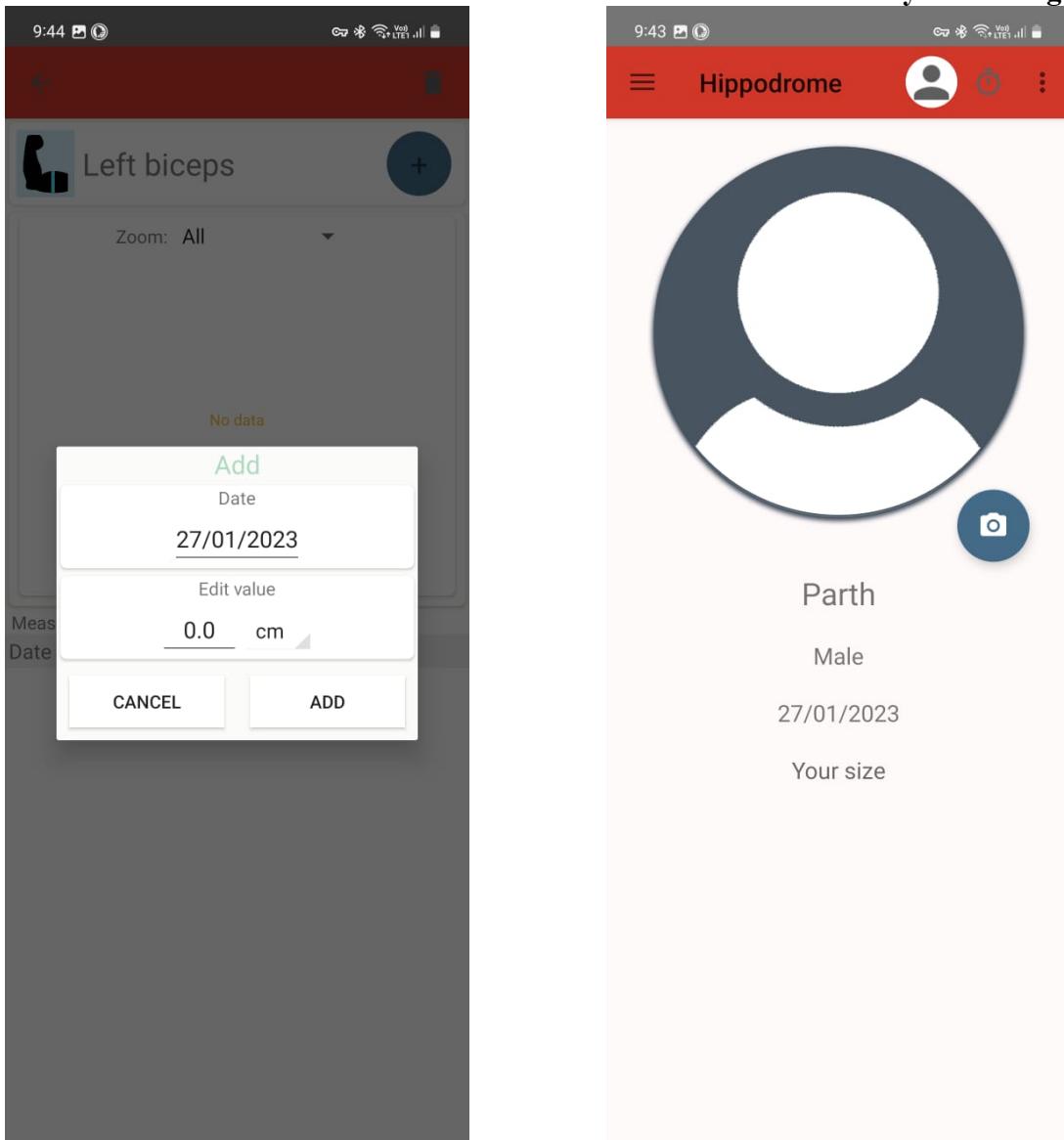


Fig ~ 5.3.1 Interface design

5.3.2 Access Control and Security

Access control and security are important considerations for any software application, particularly those related to health and fitness. Access control refers to the mechanisms that are put in place to ensure that only authorized users can access the system and its data. This can include user authentication, such as login credentials, as well as role-based access

control, which limits access to certain features or data based on a user's role within the organization.

