

Performa For The Approval Project Proposal

PNR No.:

Rollno:

1. Name of the Student:

2. Title of the Project:

3. Name of the Guide:

4. Teaching experience of the Guide:

5. Is this your first submission?

Yes

☐

No

☐

Signature of the Student

Signature of the Guide

Date:

Date:.....

Signature of the coordinator

Date:

INDEX

Sr. No.	Topic	Date
1.	Introduction	
2.	Survey Of Technologies	
3.	Requirement and Analysis	
4.	System Design	

Chapter 1

Introduction

1.1 Background

“I do not fear the man who has practiced 10000 types of kicks one time, but the man who has trained 1 kick 10000 times.” ~ Bruce lee

Within the realm of fitness and training, many **individuals** encounter challenges related to monitoring and feedback, which often lead to inconsistencies in their exercise routines. "AI-based Mentor for MMA," abbreviated as AIMMMA, is an application developed to offer users a simple yet valuable tool for tracking and enhancing their fitness experiences. While the initial focus of this application revolves around fundamental exercises such as punches, kicks, pushups, squats, and their combinations, its primary aim is to provide users with an accessible platform for monitoring, evaluating and tracking their progress.

In today's fitness landscape, an abundance of high-tech fitness apps and devices are available. However, these tools often fall short in delivering personalised guidance and feedback tailored to individual needs and goals. AIMMMA recognizes these limitations and endeavours to address them by concentrating on tracking body movement using Computer vision.

What sets this application apart is its dedication to providing a straightforward and user-friendly experience. Its core functionality centres around tracking and recording basic exercises using Computer vision that constitute vital components of MMA training. Users can depend on the application to accurately monitor, log and analyse their punches, kicks, pushups, squats, and combinations thereof.

The primary objective of this application is to assist users in maintaining a consistent

fitness routine by furnishing them with a convenient tool for tracking their workouts and seamlessly getting real-time audio feedback. It is important to emphasise that the application does not seek to replace the physical effort required for exercise but rather aims to enhance it by offering a convenient means of progress tracking.

While future iterations of the application may consider expanding its features and capabilities, the current version provides a practical solution for individuals seeking to monitor their performance in fundamental exercises. The application remains committed to delivering a straightforward and user-centric experience, making it an accessible choice for those interested in basic MMA-inspired workouts.

1.2 Objectives

- The ability to update ML models with additional data, ensuring that the models adapt and learn from new information over time.
- Use of computer vision and artificial intelligence to monitor and record movements during activities such as punching, kicking, pushups, and squats accurately.
- To leverage computer vision and AI to recognize complex combinations of movements or exercises performed by the user accurately.
- Real-time monitoring of exercise sessions, including measuring the pace of repetitions and timing intervals between sets, ensuring effective workout tracking.
- Provide real-time feedback to users during their workouts through pre-recorded audio cues or instructions, helping them improve their form and performance.
- Development of an intuitive and user-friendly interface that allows users to easily track their fitness progress, view statistics, and set goals.
- Implementing robust measures to ensure the privacy and security of user data, particularly when collecting and processing personal fitness information using AI.
- Providing post-workout analysis that helps individuals get aware about their shortcomings which can help improve their future performance.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

The purpose of the AI-based Mentor for MMA (AIMMMA) application is to simplify fitness tracking for individuals engaged in mixed martial arts (MMA) training. This application offers an intuitive platform to monitor, evaluate and record user data for their basic exercises such as punches, kicks, pushups, squats, and combinations thereof. Its primary goal is to provide users with a user-friendly tool that enhances their fitness experiences by facilitating progress tracking and analysis.

1.3.2 Scope

The scope of this application is centred on MMA training, particularly the tracking of fundamental exercises. It employs computer vision and artificial intelligence to accurately monitor and record these exercises, providing real-time feedback to users. While its initial focus is on basic exercises, the application has the potential to expand its scope to cover a broader range of MMA-related activities and even adapt to other fitness disciplines.

1.3.3 Applicability

This application is applicable to MMA enthusiasts and individuals interested in tracking their progress during training sessions. It caters to users of varying experience levels, from beginners seeking guidance to advanced practitioners looking for a convenient way to monitor their workouts. Its user-friendly design and accurate exercise tracking make it a valuable tool for anyone engaged in MMA-inspired workouts, ensuring that users can maintain consistency and improve their performance effectively.

1.4 Achievements

The AIMMMA project has reached significant milestones throughout its development journey, demonstrating exceptional accomplishments. It has not only effectively fulfilled all project requirements but has also seamlessly delivered the

anticipated functionalities.

