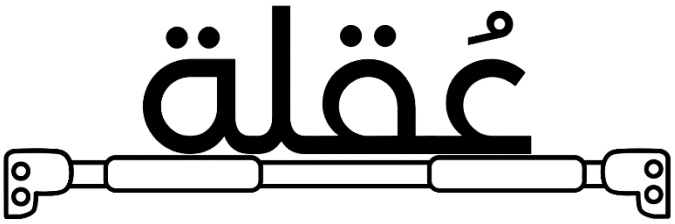
جامعة الملك سعود
King Saud University

كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

SWE321: Software Design & Architecture
2nd Semester 1445

Project

Course Code / Title: SWE 321
Semester / Year: 2nd semester/2024



Student Identification:

Section: 46474 **Group Number:** 6

Name: Norah Almubarak (Leader) **ID**: 443200845
Name: Lujain Albattah **ID**: 443200813
Name: Raghad Alotibi **ID**: 442200834
Name: Aljawharah Alotaibi **ID**: 443200509
Name: Yara Aljasir **ID**: 443200761
Name: Sarah Alshali **ID**: 443201001

Supervised By:
Dr. Abir Najjar & Dr.Rawan Alabdulrahman

Table of Contents

Revision History:

Version Number	Date	Description
1	30/1/2024	Corrected grammar mistakes
	6/2/2024	Adjusted some inconsistencies in the functional requirements
	15/2/2024	Added extra system context view.
2	2/3/2024	Corrected Phase 1 mistakes
	23/3/2024	Adjusted some inconsistencies in class diagram
	1/4/2024	Added extra user interfaces
3	20/4/2024	Corrected the Architecture Style and Domain Model
	27/4/2024	Added extra details to component diagram
	1/5/2024	Added extra details to class diagram

Updates:

Changes Done	The reason
Problem Domain Analysis, rewritten without anything related to the system	Since the problem domain should only discuss the problem
Solution Description, specify the technology used	To remove the ambiguity in the description
User Requirements, mention the payment	Since the payment was mentioned in the system requirements
Sstem Requirements, remove technical requirements	Technical Requirements should not be mentioned
Non-Functional Requirements, add model accuracy	Model Accuracy is critical to mention in the non-functional requirements
Architecture style, added extra controller	We needed the controller in component diagram
Domain Model, made it simpler	Because domain model suppose to represent the architecture in a abstract way

System Glossary:

Abbreviation	Description
Coach	A professional who provides guidance, instruction, and support to individuals seeking to improve their physical fitness and overall health.
Trainee	An individual who exercises
Users	Coaches and Trainees
AI	Artificial Intelligence is the simulation of human intelligence processes by computer systems.[1]
App	An Application is a software program that's designed to perform a specific function directly for the user or, in some cases, for another software program.[3]
Powerlifter	A person who lifts weights in three different ways, in a set order.[4]
Body Builder	Someone who regularly does special exercises to grow their muscles.[5]
Beginner	A person who is starting to exercise for the first time.[6]
OTP	One Time Password automatically generated numeric or alphanumeric string of characters that authenticates a user for a single transaction.[7]
MVC	Model View and Controller is a way to organize code's core functions into their own, neatly organized boxes.[8]
API	Application Programming Interface are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.[10]

ML	Machine learning is a discipline of AI that provides machines with the ability to automatically learn from data and past experiences while identifying patterns to make predictions with minimal human intervention.[11]
Supervised Learning	Supervised Learning, also known as supervised ML, is a subcategory of ML and AI. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately.[12]
Vay	Vay is motion analysis software that can be directly integrated into any application and works with any type of cameras to provide high-end motion analysis.[14]

1. Introduction

Nowadays, with health and fitness taking center stage in our lives, the demand for personalized and innovative solutions is more pronounced than ever. Uqla seamlessly connects trainees with certified coaches, all while utilizing AI analysis for detailed health insights to provide a holistic fitness experience. What sets Uqla apart is its integration of video recognition technology, a feature that actively monitors trainees' exercise sessions in real-time.

the **Purpose** of the app is to facilitate a personalized and interactive fitness coaching experience for trainees. It aims to connect individuals with qualified coaches, allowing them to receive customized workout plans, real-time feedback on their exercise form, and ongoing support for achieving their fitness goals. And enhancing the coaching process by providing coaches with valuable insights into the trainees' overall health and habits, enabling continuous modification and monitoring of their fitness plans. Overall, the app serves as a comprehensive platform for effective, customized fitness training and progress tracking.

The team members are intrigued by Uqla for its innovative solutions addressing significant challenges in the fitness coaching industry. The app's focus is on personalized training, integration of advanced technology like video recognition and health app data. Uqla's commitment to creating a cohesive and user-friendly fitness journey, along with the potential for scalability and positive impact on coaches, adds to the team's enthusiasm. Overall, Uqla is a compelling mission that fuels our collective passion for innovation, impact, and personal growth as it pushes the boundaries of health and fitness tech, exploring the exciting potential of AI and video recognition.

2. Problem Domain Analysis:

Individuals often struggle to find qualified fitness coaches who can deliver personalized training plans tailored to their specific goals, health conditions, and preferences. There is a notable challenge in creating workout plans that are truly aligned with an individual's unique requirements.

Moreover, there is a significant risk of serious injuries that can occur during fitness training, which needs to be more effectively mitigated. Communication between coaches and trainees is frequently restricted using regular messaging apps, which can lead to coaches feeling overwhelmed, particularly if they have a large number of trainees. This can intrude upon their personal communication with friends and family. Additionally, trainees may sometimes be reluctant to initiate communication or provide feedback, possibly due to concerns about being a nuisance.

Another issue is the lack of data-driven coaching; without comprehensive health and fitness data, it is difficult for coaches to make well-informed decisions regarding a trainee's progress and habits.

Finally, the fitness experience for many is often scattered, with issues in coordinating scheduling, exercise details, and progress tracking, which can impede the effectiveness of the fitness journey.

3. Solution Description:

Uqla app seamlessly connects users with certified trainers, utilizing ML supervised learning model for detailed health insights. Coaches create personalized workout and diet plans, with central progress tracking and integration with Health app and myFitnessPal for a comprehensive overview.

Uqla's standout feature is video recognition using computer vision integrated with the device's camera, this innovative feature actively monitors users' exercise sessions in real-time. As individuals engage in various exercises, the technology assesses their form by drawing virtual lines using Vay pattern recognition on key body parts, such as the back, arms, and legs.[14] Notably, these lines dynamically shift in color – red indicates incorrect form or misalignment, while a transition to green signals the achievement of proper posture. ensuring users maintain proper exercise form.

Users can also choose their coaches, engage with them directly, and subscribe with them seamlessly within the app. Uqla aims to redefine the fitness journey by providing a user-centric platform with personalized coaching, advanced analytics, and an intuitive interface.

4. The System Context View:

1. **Trainee choose a coach:**

The trainee opens the application and logs in, the trainee can browse through coaches, and he can search for a specific coach or filter them based on the coach's specialty and experience, then he can choose between them, after the coach accept the trainee he can pay the subscription fee, the trainee will have the schedule available to him ones the coach posts it.

2. **Trainee starts the workout plan:**

When opening the schedule, the trainee will have each exercise with details provided be the coach and video for clarification, the trainee can open the 'check your form' option which monitors users' exercise sessions in real-time by drawing virtual lines on key body parts and help them maintain of proper posture, he can record the exercise and send it to the coach to have more feedback, and under every workout there will be a note section in which he can add his note or complains about the exercise for the coach.

3. **Coach enroll trainees:**

The coach opens the application and logs in, in the home page the trainee's requests will appear in a list, the coach can browse through them and choose to accept the trainee according to their answers to the coach's form their category "beginner, powerlifter, bodybuilder", after accepting the trainee the coach will wait for the payment to be completed, after that he can start to create the workout plan in the app.

4. **Coach follow up trainee's progress:**

The coach will have each trainee in a separate page, when a trainee interacts with the workout sessions like adding a note or sending a record of the exercise the coach can add his comments, also at the end of each week/month the follow up session will appear in the trainees page in which the coach will Review the trainee's progress and determine what they would like to focus on and make adjustments accordingly.

5. Specific Requirements:

5.1 Functional Requirements

5.1.1 Trainee Requirements:

5.1.1.1. The trainee shall be able to search for coaches based on their specialty, name, and availability.
5.1.1.2. The trainee shall be able to view the coach's information.
5.1.1.3. The trainee shall be able to send a coaching request.
5.1.1.4. The trainee shall be able to view his personalized workout plan.
5.1.1.5. The trainee shall be able to track his progress towards fitness goals.
5.1.1.6. The trainee shall be able to view each exercise.
5.1.1.7. The trainee shall be able to view detailed instructions for each exercise.
5.1.1.8. The trainee shall be able to exercise using the form recognition for each exercise.
5.1.1.9. The trainee shall be able to record their exercise.
5.1.1.10. The trainee shall be able to request a live session with the coach.
5.1.1.11. The trainee shall be able to view a live session.
5.1.1.12. The trainee shall be able to track his health information.
5.1.1.13. The trainee shall be able to make payment.
5.1.1.14. The trainee shall be able to review his coach.

5.1.2 Coach Requirements:

5.1.2.1. The coach shall be able to view trainee requests.
    5.1.2.1.1 The coach shall be able to accept trainee requests.
    5.1.2.1.2 The coach shall be able to reject trainee requests.
5.1.2.2. The coach shall be able to view trainee live sessions requests.
    5.1.2.1.1 The coach shall be able to accept trainee live sessions requests.
    5.1.2.1.1 The coach shall be able to reject trainee live sessions requests.
5.1.2.3. The coach shall be able to view trainee profile information.
5.1.2.4. The coach shall be able to create workout plans for each trainee.
5.1.2.5. The coach shall be able to update the trainee workout plans.
5.1.2.6. The coach shall be able to view his trainee's workout plans.
5.1.2.7. The coach shall be able to conduct follow-up live sessions.
5.1.2.8. The coach shall be able to track the trainee's progress.
5.1.2.9. The coach shall be able to view his reviews.
5.1.2.10. The coach shall be able to view trainee recordings.
5.1.2.11. The coach shall be able to view advice for the plan.