Security, on the other hand, refers to the measures that are put in place to protect the system and its data from malicious attacks or unauthorized access. This can include encryption, firewalls, and intrusion detection systems, as well as regular security audits and vulnerability assessments.

When it comes to a health and fitness app, it's especially important to ensure that user data is kept private and secure. This may include personal information such as name, date of birth, and medical history, as well as sensitive information such as fitness data, such as workout routines and progress.

To ensure a robust access control and security for the health and fitness app the following steps can be taken:

- Implementing user authentication, such as login credentials
- Implementing role-based access control
- Encrypting sensitive data
- Using firewalls and intrusion detection systems
- Regular security audits and vulnerability assessments
- Following regulatory policies and compliance standards such as HIPAA, GDPR
- Having regular backups and recovery plan in place.

6.0 Implementation Planning and details

6.1 Implementation Environment (Single vs Multiuser, GUI vs Non-GUI)

- 1) *Single User vs. Multi-User:* A single-user health and fitness app is designed for use by a single individual, while a multi-user app allows multiple individuals to create and manage their own profiles and track their progress.
- 2) *GUI vs. Non-GUI:* A GUI (graphical user interface) health and fitness app uses visual elements such as icons, buttons, and images to make the app more user-friendly and easier to navigate. A non-GUI app, on the other hand, relies on text-based commands and inputs.
- 3) *Mobile vs Web-based:* A mobile health and fitness app is designed for use on a mobile device such as a smartphone or tablet, while a web-based app is designed for use on a computer or laptop.
- 4) *Cloud-based vs on-premises:* A cloud-based health and fitness app is hosted on a remote server and can be accessed from any device with an internet connection, while an on-premises app is installed on the user's device and can only be accessed on that device.
- 5) *Standalone vs integrated:* A standalone health and fitness app is a separate app that focuses on a specific aspect of health and fitness, while an integrated app is one that integrates with other apps or devices to provide a more comprehensive view of the user's health and fitness.

6.2 Program/Modules Specification

- **User Profile:** This module allows users to create and manage their personal profile, including information such as their name, age, weight, height, and other personal information.

- Activity Tracking: This module allows users to track their physical activity, such as steps taken, distance travelled, and calories burned. It may also include features such as GPS tracking and integration with other fitness tracking devices.
- Nutrition Tracking: This module allows users to track their food intake, including information such as calories, macronutrients, and micronutrients. It may also include features such as a food database and the ability to scan barcodes to track food intake.
- Workout Plans: This module provides users with pre-made workout plans and exercises, including information on the number of sets, reps, and weight to use. It may also include videos or animations that demonstrate how to perform each exercise correctly.
- Progress Tracking: This module allows users to track their progress over time and view their progress in terms of weight, body measurements, and other metrics.^[1]
- Social Integration: This module allows users to connect with friends and share their progress, compete with them or creating groups.
- Reminders and Notifications: This module allows users to set reminders for workouts, meals, and other activities, and receive notifications about upcoming events or milestones.
- Messaging and Support: This module allows users to connect with trainers, coaches, or other health professionals for support and guidance.
- Rewards and Gamification: This module provides users with rewards for achieving certain milestones or completing certain activities, such as badges, trophies, or virtual currency.
- Analytics: This module provides users with insights and metrics on their progress, such as calories burned, steps taken, and distance travelled.
- Integration with external devices: This module allows the app to sync data from external devices such as fitness trackers, smartwatches, or other wearables.
- In-app purchases: This module allows users to purchase additional features, such as additional workout plans or in-app currency, to enhance their experience.
-

6.3 Security Features

- Encryption: Sensitive data such as personal information, health records, and login credentials should be encrypted both in transit and at rest to protect against data breaches and unauthorized access.
- Authentication and Authorization: Users should be required to authenticate themselves before accessing the app, and the app should have a robust authorization system in place to ensure that only authorized users can access sensitive information.
- Secure Communication: The app should use secure communication protocols such as HTTPS to protect data in transit and prevent eavesdropping or tampering.
- Input Validation: The app should validate all user input to prevent SQL injection or other types of attacks.
- Security Logging: The app should have a security logging system in place to track user activity and detect potential security breaches or suspicious behaviour.
- Vulnerability Management: The app should be regularly tested for vulnerabilities and patches should be applied as needed to fix any security issues that are identified.
- Access Control: The app should have a system of access controls in place to limit user access to sensitive data and ensure that only authorized users can access this data.
- Security incident response plan: The app should have a security incident response plan in place to handle security breaches or data breaches.
- Regular Audits: The app should be regularly audited by security experts to identify any potential vulnerabilities or security weaknesses.