The application boasts a user-friendly interface, ensuring effortless navigation and interaction for its users. The project's primary objectives, centred around precise exercise tracking and real-time AI feedback, have been triumphantly achieved. Users can now confidently monitor their progress in MMA training and gain valuable insights, all within the framework of a straightforward and intuitive user experience. These accomplishments underscore AIMMMA's unwavering commitment to elevating fitness tracking and providing robust support for MMA enthusiasts, cementing its status as an invaluable tool for those looking to enhance their training regimens.

1.5 Organization of Report

1.5.1 Requirement and Analysis

- The proposed system is an AI based application developed using C++ and TensorFlow. The objective of the system is to provide users with a user-friendly and efficient tool that allows them to track, monitor and analyse their performance in real-time as well as post workout.
- Software requirements used for the system include NodeJS, TensorFlow. Additionally, web technologies such as HTML, CSS, and JavaScript will be used for the user interface. More libraries like React, Typescript would be utilized to develop a slick front-end. React apps could later be migrated to native apps using React Native.
- Hardware requirements for the system include a laptop or computer running on Windows operating system for back-end with a minimum of 8GB RAM, an Intel Core i5 processor, 11th gen or greater or equivalent of it, and a display resolution of 1024x768 or higher. And a client device to test the app such as Mobile.
- This chapter also shows which all conceptual models will be prepared while making the project/system such as Use Case Diagram, Activity Diagram and many more.
- Softwares that would be used to develop and test the applications include:

- Browser - Preferably Chrome.
- IDE - Visual Studio C++

1.5.2 System Design

- The system design is included in all conceptual schema diagrams which explains the interrelation between the modules and their dependencies.
- The features, working and flow of the entire system is described in this chapter.

1.5.3 Implementation and Testing

- Implementation process is described in this chapter.
- Describes which all testing approaches are going to take place in this project.

1.5.4 Results and Discussion

- In the end, we got a full-fledged application that runs on a server and uses RESTful APIs to control the client side app.

1.5.5 Conclusion

The proposed system is a website that provides image editing features such as background changing, image enhancement, and cropping. The system will be developed using Python's Django framework and will use web technologies like HTML, CSS, and JavaScript. The hardware requirements for the system include a computer with at least 4GB of RAM and a Windows operating system. The system aims to provide users with a user-friendly interface and efficient image editing tools for various use cases.

Chapter 2

Survey of Technology

Core Component

- **C++**, a high-performance language that is well-suited for machine learning and backend development, is at the heart of this project. C++ is known for its speed and efficiency, making it ideal for complex tasks such as training and deploying machine learning models. Additionally, C++ is a versatile language that can be used to develop a wide range of applications, from backend systems to web applications to machine learning models.
- **TensorFlow**, a popular open-source machine learning framework, is used in conjunction with C++ to develop and train the machine learning model for this project. TensorFlow provides a wide range of features for building and training machine learning models, including pre-trained models, layers, and optimizers. This makes it easy to develop and deploy complex machine learning models without having to start from scratch.
- **Protobuf**, a protocol buffer language for serialising structured data, is used to facilitate communication between the core component and the frontend. Protobuf is efficient and lightweight, making it a good choice for this task. Additionally, Protobuf is a language-agnostic format, which means that it can be used with a variety of programming languages, including C++ and JavaScript.

Neural Network Training

- **P5.js**, a JavaScript library for creating interactive graphics and animations, is used to develop a user interface for training the machine learning model. P5.js makes it easy to create visually appealing and interactive interfaces, which can help users to understand and interact with the machine learning training process.
- **ML5**, a JavaScript library for machine learning in the browser, is used to train

the machine learning model using P5.js. ML5 provides a variety of machine learning algorithms, including linear regression, logistic regression, and neural networks. This makes it easy to train a variety of machine learning models without having to leave the browser.

Frontend

- **React**, a popular JavaScript library for building user interfaces, is used to develop the frontend of the web application. React is known for its speed, scalability, and ease of use. Additionally, React is a component-based library, which makes it easy to develop complex user interfaces by breaking them down into smaller, reusable components.
- **TensorFlow.js**, a JavaScript library for training and deploying machine learning models in the browser, is used to deploy the machine learning model to the frontend. TensorFlow.js makes it easy to deploy machine learning models to the browser without having to rely on a backend server. This allows users to interact with the machine learning model directly from the browser, without having to send data back and forth to a server.
- **Vis.js**, a JavaScript library for data visualisation, is used to visualise the neural network training process. Vis.js makes it easy to create interactive and informative data visualisations, which can help users to understand the machine learning training process.

Database

- **MongoDB**, a NoSQL database that is scalable and flexible, is used to store the data for the web application. MongoDB is a good choice for this task because it is easy to use and can handle large amounts of data efficiently.