5.1.3 User Requirements:

5.1.3.1. The user shall be able to create an account.
5.1.3.2. The user shall be able to log into the system.
5.1.3.4. The user shall be able to log out of their account.
5.1.3.5. The user shall be able to edit their account.
5.1.3.6. The user shall be able to review the app.
5.1.3.7. The user shall be able to communicate through live sessions.
5.1.3.8. The user shall be able to reset their password.

5.1.4 System requirements:

5.1.4.1. The system shall be able to send a random password to a new user's phone number.
5.1.4.2. The system shall approve payment before allowing the coach to initiate personalized training plans.
5.1.4.3. The system shall be able to analyze a trainee's exercise form.
5.1.4.4. The system shall allow trainees to write their feedback at the end of their program.
5.1.4.5. The system shall be able to assist coach with planning decisions

5.2 Non-Functional Requirements

5.2.1. The system shall provide workout plans for trainees in less than 5 seconds. *(performance)*
5.2.2. The user shall learn how to use the system's core features in less than 10 minutes. *(usability)*
5.2.3. The system shall be available 99.0% of the time. *(reliability)*
5.2.4. The system shall be able to accommodate up to 40,000 users. *(scalability)*
5.2.5. The system shall support both Arabic and English languages. *(localizability)*
5.2.6. The system shall be able to successfully integrate with external systems, such as the health app and MyFitnessPal with a data loss rate of less than 1%. *(integrability)*
5.2.7. The system shall achieve at least 80% accuracy rate for supervised learning model in assessing the trainee's exercise. [13] *(Accuracy)*

5.3 Design Constraints

5.3.1. The system shall integrate with Health app and MyFitnessPal.
5.3.2. The system shall be developed as a mobile app.
5.3.3. The system shall support iOS platform.
5.3.4. The system shall enable online transaction by integrating with Apple pay.

6. Challenges:

We acknowledge that challenges are inherent in any project, and being prepared to address them is crucial for success. Identifying potential hurdles allows us to proactively navigate them and enhance the overall project outcome.

A potential challenge that we may face is Certification and Quality Assurance, ensuring that these coaches meet quality standards and are adequately certified could be a challenge.
Also given that fitness apps deal with personal health data and payment, ensuring robust data security and privacy measures is essential. Any breaches or concerns about data security can lead to a loss of user trust.

Another challenge revolves around determining the optimal architecture style and overall system structure for Uqla, the architectural decision-making process is important to the app's performance, scalability, and overall success.

Despite these challenges, we are committed to overcome any obstacles that we may face and learn from each experience. By acknowledging and addressing these potential hurdles, we aim to strength our development process, ensuring the delivery of a robust and user-friendly fitness app that seamlessly integrates AI technologies and meets the evolving needs of our users.

7. Projection:

By the end of this semester, our goal is to develop a system that effectively fulfills the specified functionalities while maintaining a high standard of quality. We aim to carefully select the most suitable architecture to implement the system, ensuring optimal performance and scalability.

The functions and the tasks of the system that is expected to be accomplished.

- Coach-Trainee Matching: Facilitate the process of connecting trainees with qualified coaches based on their specific fitness goals, preferences, and availability.
- Customized Workout Plans: Enable coaches to create personalized workout plans for trainees, considering their fitness level, goals, and any specific requirements or limitations.
- Real-time Exercise Form Feedback: Provide a mechanism for trainees to receive real-time feedback on their exercise form, leveraging technologies such as video analysis or motion tracking.
- Communication and Support: Offer interactive communication channels, such as messaging or follow-up live sessions, to facilitate ongoing support and guidance between coaches and trainees.
- Progress Tracking: Enable trainees to track their progress, including workout completion, weight or measurement changes, and other relevant metrics.
- Data Analytics and Insights: Provide coaches with valuable insights into trainees' overall health, habits, and progress through data analytics. This can help coaches make informed decisions and modify fitness plans as needed.
- Exercise Library: Provide a comprehensive library of exercises with detailed instructions, videos, and variations to assist trainees in performing exercises correctly and efficiently.
- Notifications and Reminders: Send notifications and reminders to both coaches and trainees for upcoming sessions, progress updates, or other important events.

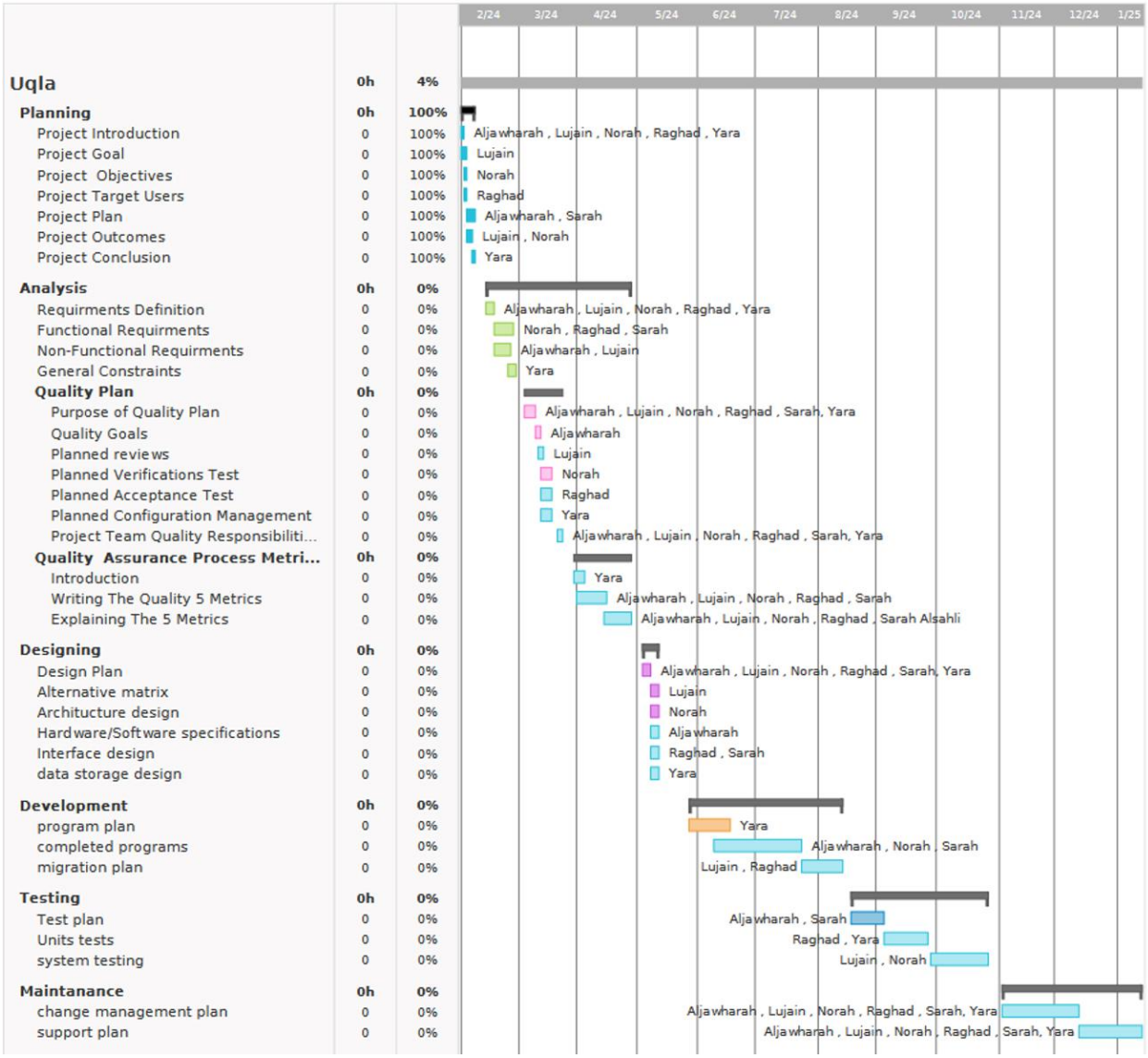


*Figure7.1: this figure shows **Gantts chart** execution plan for Uqla.*

8. Methodology:

After a thorough analysis of our system from the perspectives of business requirements, project objectives, and user needs, and considering our architecture, we decided that **Agile Methodology** is the suitable methodology for Uqla App.

The reasons behind our decision to adopt the Agile methodology stem from our app's requirement for continuous adaptation to market trends and user preferences.

A critical factor in our choice is the need for user feedback after each sprint, aiming to achieve customer satisfaction and enhance our app's adaptability.

Agile's approach of dividing the project into short, manageable sprints grants our team the agility to promptly incorporate and implement changes.

Therefore, we will proceed with multiple sprints, each encompassing four phases for every component: Plan, Analysis, Design, implementation, testing, and Deployment. [17]

For **Uqla** System, integrating with Health and MyFitnessPal is a key feature, but it may also present challenges in fetching data from these apps. Additionally, integrating Vay for form recognition introduces a higher level of complexity. Ensuring accuracy as well as privacy for each trainee. Another challenge is creating live sessions. Creating live sessions requires a strong backend infrastructure to make sure that there is no latency. So, our team will take enough time to ensure these things.