6.4 Coding Standards

- Naming Conventions: Establishing clear naming conventions for variables, functions, and classes can help to make the code more readable and easier to understand. This includes things like using camelCase for variable and function names and using Pascal Case for class names.
- Formatting: A consistent formatting style can make the code more readable. This includes things like using indentation, line breaks, and whitespace to make the code more visually appealing and easier to navigate.
- Comments: Comments should be used to explain the purpose and function of the code and should be included to describe the reasoning behind specific decisions.
- Error handling: The app should have clear error handling strategies in place to handle unexpected situations, such as invalid user input or network errors. This includes things like displaying clear error messages to the user, logging errors for debugging, and handling edge cases.
- Security: The app should follow best practices for security to protect user data and prevent malicious attacks. This includes things like encrypting sensitive data, using secure authentication methods, and protecting against SQL injection.
- Performance: The app should be optimized for performance to ensure that it runs smoothly and efficiently. This includes things like minimizing the use of resource-intensive operations, and caching data when appropriate.
- Maintainability: The app should be written in a way that makes it easy to maintain and update over time. This includes things like using modular code, separating concerns, and following SOLID principles.
- Code Review: Code review should be mandatory before pushing the code to production to have a better code quality and to find potential bugs or errors.

6.5 Sample Coding

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.easyfitness">

    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        android:maxSdkVersion="28" />
    <uses-permission android:name="android.permission.ACTION_HEADSET_PLUG" />

    <uses-feature
        android:name="android.hardware.camera.any"
        android:required="false" />
    <uses-feature
        android:name="android.hardware.camera.autofocus"
        android:required="false" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/logo1"
        android:label="Hippodrome"
        android:requestLegacyExternalStorage="true"
        android:theme="@style/AppTheme"
        android:preserveLegacyExternalStorage="true"> <!--for when
requestLegacyExternalStorage will be removed-->

        <activity
            android:name=".MainActivity"
            android:configChanges="orientation|screenSize"
            android:label="Hippodrome"
            android:windowSoftInputMode="stateUnchanged">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
                <action android:name="android.intent.action.VIEW" />
            </intent-filter>
        </activity>
        <activity
            android:name=".intro.MainIntroActivity"
            android:theme="@style/Theme.Intro" />
        <activity
            android:name="com.theartofdev.edmodo.cropper.CropImageActivity"
            android:theme="@style/Base.Theme.AppCompat" />

        <meta-data
            android:name="firebase_crash_collection_enabled"
            android:value="@bool/FIREBASE_CRASH_ENABLED" />