Communication

- **HTTP**, a protocol for communication between the client and server, is used to communicate between the frontend and the core component. HTTP is a

well-established protocol that is supported by all major browsers and servers. Additionally, HTTP is a stateless protocol, which means that each request is independent of all other requests. This makes it easy to scale the web application horizontally by adding additional servers.

- **WebSockets**, a protocol for real-time communication between the client and server, is used to provide real-time feedback to the user during the neural network training process. WebSockets allows the frontend to receive updates from the core component without having to send a request. This makes it possible to create a more responsive and interactive user experience.

Chapter 3

Requirement and System Analysis

3.1 Problem Definition

3.1.1 Problem Definition

Current system lacks the ability to track body pose and classify exercises effectively. They do not analyse the speed of movements, monitor reps and sets using AI technology. Losing crucial real time and deep insights on form and speed of exercises.

3.1.2 Existing System

AI-powered fitness apps can design effective diet and workout plans based on user data, biological facts, and expert opinions. Some apps can track progress, but this may require manual input. Other apps integrate with wearables to track progress. Some apps provide video or 3D instructions on how to do exercises properly. Most apps use scientific data to provide general workout and diet plans and track progress. Few apps have body tracking capabilities, but this is not a core feature.

3.1.3 Proposed System

The proposed system can efficiently track body pose, provide feedback on camera angle and lighting before workouts for optimal body pose classification and analysis, accurately track particular movements (e.g., push-ups, punches, squats, kicks),

efficiently track pose changes and classify them (which can be used to track combos, sets, and reps of exercises), and is user-friendly and free. Most importantly, the system is extensible to newer exercises without significant code refactoring or model retraining.

3.2 Requirement Analysis

Functional

1. Record
2. Pose Detection and Analysis
3. Light Detection and Analysis
4. Camera Angle Detection and Analysis
5. Audio feedback.

Non-Functional

1. Usability
2. Reliability
3. Performance
4. Supportability
5. Safety
6. Security

3.3 Planning and Scheduling

	Task	Start	End	2023		2024	
				Nov	Dec	Jan	Feb
	Project ⊖	11/1/23	2/9/24				
1	Planning	11/1/23	11/22/23				
2	Designing	11/22/23	12/21/23				
3	Data Analysis	12/4/23	1/6/24				
4	Implementation	12/4/23	1/18/24				
5	Testing	1/18/24	2/9/24				