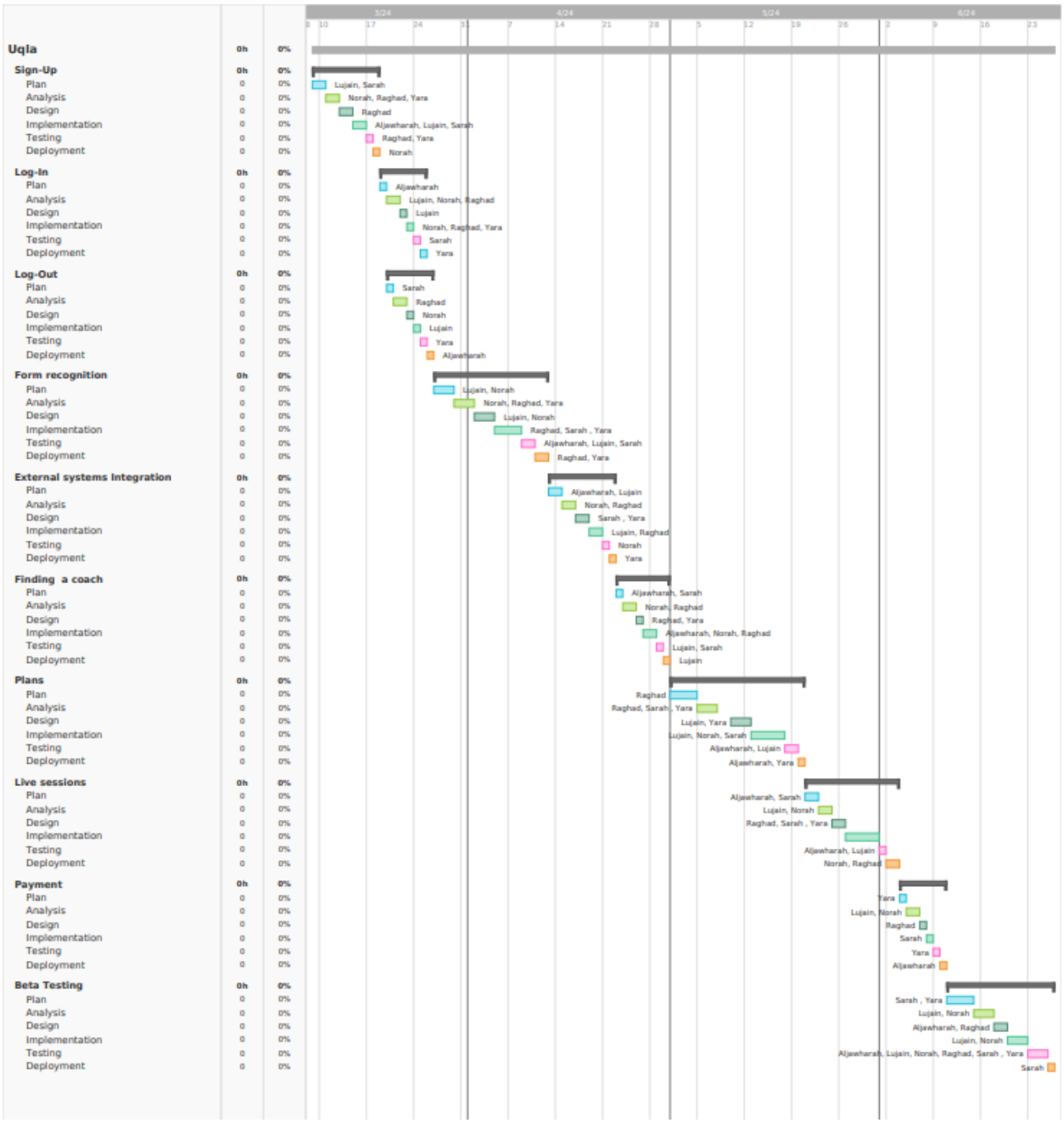


*figure 8.1 this shows the plan for Uqla using **Gantt chart***

Iteration number	Module/Subsystem	duration
1	Sign-Up, Log-In, Log-Out	**18 days**
2	Form recognition, External systems Integration	**27 days**
3	Finding a coach, Plans, Live sessions, Payment	**49 days**
4	Beta Testing	**16 days**

*Table 8.2 this table shows **iterations modules plan along estimated durations** For Uqla.*

9. System Architecture:

9.1. Design Decisions:

Design Issue 1: How will Uqla securely store the users' data?

A safe and expandable cloud storage option like Amazon S3 (Simple Storage Service) would be used to store user data in Uqla. Redundancy and data security are included into the robust and highly available object storage that Amazon S3 provides. To guarantee the privacy and accuracy of user data, it offers encryption both in transit and at rest. To further limit data access based on user responsibilities and permissions, access control measures will be put in place. [22]

Design Issue 2: How will Uqla help fitness instructors and trainees communicate and provide feedback in real time?

Fitness instructors and trainees will be able to communicate easily thanks to Uqla's user-friendly interfaces. Using Real-time live session.

Design Issue 3: How does Uqla create individualized training programs for each user based on their goals, medical history, personal preferences, and historical data To provide personalized training programs for each user, Uqla will leverage machine learning and automated technologies. The system will provide training regimens tailored to each user's requirements by taking into account variables including personal goals, body changes, health problems, and past performance information. By providing individual training, this option seeks to increase user motivation and engagement by providing all this information and helping the trainee to develop.

Design Issue 4: How will Uqla use the external systems to deliver a complete fitness experience across the board?

Trainee exercise logs, nutritional information and health indicators will be accessed and shared securely by integrating with the Health app and MyFitnessPal. Users will benefit from this integration by having a unified system, also by providing tips generated using Open AI to the coach while making fitness plan, these tips is based on the informations provided from the external system.

Design Problem 5: How will Uqla ensure scalability and performance to handle a large user base and growing demand?

Uqla will be designed to handle a large user base and growing demand. The system will leverage scalable cloud infrastructure and distributed computing technologies (external systems) to ensure high performance and responsiveness. This system will enable us to be able to support multiple users and provide a user-friendly and accessible experience even at the peak of the program.

9.2. Domain Model:

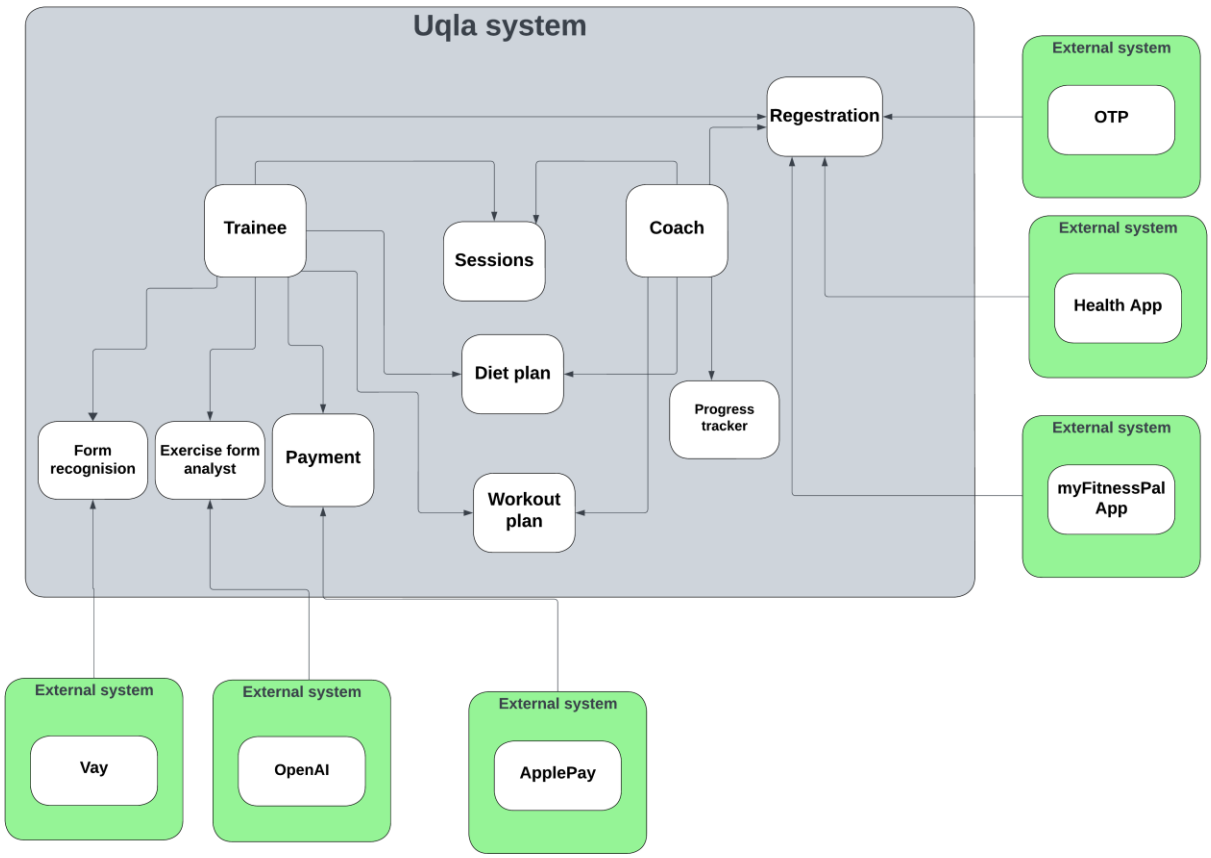


*Figure9.2.1.: this figure shows the **domain model** for Uqla.*

Uqla Components:

- **Registration:** The component which is responsible for signing up/ logging in users.
- **Trainee:** Individual who seeks fitness guidance and searches for a coach.
- **Coach:** Certified individual who creates workout plans and monitor trainee's progress.
- **Sessions:** The component that represents periodic follow up to assess trainee's progress.
- **Payment:** The component that represents the financial transactions.
- **workout plans:** The component that represents the exercise plan for trainee which the coach creates.
- **diet plans:** The component that represents the diet plan for trainee which the coach creates.
- **Health insights:** The component represents the blackboard that provides knowledge source, alerting and providing feedback on incorrect exercise forms during workout sessions.

- **Progress tracker:** The component responsible for tracking and visualizing a trainee's progress over time, including metrics such as weight loss/gain, muscle mass changes, and improvements in fitness levels.
- **Exercise form analyst:** The component that constantly monitors a trainee's exercise form during workout sessions in real-time, offering instant feedback and suggestions for improvement to ensure proper technique and minimize the risk of injury.
- **Form recognition:** The component that draws virtual lines on key body parts during exercise sessions to provide real-time feedback on form and alignment.
- **Health App External System:** The component that provides trainee's health information for analysis.
- **MyFitnessPal App External System:** The component that provides trainee's diet information for analysis.
- **OTP External System:** The component that provides numeric or alphanumeric string of characters that authenticates a user for a single transaction.[7]
- **Apple Pay External System:** This component provides a secure payment system.
- **OpenAI External System:** The component that integrates with OpenAI's technology for advanced AI analysis, providing additional insights and capabilities to the Uqla platform.
- **Vay External System:** The component that integrates video recognition technology into the app, allowing for real-time analysis of exercise form and technique during workout sessions.

9.3. Architectural Style:

For Uqla, we've chosen to adopt the **Interaction Oriented Model-View-Controller (MVC)** architectural style, which is an established design pattern that delineates an application into three principal logical components: Model, View, and Controller. This pattern offers several advantages that align well with the goals and requirements of Uqla.