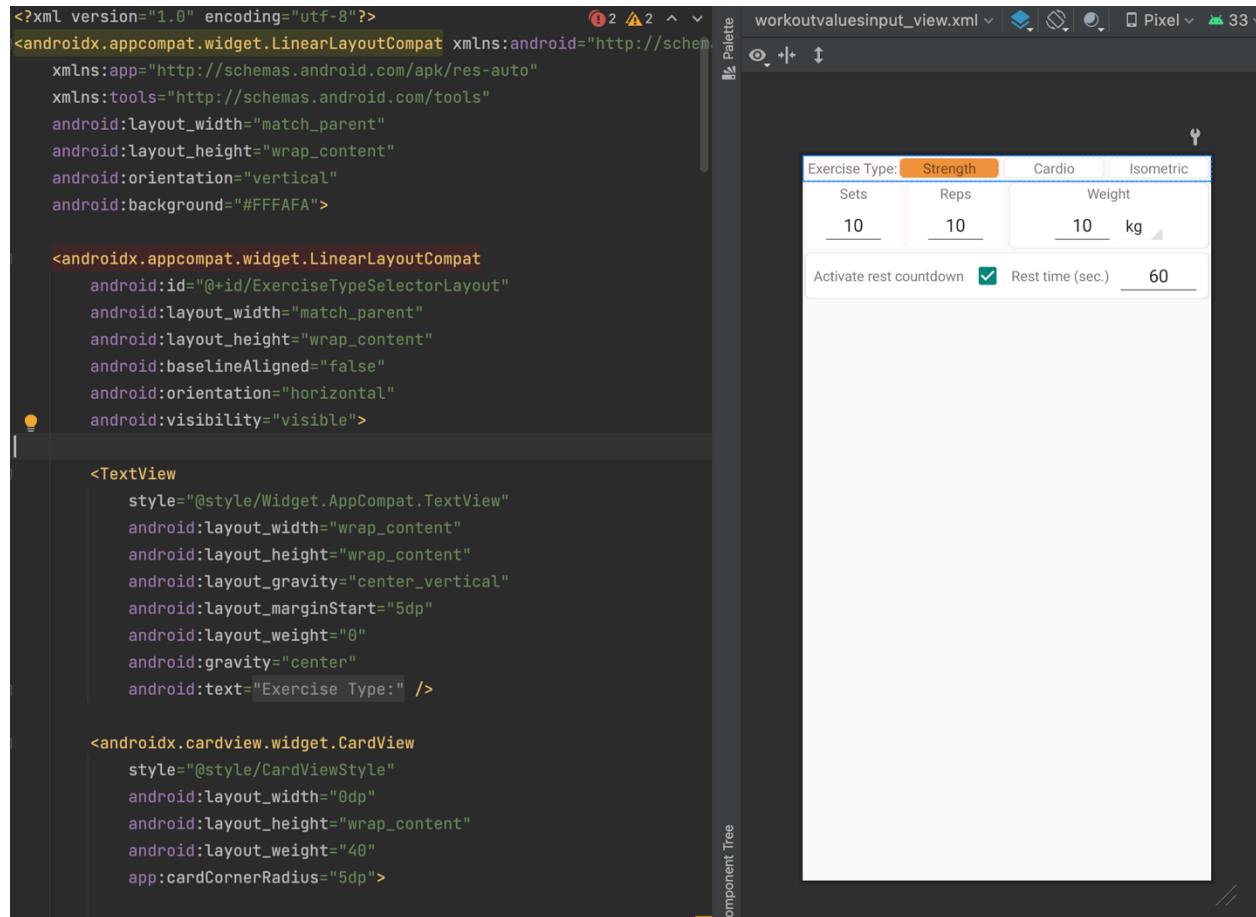
        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="${applicationId}.fileprovider"
            android:exported="false"

```

```
        android:grantUriPermissions="true">
        <meta-data
            android:name="android.support.FILE_PROVIDER_PATHS"
            android:resource="@xml/provider_paths" />
    </provider>

    <receiver
        android: name=".utils.AlarmReceiver"
        android: process=":remote" />

</application>
</manifest>
```



The screenshot shows the Android Studio interface with two tabs open: 'tab_weight.xml' and 'tab_graph.xml'. The 'tab_weight.xml' tab displays the following XML code:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFAFA">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:name="tab_weight"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:orientation="vertical"
            android:paddingLeft="0dp"
            android:paddingTop="0dp"
            android:paddingRight="0dp"
            android:paddingBottom="0dp">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="start"
                android:gravity="center"
                android:orientation="horizontal">

                <androidx.cardview.widget.CardView
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_gravity="center_vertical|center_horizontal"
                    android:layout_margin="5dp">

```

The preview window shows a card-based dashboard with the following data:

	Weight	Fat
70.5	50.9	
No chart data available.	No chart data available.	
Muscles	Water	
20.2	20.5	
No chart data available.	No chart data available.	
Size	50	
Index		
BMI	20	no height available
FFMI	20	no fat available
BMR	1864	normal
Daily Cals	2731	