Figure 1: GANTT Chart

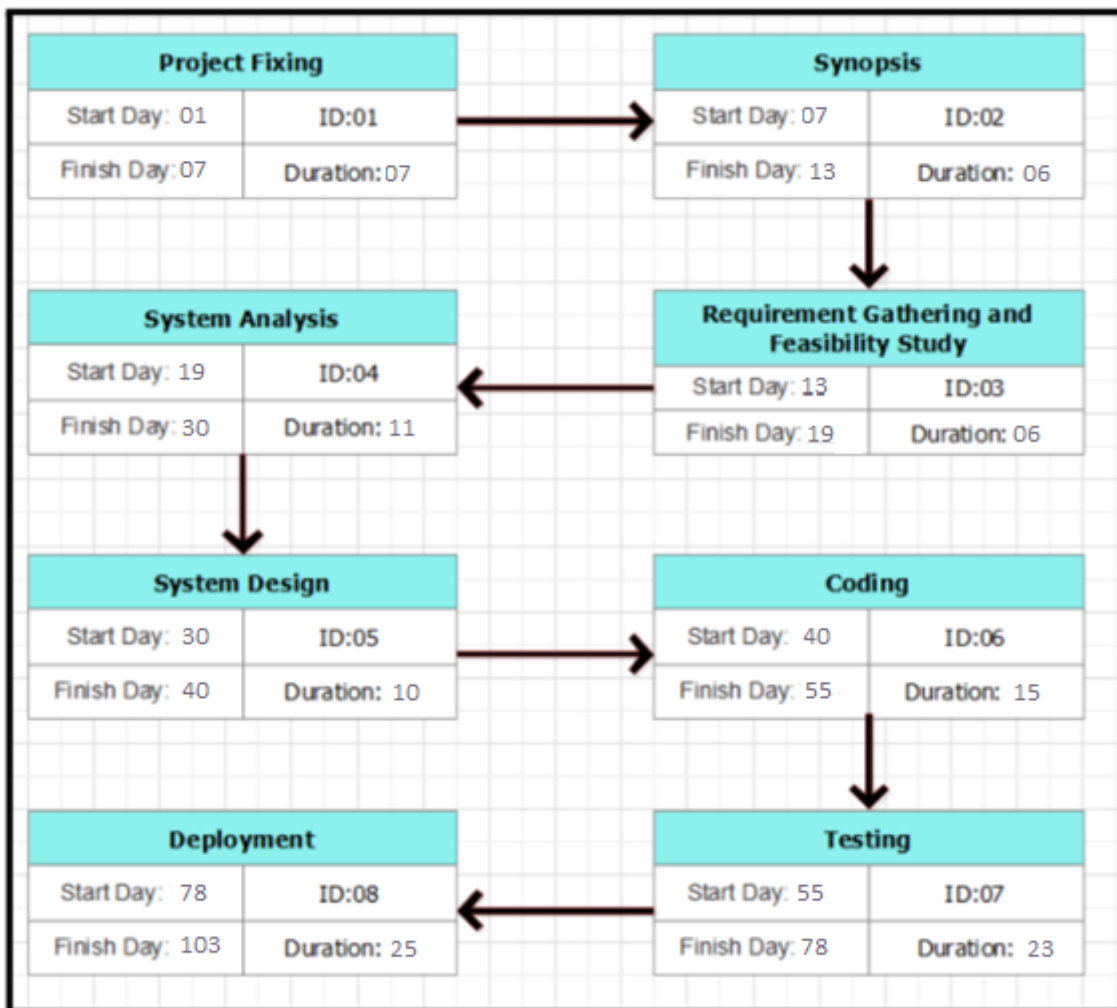


Figure 2: PERT Chart

3.4 Hardware Requirements

Computer

- CPU: Intel i5 12th gen or equivalent AMD CPU
- GPU: Nvidia RTX 2050 or higher (GTX 1650 models are also powerful enough)
- RAM: 8-10 GB
- Storage: Enough storage space to store your application data and machine learning models

Client

- Device: Any mobile or computer device with a camera and a browser that supports the Navigation API
- Speaker: A speaker or Bluetooth handset to receive audio feedback

3.5 Software Requirements

IDE:

- Visual Studio: A complete integrated development environment for building modern applications for Windows, macOS, Linux, iOS, and Android.
- Visual Studio Code: A lightweight but powerful code editor with support for many languages and frameworks.

Compiler:

- Microsoft C++: A compiler for the C++ programming language, developed by Microsoft.

Storage:

- MongoDB: A document database that stores data in JSON-like documents.
- MongoDB Driver for C++: A C++ driver for MongoDB.

Frameworks:

- TensorFlow: An open-source machine learning framework for building and deploying machine learning models.

Libraries:

- React: A JavaScript library for building user interfaces.
- ml5.js: A JavaScript library for machine learning in the browser.
- p5.js: A JavaScript library for creative coding.

- tfjs: A JavaScript library for TensorFlow.js.

3.6 Preliminary Product Description

AIMMMA is an AI-powered fitness tracking application designed to help MMA enthusiasts of all levels improve their training and performance. It uses computer vision and artificial intelligence to accurately monitor and track fundamental MMA exercises, such as punches, kicks, pushups, and squats. AIMMMA provides real-time feedback to users during their workouts, helping them improve their form and technique. It also generates post-workout analysis to help users identify areas for improvement.

Chapter 4

System Design

4.1 Basic Modules

4.1.1 Pose Classification and Research Interface

Pose classification made with ml5 and p5 libraries. They provide a rough prototypical interface on a white canvas. p5 library enables artists, designers, engineers to develop quick interfaces for faster prototyping ideas. ml5 and p5 together can be used to create a sophisticated interface to train various models for this application.

4.1.2 Reps Analyzer

An analyzer that can detect a rep which means a count of any exercise that the system is designed to classify. This module will use neural networks to detect the reps in a given stream of raw data.

4.1.3 Sets Analyzer

An analyzer that can detect a set which means a collection of reps and rests on the exercises that the system is designed to classify. This module will use neural networks to detect the sets in a given stream of analysed rep data.

4.1.4 Tempo Analyzer

An analyzer that can keep track of the tempo of each rep that is analysed from the Reps Analyzer.

4.1.5 Recorder

Recorder is a crucial component which requires access to the camera API in a web browser that supports Javascript Web APIs on a device with a camera. This component is accessed by many of the detectors. Recorder module will not only record but stream data to various processes from front-end to back-end and from

back-end to various modules.