The model will encapsulate the data and the business rules pertinent to the system. It will process data logic and interact with the database for data storage and retrieval. This will form the backbone of Uqla's operations, handling the core functionalities and data manipulations.

The view will be responsible for presenting data to the user in a structured and coherent manner. It will be the interface through which users interact with Uqla, ensuring that information is displayed effectively, and user input is captured accurately.

The controller will serve as the intermediary that connects the view and the model, facilitating communication between the user interface and data model. It will interpret user inputs and translate them into actions to be performed by the model.

The adoption of the MVC pattern is intended to support Uqla's scalability and maintainability. By separating concerns, Uqla's codebase becomes more organized and manageable, which is crucial as the system evolves. This separation of concerns also enables individual components to be modified or extended without impacting others, promoting flexibility in development, and simplifying maintenance. It will also assist in reducing code redundancy by abstracting the business logic and data from the user interface.[15]

During the initial design phase for Uqla, we initially considered the client-server architecture, but we decided against it for multiple reasons. This architecture segments the workload between service requesters (clients) and service providers (servers). A key concern is the potential for slow performance, primarily due to the constant back-and-forth communication between clients and servers. This can lead to increased wait times as data travels across the network. Additionally, the servers may become overloaded with requests, creating a bottleneck that can further degrade system responsiveness.

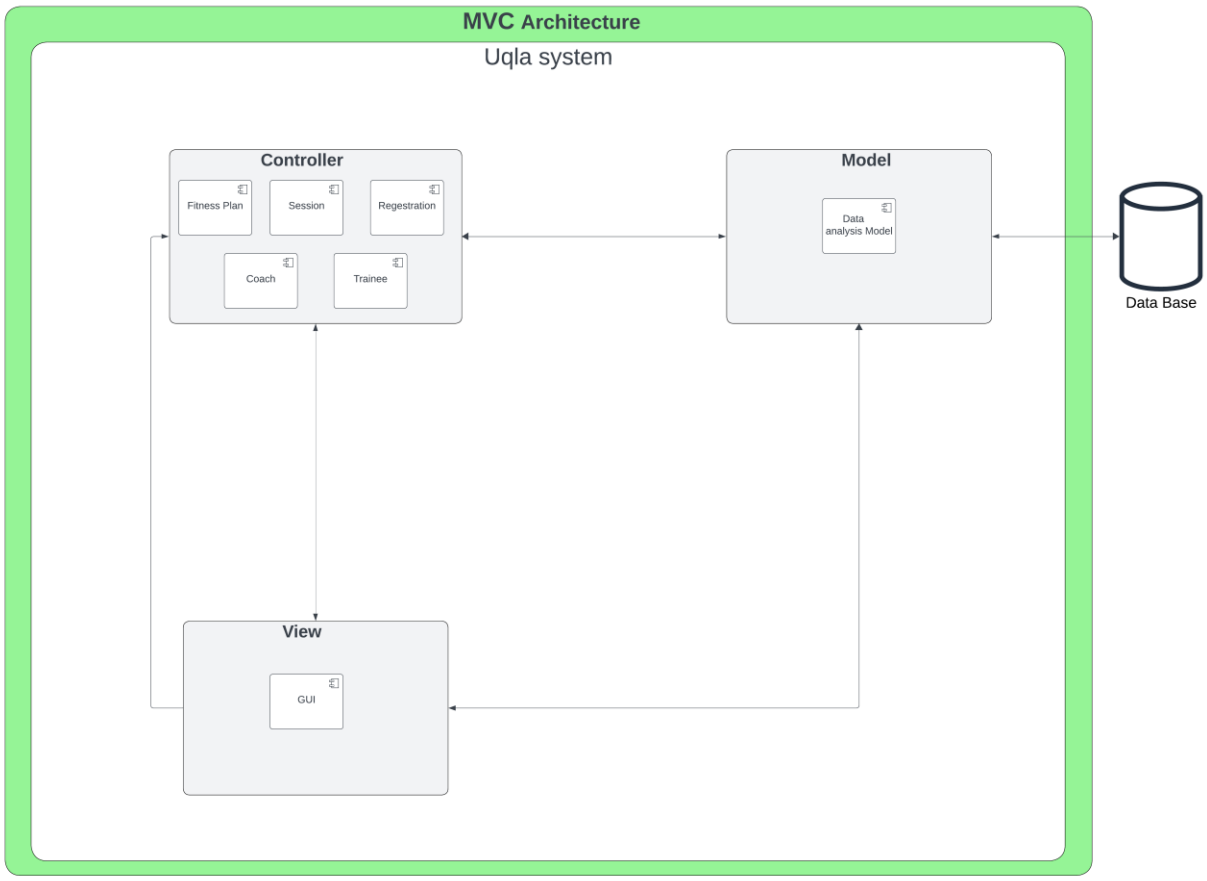


*Figure9.3.1: this figure shows the **high-level diagram model** for Uqla*

9.4. Structural Model:

9.4.1. Class Diagram

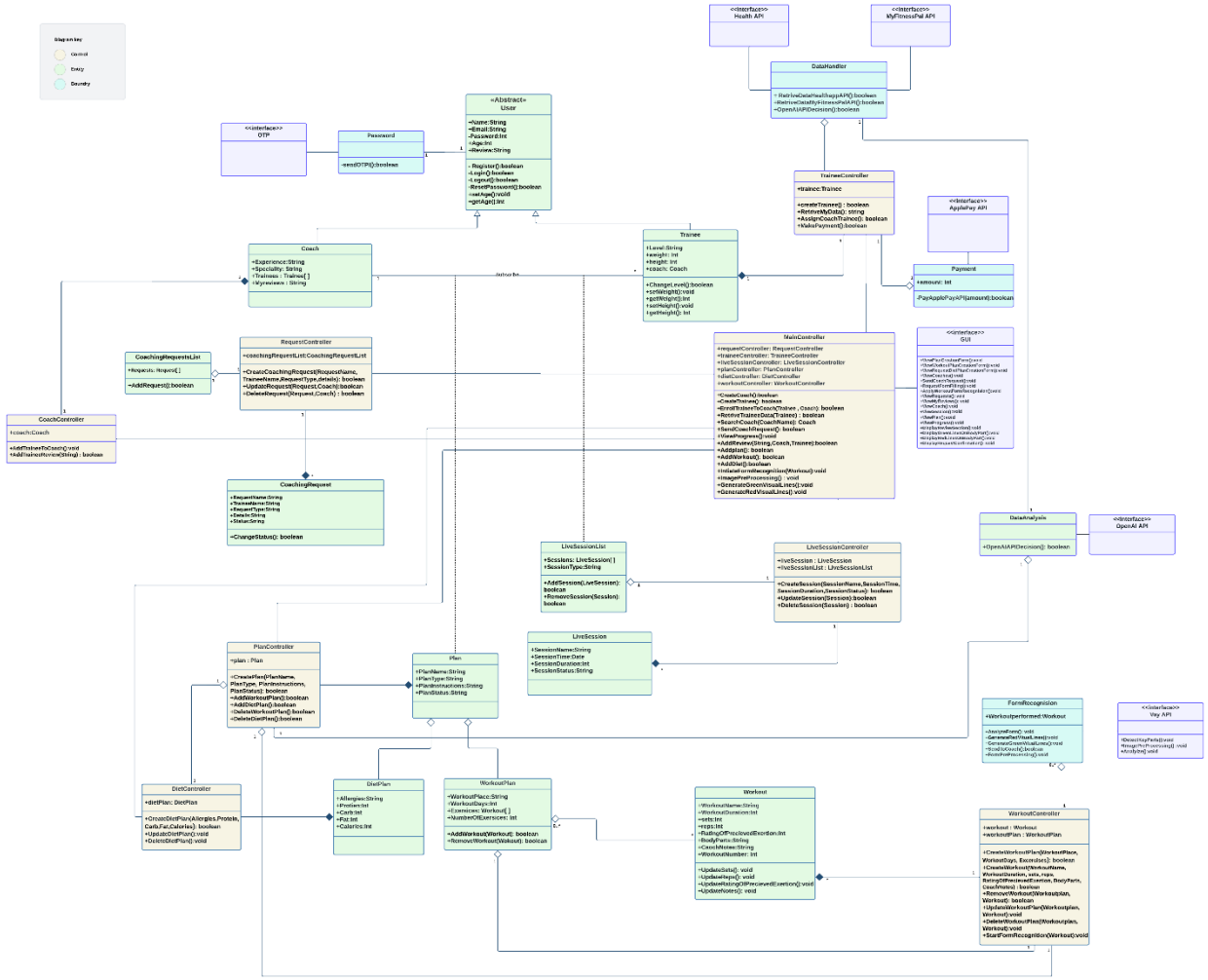


*Figure9.4.1: this figure shows the **class diagram** for Uqla, for clearer vision* **Click here**.

Note: Rating of Perceived Exertion (RPE) attribute in Workout class is referring to is a way to measure the intensity of workout.