The screenshot shows the Android Studio interface with two tabs open: 'tab_weight.xml' and 'tab_graph.xml'. The 'tab_graph.xml' tab displays the following XML code:

```

<ScrollView xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fillViewport="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:isScrollContainer="true"
        android:orientation="vertical"
        android:paddingLeft="0dp"
        android:paddingTop="0dp"
        android:paddingRight="0dp"
        android:paddingBottom="0dp">

        <androidx.cardview.widget.CardView
            style="@style/CardViewStyle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:cardCornerRadius="5dp">

            <TableLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="8dp"
                android:layout_marginTop="5dp"
                android:layout_marginBottom="5dp"
                android:stretchColumns="1">

                <TableRow android:gravity="center_vertical">

```

The preview window shows a graph-based dashboard with the following settings:

- Exercise: Max Weight (Reps ≥ 1)
- Display: Max Weight (Reps ≥ 1)
- Filter: ALL, LAST YEAR, LAST MONTH, LAST WEEK

A note at the bottom right of the preview window states: "No chart data available."

7.0 Testing (choose appropriate testing strategy or techniques suitable to your system)

7.1 Testing Plan

- Test Objectives: The testing plan should begin by clearly defining the testing objectives, which should align with the overall goals and objectives of the app development.
- Test Cases: The testing plan should include a list of test cases that cover all the functional and non-functional requirements of the app. These test cases should be detailed and include expected results and pass/fail criteria.
- Test Data: The testing plan should specify the test data that will be used to test the app, including sample data and test cases.
- Test Environment: The testing plan should define the test environment that will be used to test the app, including the devices, operating systems, and network conditions that will be used.
- Test Schedule: The testing plan should include a schedule for testing, including the start and end dates, and the resources that will be required.
- Test Resources: The testing plan should outline the resources that will be required to complete the testing, including equipment, personnel, and software.
- Test Deliverables: The testing plan should specify the deliverables that will be produced as a result of the testing, such as test reports, bug reports, and test results.
- Test Sign-off: The testing plan should include a process for obtaining sign-off on the testing, including acceptance criteria and a sign-off form.
- Maintenance: The testing plan should include a process for maintaining the app, including updating the app based on user feedback, fixing bugs and errors, and implementing new features.
- Monitoring and reporting: The testing plan should include a process for monitoring the testing process and reporting any issues or bugs that are identified.

7.2 Testing Strategy

- Define the testing objectives: The first step in creating a testing strategy is to define the testing objectives, which should align with the overall goals and objectives of the app development.
- Identify the target audience: Identify the target audience for the app, including the type of users and their needs. This will help to ensure that the testing is relevant and addresses the needs of the users.
- Choose the right testing methods: Choose the right testing methods for the app, including unit testing, functional testing, performance testing, security testing, usability testing, and acceptance testing.
- Create a test plan: Create a detailed test plan that outlines the testing methods that will be used, the resources required, and the schedule for testing.
- Define test cases: Define test cases that cover all the functional and non-functional requirements of the app. These test cases should be detailed and include expected results and pass/fail criteria.
- Define test data: Define the test data that will be used to test the app, including sample data and test cases.
- Define test environment: Define the test environment that will be used to test the app, including the devices, operating systems, and network conditions that will be used.
- Define test schedule: Define a schedule for testing, including the start and end dates, and the resources that will be required.
- Monitoring and reporting: Monitor the testing process and report any issues or bugs that are identified.
- Continuously improve: Continuously improve the testing strategy based on the feedback and results from the testing process.