4.1.6 Storage

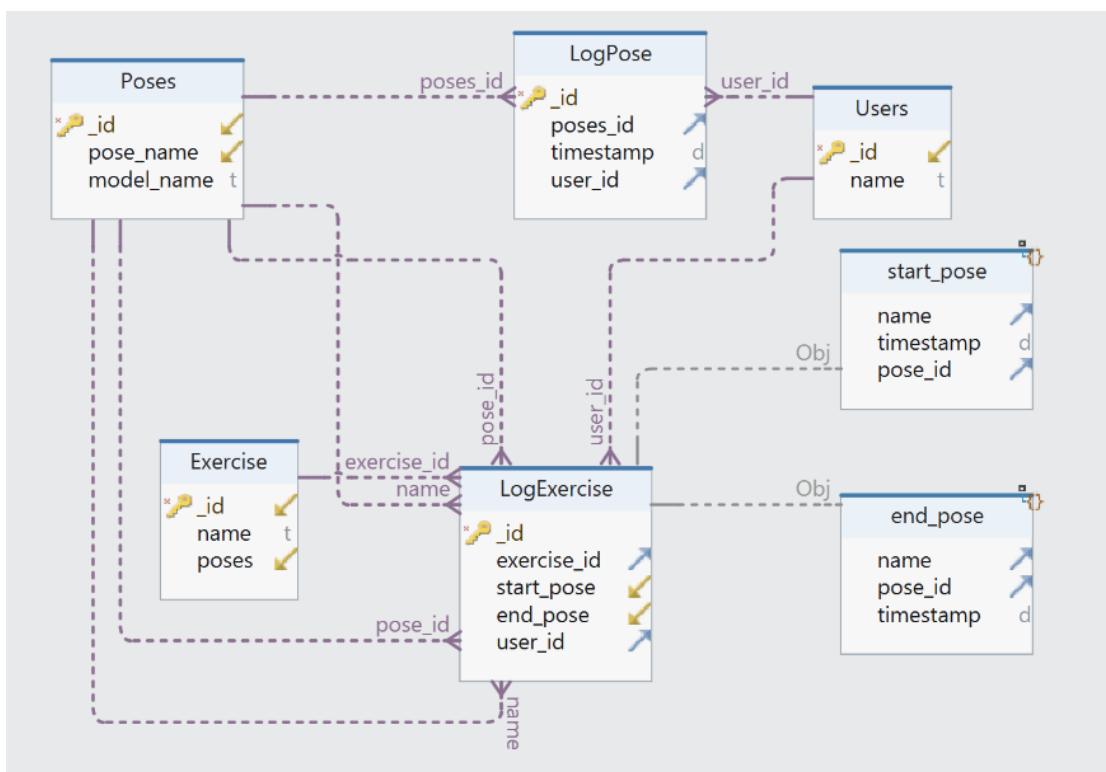
This module will handle all the storage related to operations, saving user generated, computer generated data for future analysis.

4.1.7 Audio Feedback

This is the front end module that will constantly listen to the backend updates using websockets technology. Controlled by the backend modules (analyzers) this will store and play appropriate pre-recorded audio cues for any feedback.