9.4.2. Component Diagram:

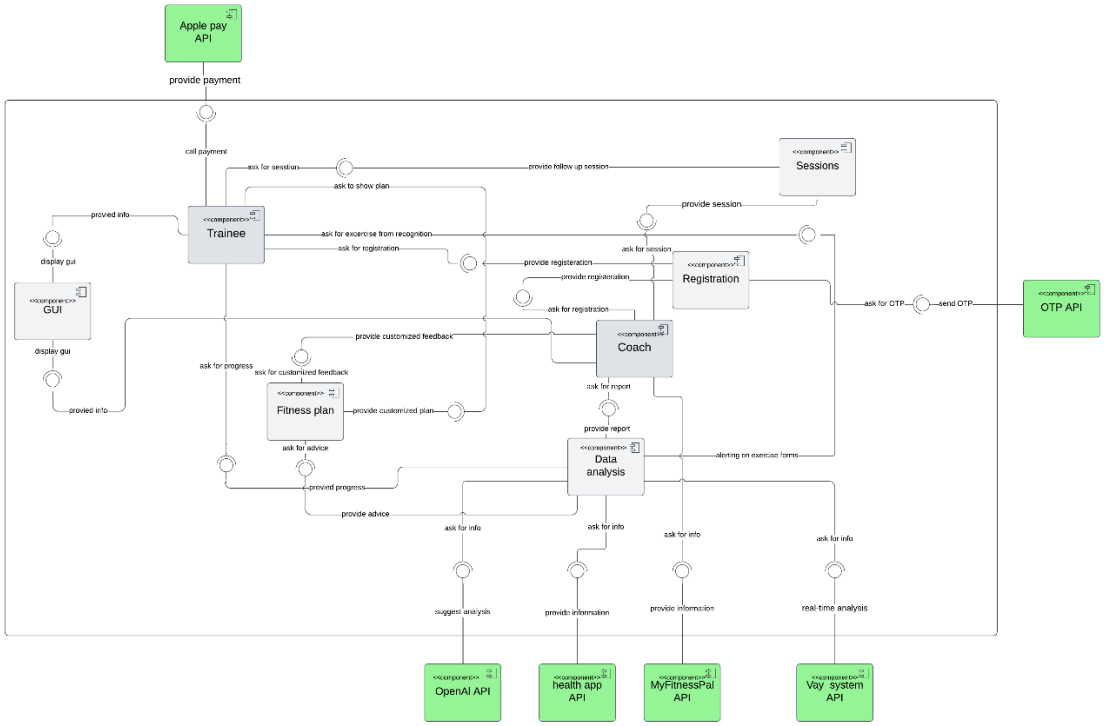


*Figure9.4.2: this figure shows the **component diagram** for Uqla, for clearer vision **Click here.***

COMPONENT NAME	Fitness plan
DESCRIPTION	- The Fitness Plan component manages the structuring and customization of workout and diet plans for trainees. It facilitates communication between trainees and coaches, ensuring personalized fitness plans and feedback mechanisms.
FUNCTIONALITY/BEHAVIOR	- **Plan Customization:** Tailors workout and diet plans to individual trainee goals and capabilities. - **Coaching Interaction:** Manages coaching requests and feedback loops between trainees and coaches. - **Controller Coordination:** Ensures synchronization and execution of planned activities through coordination with the main controller.
CLASSES	- **WorkoutPlan Class:** Structures comprehensive fitness plans with multiple workouts. - **CoachingRequest Class:** Represents trainee requests for modifications or feedback on fitness plans. - **CoachingRequestList Class:** Manages a list of coaching requests for proper handling and response. - **RequestController Class:** Coordinates communication between coaching requests and the main controller. - **Diet Plan Class:** Incorporates dietary components into fitness plans, providing nutrition guidelines. - **Plan Controller:** Manages interactions between fitness plans and the main controller. - **Diet Controller:** Handles communication between the diet plan and the main controller.

	- **Workout Controller:** Facilitates communication between the workout plan and the main controller. - **Plan Class:** Represents structured fitness plans combining workouts and diet guidelines.

COMPONENT NAME	GUI
DESCRIPTION	- The GUI component handles the graphical user interface for the application. It provides a user-friendly interface for users to interact with the system and access its functionalities.
FUNCTIONALITY/BEHAVIOR	- **Manages the display and layout of graphical elements**, such as windows, buttons, menus, and forms. - **Handles user input and events**, such as mouse clicks and keyboard interactions. - **Communicates with other components and controllers** to retrieve and display relevant information. - **Provides visual feedback and updates** to the user based on system responses and actions.
CLASSES	- **Main Controller Class:** The GUI component may rely on the Main Controller class to coordinate and manage the sub-controllers in the system. It communicates with the Main Controller to retrieve data and perform actions based on user interactions. - **GUI class:** Handles the graphical user interface for the application.

COMPONENT NAME	Trainee
DESCRIPTION	- This component manages trainee users in the system, including their training, progress, and payment function
FUNCTIONALITY/BEHAVIOR	- The Trainee Component **handles the management of trainee users** within the system. It includes attributes and methods related to their training and progress.
CLASSES	- **Trainee Class:** Represents trainee users and stores their personal information, - **Payment Class:** Handles payment processing for trainees using the Apple Pay API. - **TraineeController Class**: is responsible for receiving trainee requests, relaying them to the appropriate components, and providing responses back to the trainee.

COMPONENT NAME	Apple Pay API
DESCRIPTION	- The Apple Pay API Component enables secure payment transactions within the system by integrating with the Apple Pay API
FUNCTIONALITY/BEHAVIOR	- **API Integration:** Integrates with the Apple Pay API to enable payment transactions within the system. - **Secure Transactions:** Ensures security and encryption of payment data to protect user information.
CLASSES	- **Apple Pay API Class:** Interfaces with the Apple Pay API to facilitate secure payment transactions within the system.

COMPONENT NAME	OpenAI API
DESCRIPTION	- The OpenAI API Component harnesses the power of natural language processing and other artificial intelligence capabilities provided by OpenAI.
FUNCTIONALITY/BEHAVIOR	- **API Integration:** Integrates seamlessly with the OpenAI API to access its suite of natural language processing and AI capabilities.
CLASSES	- **OpenAI API Class:** Interfaces with the OpenAI API to leverage its natural language processing and AI capabilities for various functionalities within the system.

COMPONENT NAME	Sessions
DESCRIPTION	- The component that represents periodic follow-up with the coach to assess trainee's progress.
FUNCTIONALITY/BEHAVIOR	- **Receive the live session request** from the trainee. - **Schedule the live session** with the coach in live sessions trainee list.
CLASSES	- **liveSessionList Class:** Represents a list of live training sessions for a trainee. - **liveSession Class:** Represents One live training session. - **LiveSession Controller:** coordinate between the live sessions and main controller

COMPONENT NAME	MyFitnessPal API
DESCRIPTION	- The component that provides trainee's diet information for analysis.
FUNCTIONALITY/BEHAVIOR	- **Retrieve information** on nutrition, calorie tracking. - **Provide information** for trainee fitness plans.
CLASSES	- **MyFitnessPal API Class:** Integrates with the MyFitnessPal API to retrieve information on nutrition, calorie tracking, and provide information for trainee fitness plans.

COMPONENT NAME	Data analysis
DESCRIPTION	- The component provides real-time monitoring of a trainee's exercise form, offering instant feedback and suggestions for improvement. It draws virtual lines on key body parts during workout sessions taking it from Vay API, to ensure proper technique and alignment, minimizing the risk of injury.
FUNCTIONALITY/BEHAVIOR	- **Analyzes a trainee's exercise form** during workout sessions. - **Offers immediate feedback on form and alignment** by drawing virtual lines on key body parts. - **Provides suggestions** for improvement to ensure proper technique and minimize the risk of injury. - **Provides suggestions** for enhancing the fitness plan based on trainee's data.
CLASSES	- **FormRecognition Class:** Apply various vascular lines. - **DataHandler Class:** retrieve the data form the external systems and analysis it. - **Data analysis:** retrieve decision from data handle to produce open ai decision.

COMPONENT NAME	Vay system API
DESCRIPTION	- The component that integrates video recognition technology into the app, allowing for real-time analysis of exercise form and technique during workout sessions.
FUNCTIONALITY/BEHAVIOR	- **Human motion analysis** - **Analyze** all possible angles. - Accurate **range of motion** - **Mistake detection.** - **Reptation counting**
CLASSES	- **VayAPI Class:** Integrates with the Vay system API to access and utilize specific functionalities or services provided by the Vay system.

COMPONENT NAME	Registration
DESCRIPTION	- The component which is responsible for signing up/ logging in users.
FUNCTIONALITY/BEHAVIOR	- **Allow users (trainee and coach)** to sign up and login using their information as name, email, password
CLASSES	- **User class:** Provide information about the user and the coach. - **Password class:** To handle OTP API.

COMPONENT NAME	OTP API
DESCRIPTION	- Authentication way if the user forgets or want to reset the password.
FUNCTIONALITY/BEHAVIOR	- Concerning the **creation and verification of OTP** for managing the password
CLASSES	- **OTP Class:** Provides functionality related to generating and validating OTP for secure user authentication.

COMPONENT NAME	Coach
DESCRIPTION	- This component is crucial for taking data from the Data Analysis and for delivering the trainee personalized feedback, including edits and plan directions, as well as live sessions.
FUNCTIONALITY/BEHAVIOR	- **Custom fitness plan management** - **Providing feedback**
CLASSES	- **Coach Class:** Represents a coach user in the system, with attributes and methods related to coaching trainees. - **Coach Controller:** Coordinate between the coach and main controller.

COMPONENT NAME	Health app API
DESCRIPTION	- This component is the one who provides the information for data analysis component.
FUNCTIONALITY/BEHAVIOR	- **Provides trainee's health information** for analysis.
CLASSES	- **Health AppAPI Class:** Integrates with external health tracking apps or devices to fetch health-related data for trainees.

9.4.3. Deployment Diagram:

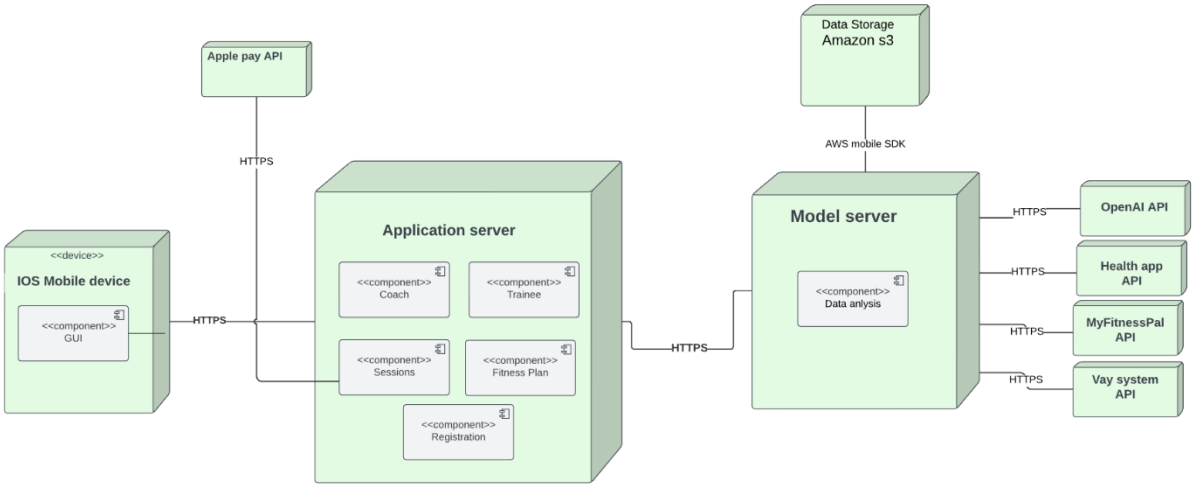


*Figure9.4.3.: this figure shows the design level **deployment diagram** for Uqla, for clearer vision* **Click here**.