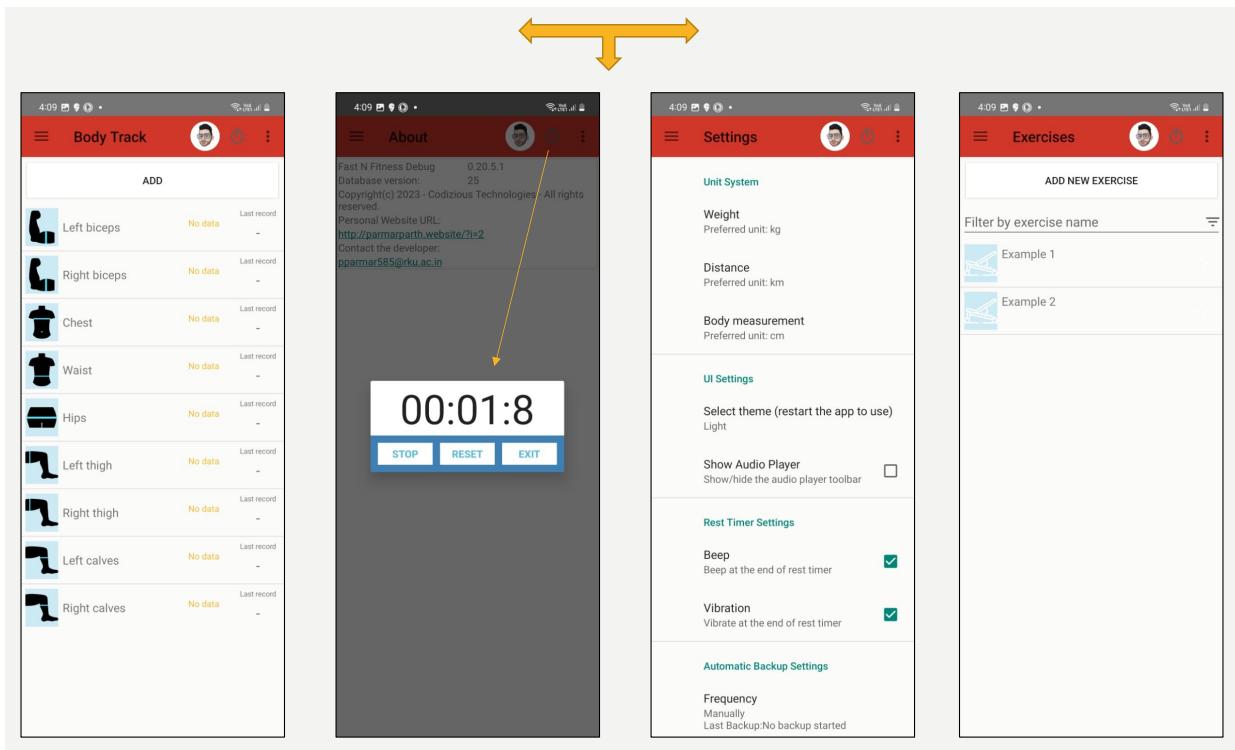
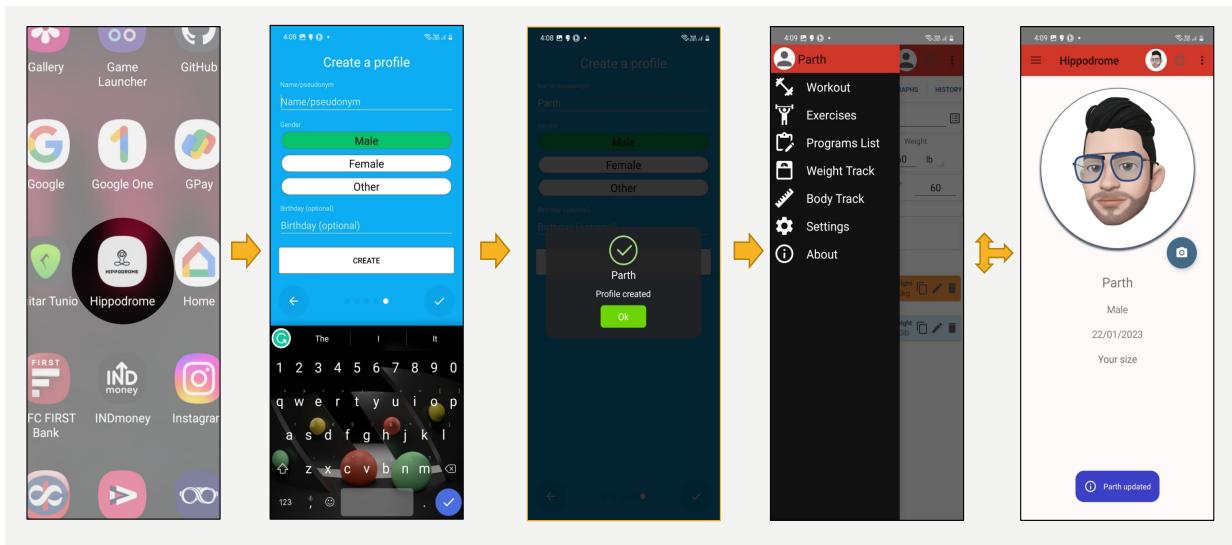
7.3 Testing Methods