4.2 Data Design

4.2.1 Schema Design:



4.3 Procedural Design

4.3.1 Logic Diagrams:

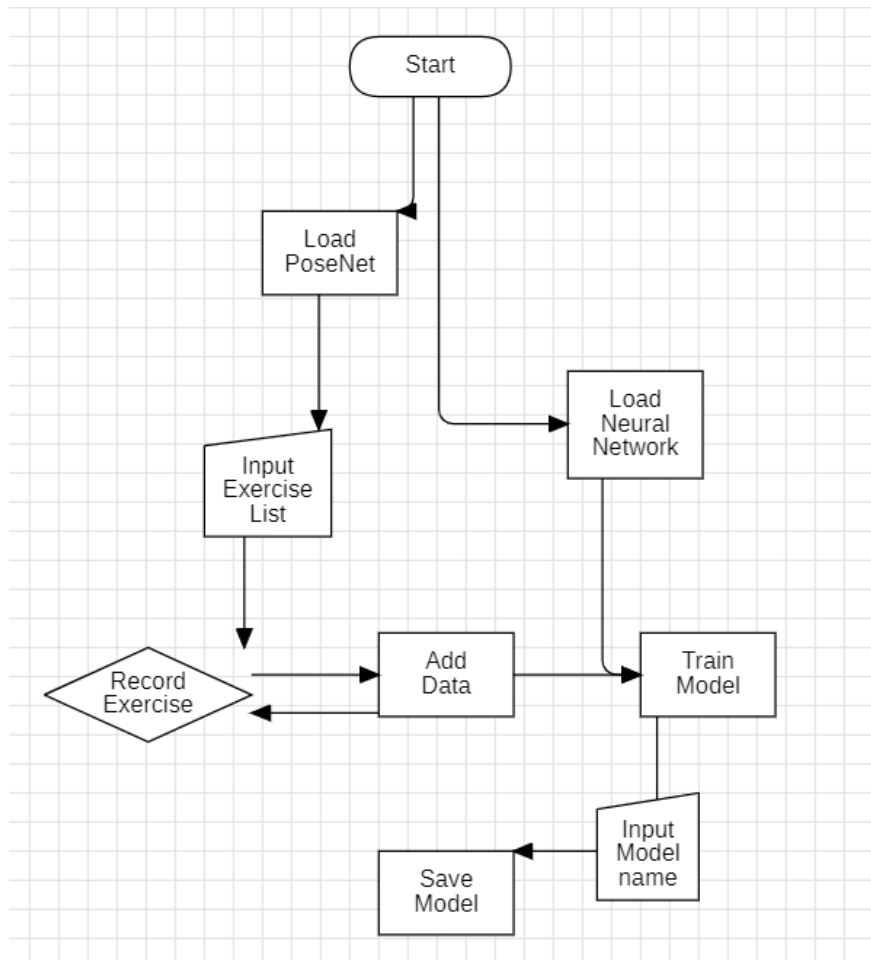


Figure: Pose Classification and Research Interface.