9.5. Behavioral Model:

9.5.1. Use Case Diagram

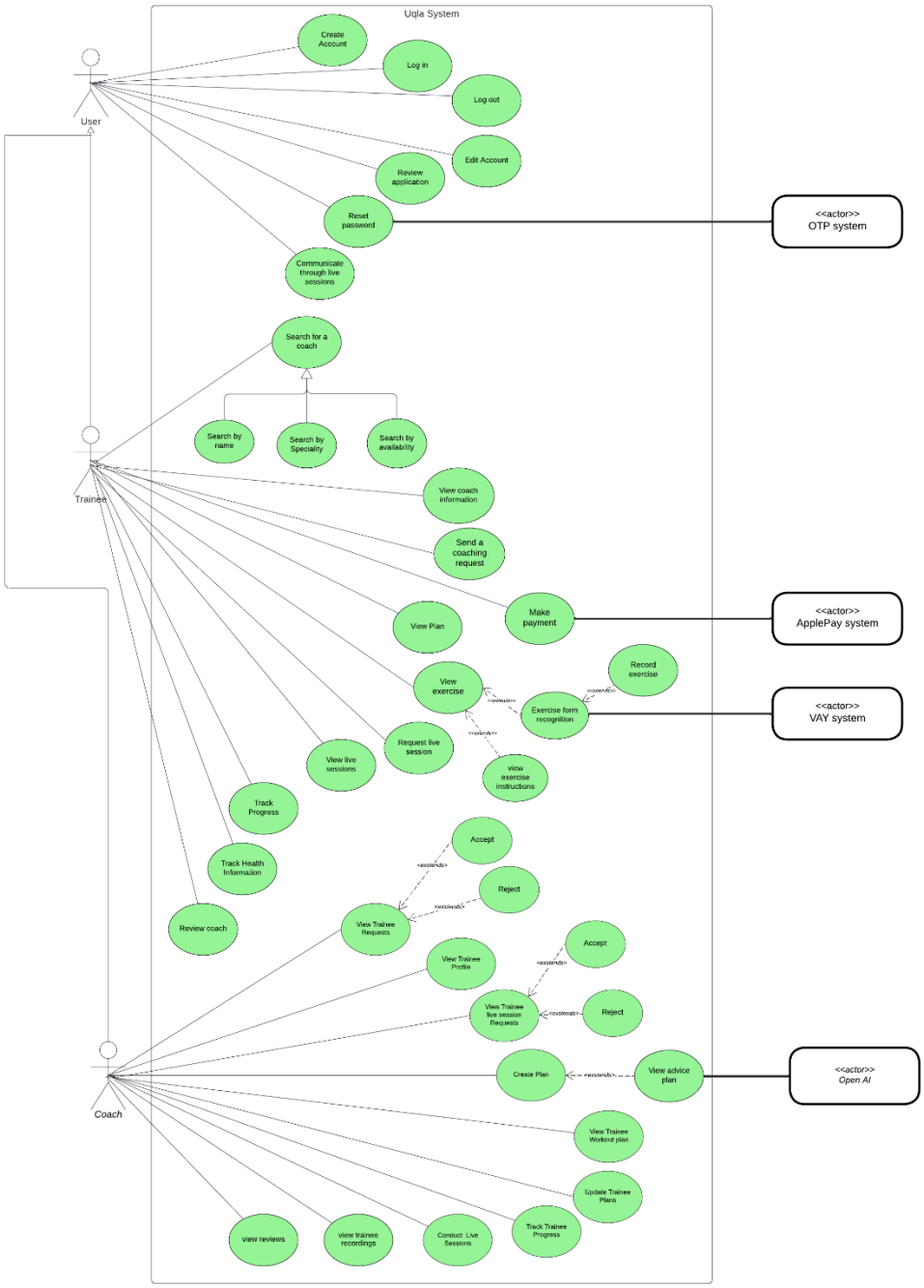


*Figure9.5.1: this figure shows the **use case diagram** for Uqla, for clearer vision* **Click here.**

9.5.2. Use Case Description:

9.5.2.1. Create a plan:

Use Case Description	
System: Uqla	
9.5.2.1.1. Use Case Name: Create a plan	
9.5.2.1.2. Description: This use case describes how a registered coach creates a plan for the trainee.	
9.5.2.1.3. Actors	
Primary Actor: Coach	**Other actors:** Health app, MyFitnessPal app
Stakeholders: Trainee	
9.5.2.1.4. Relationships: **Includes:** None **Extends:** View plan's advicer	
9.5.2.1.5. Pre-Conditions: 1. The trainee must have sent a plan request. 2. The coach must have a registered coach account. 3. The coach must be logged in successfully. 4. The coach must select the trainee from his/her trainees list.	
9.5.2.1.6. Steps	
Actor (Coach)	**System**
1. The coach selects the "Create a plan "option.	
	2. The system displays a form for the coach where he/she can enter plan name, plan type, plan instructions, and plan status for the trainee. **3.** The system requests the coach to fill the form.
4. The coach fills the form by specifying plan name, plan type, plan instruction and plan status.	
	5. The system presents health and fitness data obtained from the Health app. **6.** The system presents health and fitness data obtained from the MyFitnessPal app. **7.**The system displays a "show AI advices" or "skip" options to the coach.
8. The coach choose "skip" option.	
	9. The system displays a form for the coach where he/she can enter workout's plan place, workout days, and exercises for the trainee. **10.** The system requests the coach to fill the form.

11. The coach fills the form by enter workout's plan place, workout days, and exercises. **12.** The coach adds workouts.	
	13. The system displays workout plan information and asks coach to press "Confirm Workout Plan".
14. The coach confirms the workout plan.	
	15. The system displays a form for the coach where he/she can enter allergies, protein, carb, fat, calories for the trainee. **16.** The system requests the coach to fill the form.
17. The coach fills the form by specifying allergies, protein, carb, fat and calories.	
	18. The system displays diet plan information and asks coach to press "Confirm Diet Plan".
19. The coach confirms the diet plan.	

9.5.2.1.7. Alternative and Exceptional flows:

1. **The coach doesn't press "Confirm Workout Plan"**

if in Step 11, coach doesn't press" Confirm Workout Plan":

1. The system saves the form in the draft files.
2. Steps 7-12 are repeated to allow the coach to revisit and confirm the workout plan.

2. **The coach doesn't press "Confirm Diet Plan"**

if in Step 16, coach doesn't press" Confirm Diet Plan":

3. The system saves the form in the draft files.
4. Steps 13-17 are repeated to allow the coach to revisit and confirm the workout plan.

3. **Health app data retrieval failure**

if, in Step 5, the system encounters a failure in retrieving health s data from the Health app:

1. The system notifies the coach about the data retrieval issue.
2. The coach proceeds to create the workout plan without the additional health and fitness data.

4. **MyFitnessPal data retrieval failure**

if, in Step 6, the system encounters a failure in retrieving fitness s data from the MyFitnessPal app:

1. The system notifies the coach about the data retrieval issue.
2. The coach proceeds to create the workout plan without the additional health and fitness data.

5. **Missing field/ Incorrect format**

5.1 if,in Step 4, the coach misses a required field or enter incorrect format:

1. The system displays a message indicating required field or format.
2. Step 4 is resumed

5.2 if, in Step 9, the coach misses a required field or enter incorrect format:

1. The system displays a message indicating required field or format.
2. Step 9 is resumed

5.3 if, in Step 15, the coach misses a required field or enter incorrect format:

1. The system displays a message indicating required field or format.

2. Step 15 is resumed **6. The coach choose "show AI advices" option** if, in Step 8, the coach choose "show AI advices" option: 1. The view advice plan will be executed.
9.5.2.1.8 Post Conditions: • **Successful Condition:** The system successfully adds the workout plan to the trainee's workout plan list. • **Fail Condition:** No Workout Plan is made for the trainee.

*Table9.5.2.1.: this table shows the use case description for **creating a workout plan** for Uqla*

9.5.2.2. Send coaching request:

Use Case Description	
System: Uqla	
9.5.2.2.1. Use Case Name: Send coaching request	
9.5.2.2.2. Description: This use case describes how a registered trainee sends a coaching request for the coach.	
9.5.2.2.3. Actors	
Primary Actor: Trainee	**Other actors:**
Stakeholders:	
9.5.2.3.4. Relationships: **Includes:** None **Extends:** None	
9.5.2.3.5. Pre-Conditions: **1.** The trainee must be logged in successfully.	
9.5.2.2.6. Steps	
Actor (Trainee)	**System**
1. The trainee selects the "View list of coaches "option.	
	2. The system displays a list of coaches
2. The trainee selects a coach **3.** The trainee selects the "Send coaching request "option.	
	4. The system displays a form for the trainee where he/she can enter Request Name, Trainee Name, Request Type, details. **5.** The system requests the trainee to fill in the form.
6. The trainee fills in the form by specifying the Request Name, Trainee Name, Request Type, details.	
	7. The system displays request information and asks trainee to press "send"
	8. The system displays coaching request confirmation.
9.5.2.2.7. Alternative and Exceptional flows:	
1. The trainee doesn't press "send" If, in Step 9, trainee doesn't press "send": 1. The system saves the form in the draft files. 2. Steps 7-9 are repeated to allow the trainee to revisit and confirm the coaching request. **2. Missing field/ Incorrect format** If, in Step 6, the coach misses a required field or enter incorrect format: 1. The system displays a message indicating required field or format.	

2. Step 6 is resumed
9.5.2.2.8. Post Conditions: • **Successful Condition:** The system successfully sends the coaching request to the coach. • **Fail Condition:** No coaching request is sent to the coach.