- There are several different testing methods that can be used to test a health and fitness app, including:
- Unit Testing: Unit testing is a method of testing individual units of code, such as functions or methods, to ensure that they work as expected. This type of testing is typically done by developers and can help identify and fix bugs early in the development process.
- Functional Testing: Functional testing is used to test the functionality of the app, including its user interface and features. This type of testing is typically done by quality assurance (QA) teams and can help identify any issues that may arise during real-world usage.
- Performance Testing: Performance testing is used to evaluate the performance of the app under various conditions, such as different device types and network conditions. This type of testing can help identify any issues with the app's performance and scalability.
- Security Testing: Security testing is used to evaluate the app's security features and identify any vulnerabilities that may be present. This type of testing can include penetration testing, which simulates attacks on the app to identify any vulnerabilities, and vulnerability scanning, which scans the app for known vulnerabilities.
- Usability Testing: Usability testing is used to evaluate the app's user interface and ensure that it is easy to use and understand. This type of testing typically involves recruiting users to test the app and provide feedback on its usability.
- Acceptance Testing: Acceptance testing is used to verify that the app meets the requirements and is ready for release. This type of testing is typically done by the customer or end-user and can include both functional and non-functional testing.

7.4 Test Cases (Purpose, required output, Expected Result)

- 1) Purpose: Verify that the app can accurately track and display a user's fitness data.
 - a. Required output: The app should be able to track and display data such as steps taken, distance travelled, and calories burned.
 - b. Expected result: The app should accurately track and display the user's fitness data, with no errors or inaccuracies.
- 2) Purpose: Verify that the app can provide personalized workout plans.
 - a. Required output: The app should be able to create workout plans based on the user's fitness level, goals, and preferences.
 - b. Expected result: The app should provide workout plans that are tailored to the user's needs and goals.
- 3) Purpose: Verify that the app can provide reminders and notifications.
 - a. Required output: The app should be able to send reminders and notifications to the user, such as reminding them to stay hydrated or to take their medications.
 - b. Expected result: The app should send reminders and notifications to the user as scheduled, and they should be received and displayed correctly.
- 4) Purpose: Verify that the app can handle user data security and privacy
 - a. Required output: The app should be able to handle user data security and privacy and comply with relevant regulations
 - b. Expected result: The app should handle user data security and privacy, and it should pass all relevant regulations and compliance standards.
- 5) Purpose: Verify that the app can handle different languages and localization
 - a. Required output: The app should be able to handle different languages and localization settings
 - b. Expected result: The app should handle different languages and localization settings seamlessly and accurately.