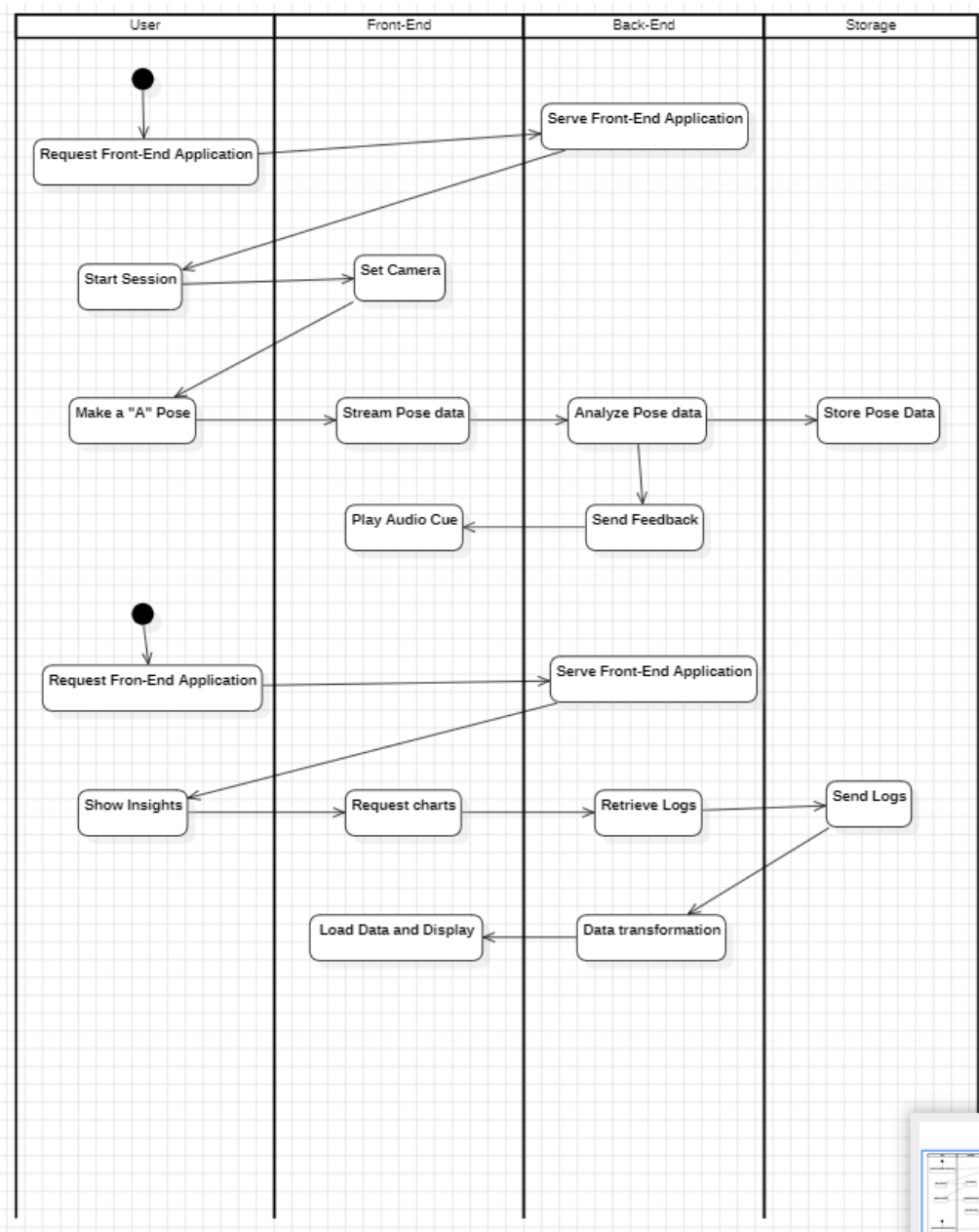


Figure: Activity Diagram

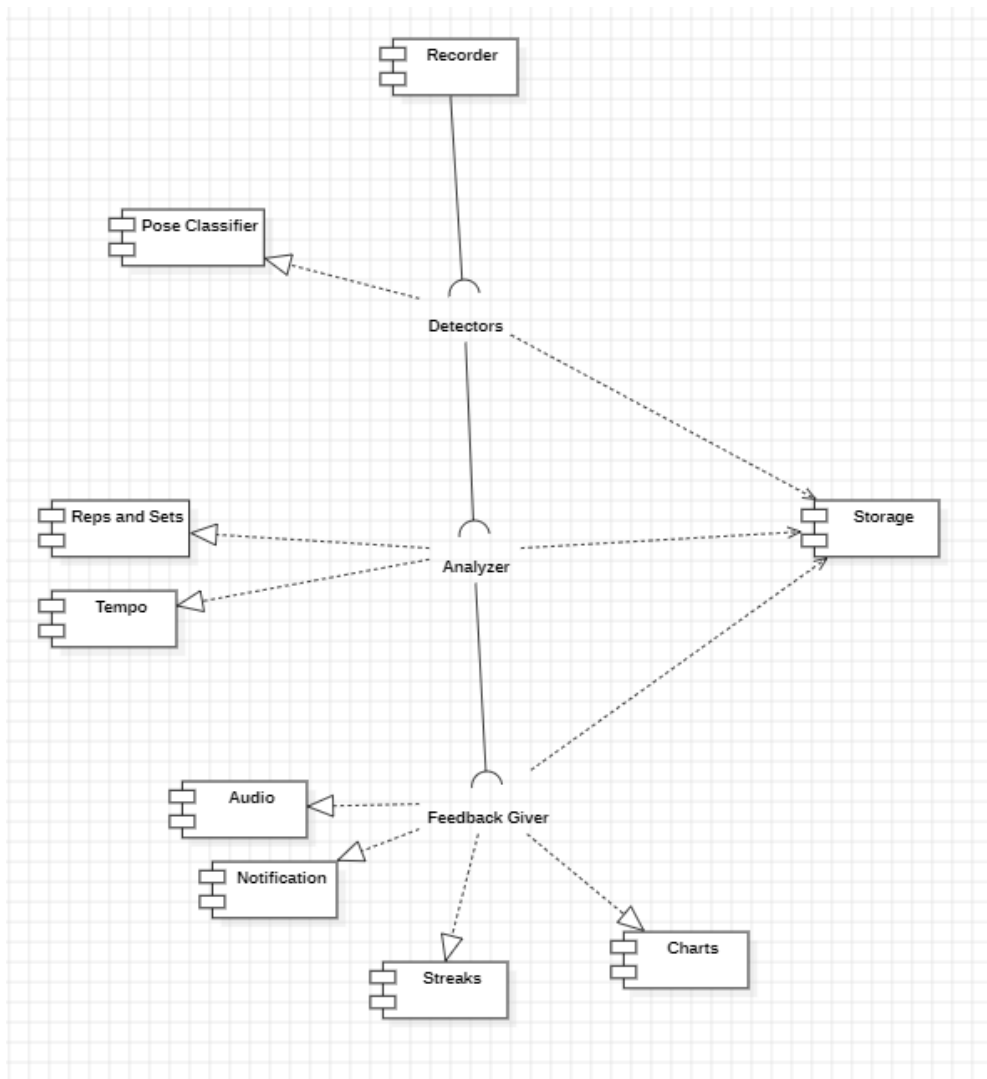
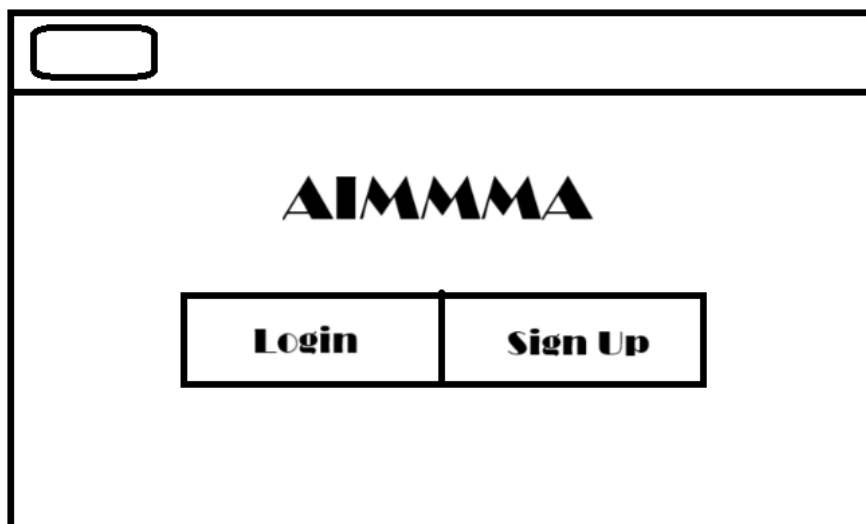