*Table9.5.2.2.: this table shows the use case description for **viewing a workout plan** for Uqla*

9.5.2.3. View form recognition report:

Use Case Description	
System: Uqla	
9.5.2.3.1. Use Case Name: Exercise Form Recognition	
9.5.2.3.2. Description: This use case describes how a trainee can use form recognition when exercising.	
9.5.2.3.3. Actors	
Primary Actor: Trainee	**Other actors:** Vay
Stakeholders:	
9.5.2.3.4. Relationships: **Includes:** None **Extends:** None	
9.5.2.3.5. Pre-Conditions: 1. The trainee must be logged in successfully. 2. The coach must have created a Workout plan. 3. The trainee must access a specific exercise.	

9.5.2.3.6. Steps		
Primary Actor (Trainee)	**System**	**Secondary actor (Vay)**
1. Trainee selects the "Exercise using form recognition" option.		
	2. System opens the camera.	
3. Trainee perform the exercise.		
	4. System receives video streams and send it to Vay.	
		5. Vay analyzes the key parts placement and movement.
	6. System displays green visual lines on the correct body key part indicating correct form.	

9.5.2.1.7. Alternative and Exceptional flows:

- **Vay detected a wrong form:**
  If, in Step 6, Vay detected a wrong form:
  1. System displays red visual lines on the mistaken body key part indicating wrong form.
- **System couldn't open the camera:**
  If, in Step 2 system couldn't open the camera:
  1. System displays a message indicating camera is not available.

2. The use case ends with failure condition. • **Trainee doesn't perform the exercise:** If, in Step 3 trainee doesn't perform the exercise: 1. The use case ends with failure condition.
9.5.2.1.8 Post Conditions: • **Successful Condition:** The system successfully shows the trainee the form recognition lines. • **Fail Condition:** No form recognition lines visible to trainee.

*Table9.5.2.3.: this table shows the use case description **for viewing form recognition** report for Uqla*

9.5.3. Sequence Diagram:

9.5.3.1. Create a plan:

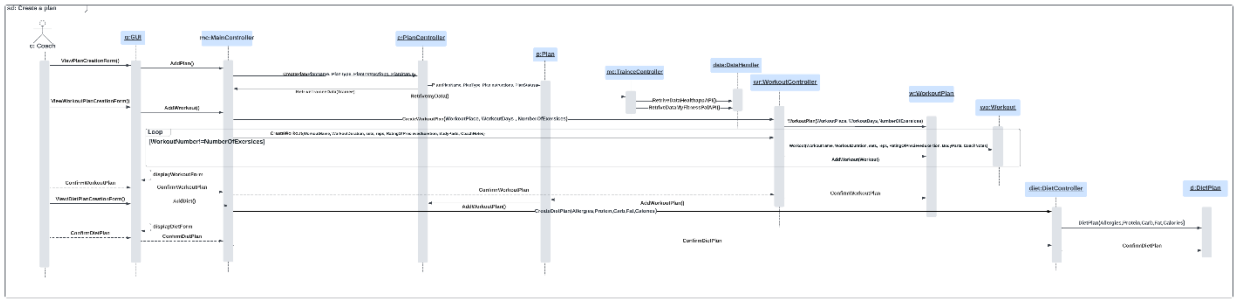


*Figure9.5.3.1: this figure shows the sequence diagram for **creating a workout plan** for Uqla for clearer vision* **Click here**.

9.5.3.2 Send coaching request:

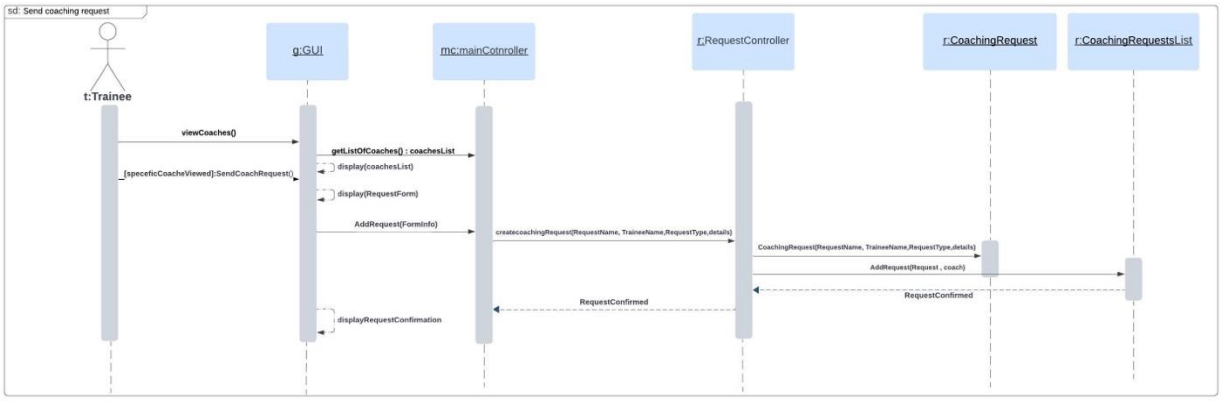


*Figure*9.5.3.2: this figure shows the sequence diagram for ***Send coaching request*** for Uqla *for clearer vision* **Click here**.

9.5.3.3 View form recognition report:

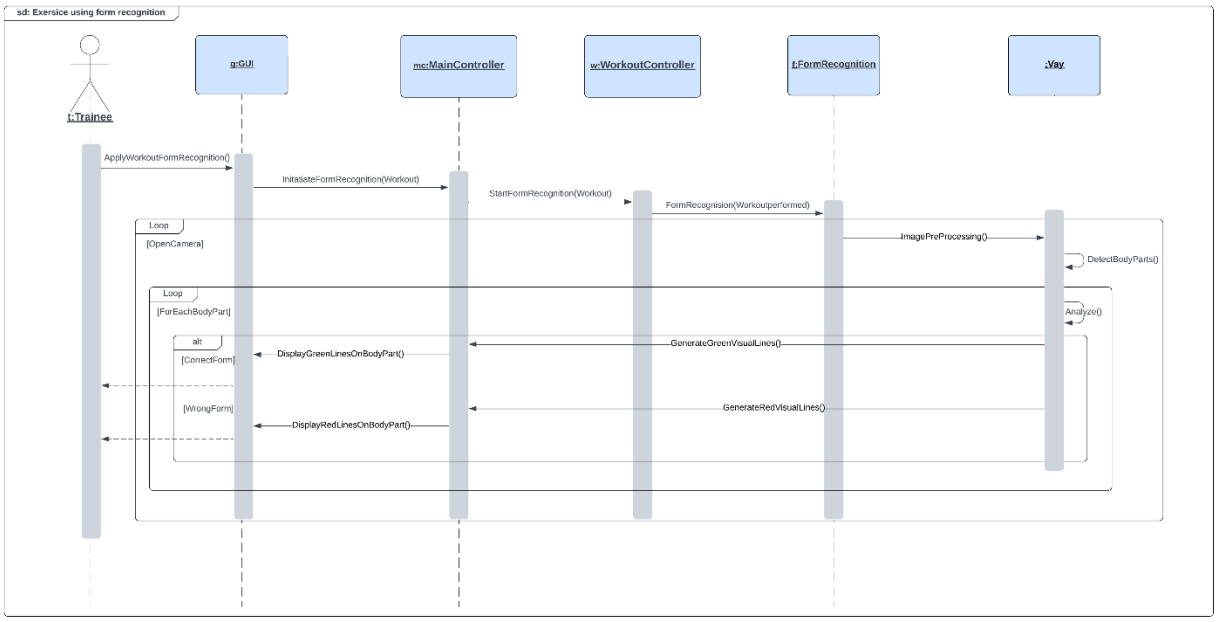


*Figure9.5.3.3: this figure shows the sequence diagram **viewing form recognition** report for Uqla for clearer vision* **Click here**.

9.5.4. State Machine Diagram:

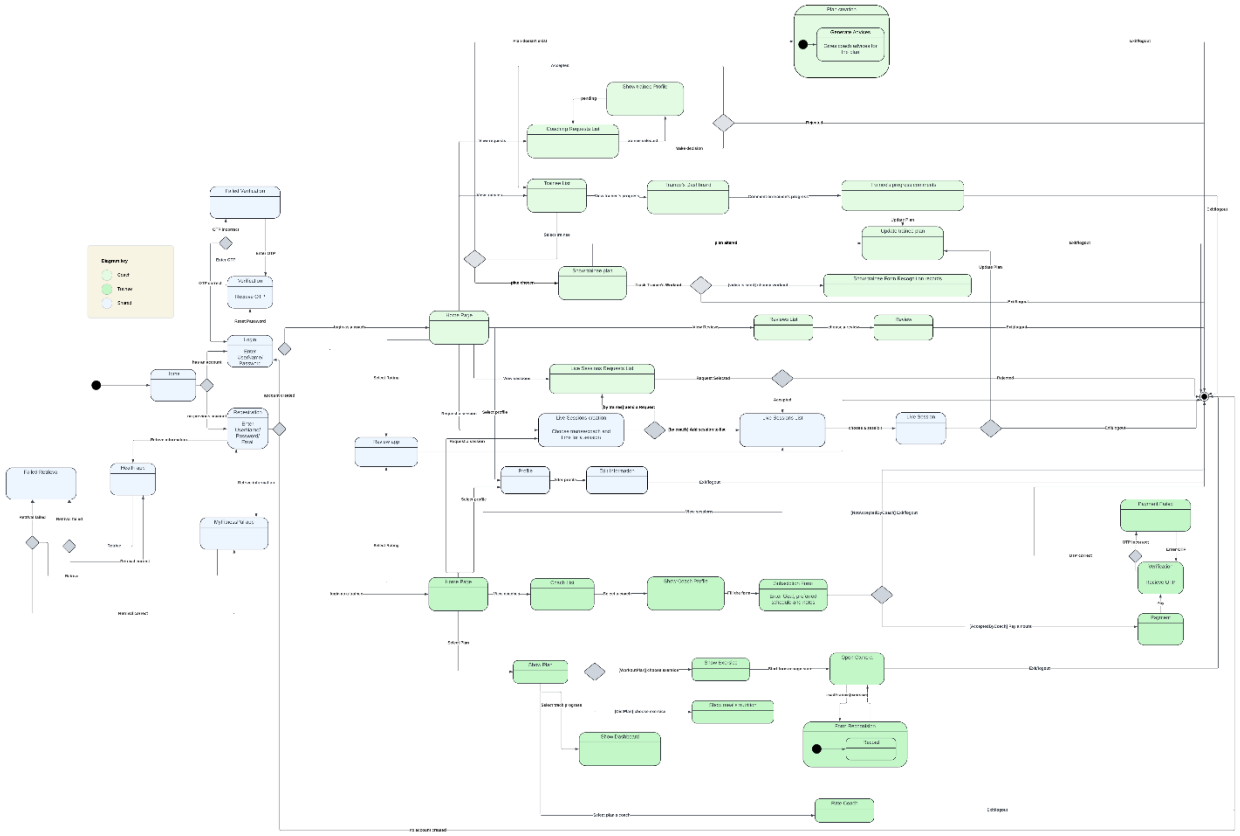


*Figure9.5.4.: this figure shows the design level **state machine** diagram for Uqla, for clearer vision* **Click here.**