8.0 Screen shots and User manual



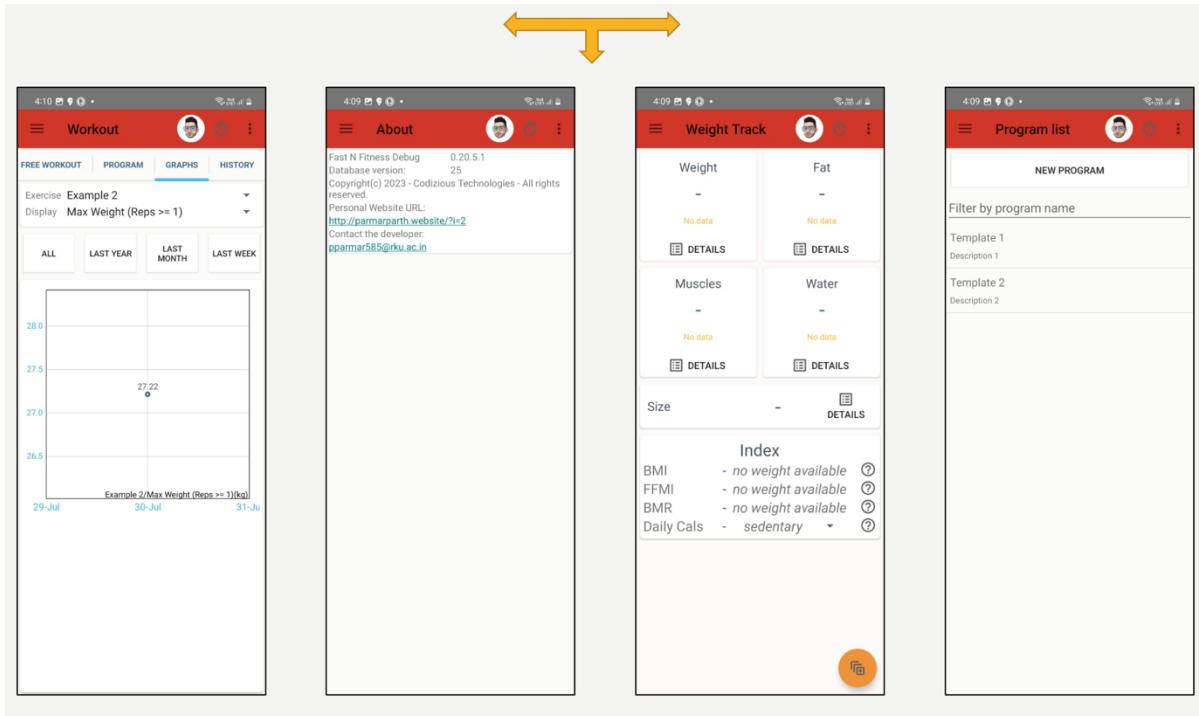


Fig ~ 8.0 Screen shots

9.0 Limitation and Future Enhancement

- Limited data storage: Some health and fitness apps have limited storage capacity, which can make it difficult for users to save and access large amounts of data over time.
- Limited device compatibility: Some health and fitness apps may only be compatible with certain types of devices, such as smartphones or tablets, which can limit their accessibility for users who own different types of devices.
- Limited features and functionality: Some health and fitness apps may only offer a limited set of features and functionality, which can make it difficult for users to track all the information and metrics they need to stay on top of their health and fitness goals.
- Limited accuracy: Some health and fitness apps use sensors or algorithms that may not be completely accurate, which can lead to inaccurate data and tracking.
- Limited personalization: Some health and fitness apps may not offer enough personalization options, which can make it difficult for users to tailor the app to their specific needs and preferences.
- Limited scalability: Some health and fitness apps may not be designed to handle many users, which can make it difficult for the app to scale up as its user base grows.

Despite these limitations, there are always ways to enhance and improve a health and fitness app. Here are a few examples:

- Increase data storage: Future enhancements could include increasing the storage capacity of the app, which would allow users to save and access more data over time.
- Increase device compatibility: Future enhancements could include making the app compatible with a wider range of devices, which would increase accessibility for users.

- Add new features and functionality: Future enhancements could include adding new features and functionality to the app, such as new workout plans, tracking options, or social integration.

10.0 Conclusion and Discussion

In conclusion, the development of a health and fitness app involves a wide range of considerations that must be considered to ensure its success. A well-designed and developed health and fitness app can provide users with the tools and resources they need to improve their physical and mental well-being.

It's important to clearly define the goals and objectives of the app, as well as its scope and target audience. The app should be based on a thorough review of existing literature and technology, and a well-planned development approach should be used.

It's also important to consider the hardware and software requirements of the app, as well as any constraints such as regulatory policies or safety and security considerations.

As part of the development process, it is important to identify and manage risks, and to have a clear plan in place for testing and launching the app.

Once the app is developed and launched, it's important to provide ongoing support and maintenance, such as addressing any bugs or issues that arise and updating the app to ensure compatibility with new operating systems and hardware.

Overall, the development of a health and fitness app requires a comprehensive approach, with careful planning and attention to detail to ensure a successful outcome.

References

•M.H. Mobasher *et al.*

Smartphone breast applications—what's the evidence?

Breast (2014)

•L. Willcocks *et al.*

Beyond the IT productivity paradox

European Management Journal (1996)

•D.E. Conroy *et al.*

Behavior changes techniques in top-ranked mobile apps for physical activity

Am J Prev Med (2014)

•C.L. Backinger *et al.*

Where there's an app, there's a way?

Am J Prev Med (2011)

•C.M. Wharton *et al.*

Dietary self-monitoring, but not dietary quality, improves with use of smartphone app technology in an 8-week weight loss trial

J Nutr Educ Behav (2014)