Figure: Components Diagram

4.4 User Interface Design



Start Session

Show Insights

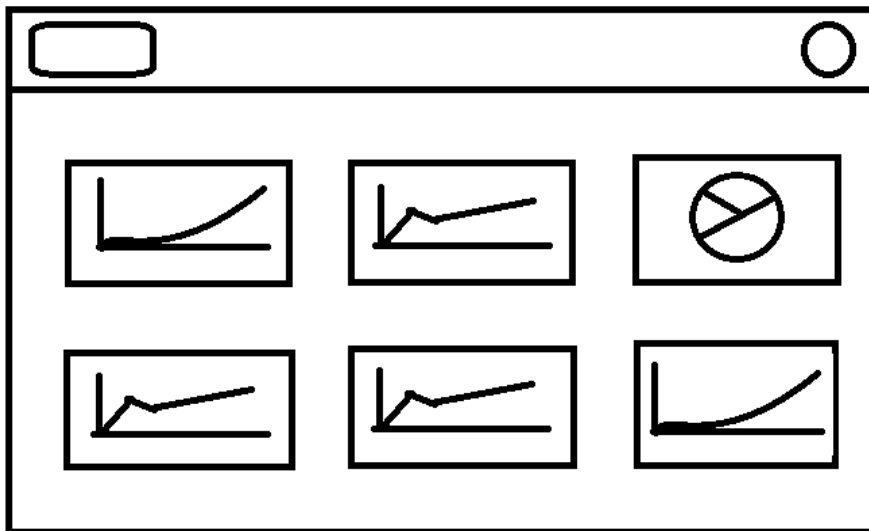
camera screen here

Make a choice if the camera is placed appropriately

Rep

Set

Tempo: Exercise:



4.5 Security Issues

- Data collection and storage: AI-based fitness tracking applications collect a large amount of personal data, including body measurements, activity levels, and location data. This data can be used to track users' movements and habits, and it could potentially be misused or sold to third parties without users' consent.
- Data security: AI-based fitness tracking applications store a large amount of sensitive data, and they must be properly secured to protect it from unauthorised access, theft, or loss. If this data is compromised, it could be used for identity theft, fraud, or other malicious purposes.
- Model bias: AI-based fitness tracking applications use machine learning models to track and analyze user data. These models can be biased, which could lead to inaccurate or unfair results. For example, a model that is trained on data from primarily male athletes may not be able to accurately track the performance of female athletes.
- Feedback accuracy: AI-based fitness tracking applications provide users with feedback on their form and technique. However, the accuracy of this feedback may

vary depending on the quality of the application's AI models. If the feedback is inaccurate, it could lead to users developing bad habits or even injuring themselves.

4.6 Test Cases Design

Sr. No.	Test Case	Functionality	Expected
1	Authentication	Login, Logout	Show dashboard, Show homepage
2	Start Session	Recording, Analyzing	Show exercise label and count
3	Analyze and store exercise in Session #1	Audio Feedback	Play audio cues and store the analysis
4	Analyzed and store exercise in Session #2	Visual Feedback	Show visual cues and store the analysis
5	Stop Session	Returns Dashboard	Show Dashboard
6	Show Charts	Loads charts	Displays chart page
7	Analyze and store while training #1	Classify pose and store timestamps	Classifies the correct pose defined in schema design
8	Analyze and store while training #2	Analyze changes in pose and analyze exercise life cycles	Uses a classified stream of data and keeps track of current exercise rep and count, by automatically

			detecting the exercise.
9	Model accuracy #1	Classify at 3 feet	Classifies accurately > 90%
10	Model accuracy #2	Classify at 4 feet	Classifies accurately > 90%
11	Model accuracy #3	Classify at 5 feet	Classifies accurately > 90%
12	Model accuracy #4	Classify from Front	Classifies accurately > 90%
13	Model accuracy #5	Classify from Front-side	Classifies accurately > 90%
14	Model accuracy #6	Classify from Side	Classifies accurately > 90%