9.6. User Interface:

9.6.1. User Interface:

Figure9.6.1.1: this figure shows the Welcome page for Uqla

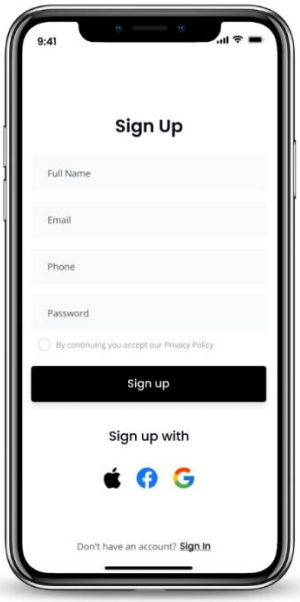


Figure9.6.1.2: this figure shows the and signup page for Uqla

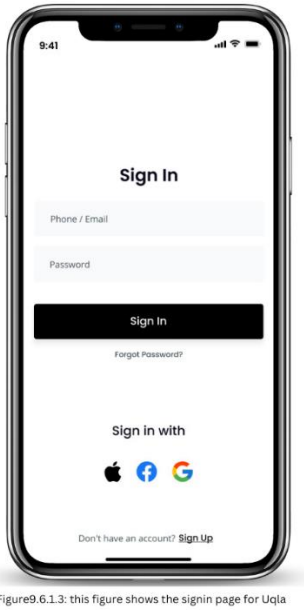


Figure9.6.1.3: this figure shows the signin page for Uqla

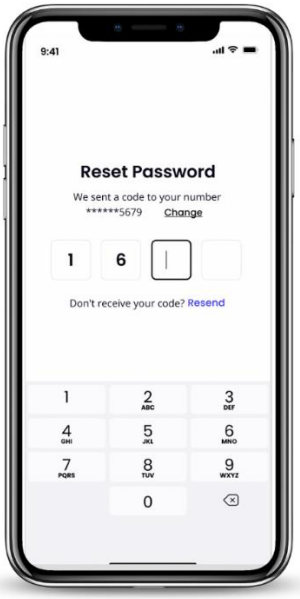


Figure9.6.1.4: this figure shows the reset password verification OTP page for Uqla

9.6.2. Trainee Interface:

Figure9.6.2.1: this figure shows Syncying Health data page for Uqla

Figure9.6.2.2: this figure shows Syncying Fitness data page for Uqla

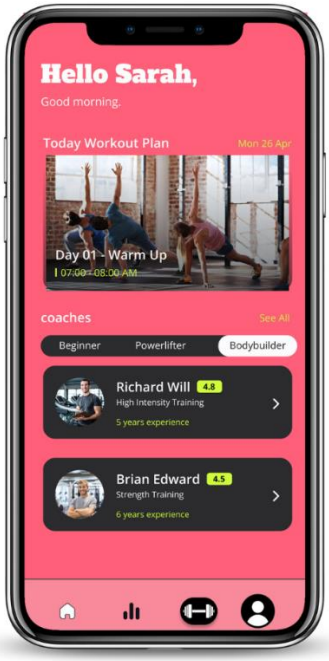


Figure9.6.2.3: this figure shows the Home page for Uqla

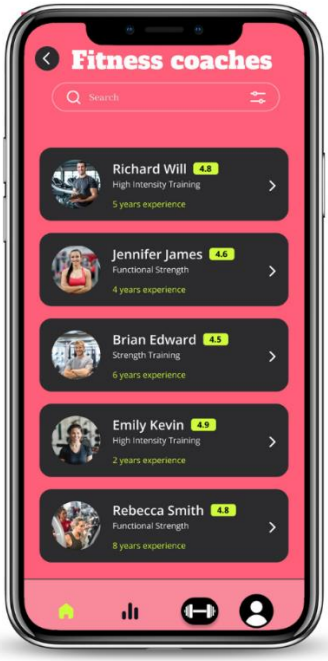


Figure9.6.2.4: this figure shows the coaches list page for Uqla

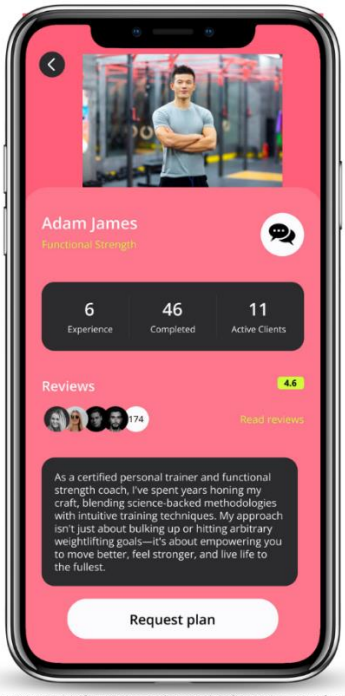


Figure9.6.2.5: this figure shows the coach information page for Uqla

Figure9.6.2.6: this figure shows the request plan from coach page for Uqla

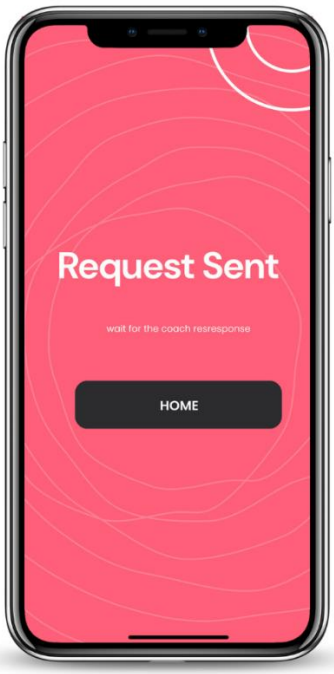


Figure9.6.2.7: this figure shows the request confirmation page for Uqla

Figure9.6.2.8 this figure shows the plans page for Uqla

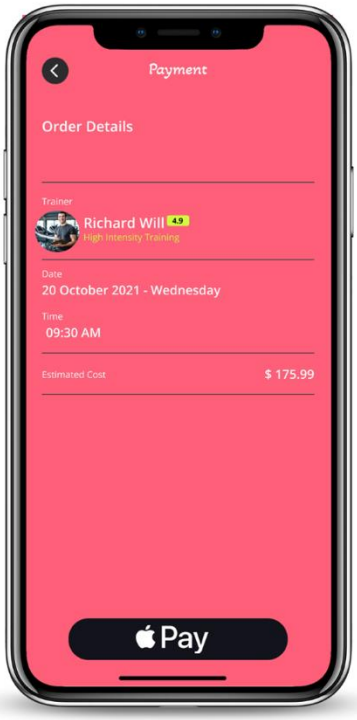


Figure9.6.2.9 this figure shows the payment page for Uqla

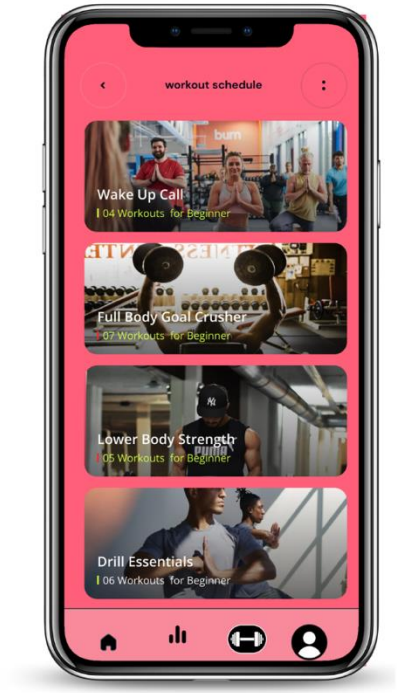


Figure9.6.2.15 this figure shows the workout plan page for Uqla

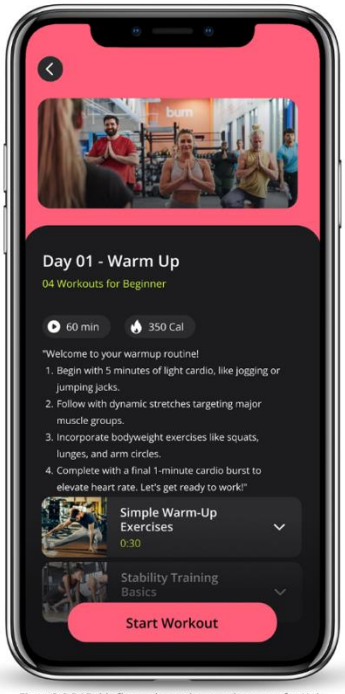


Figure9.6.2.15 this figure shows the exercises page for Uqla

Figure9.6.2.17 this figure shows the video demonstration page for Uqla

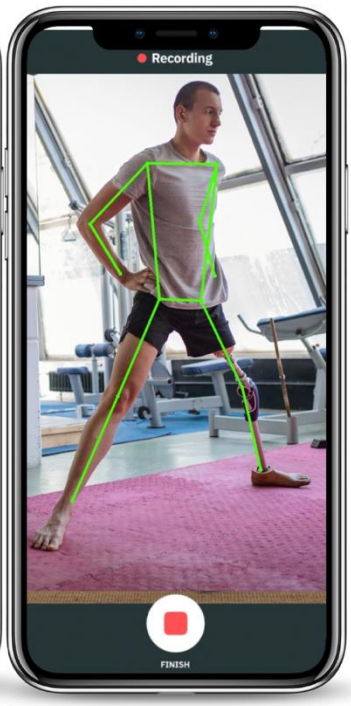


Figure9.6.2.12 this figure shows the form recognision and recording page for Uqla

Figure9.6.2.10 this figure shows the booking sessions page for Uqla

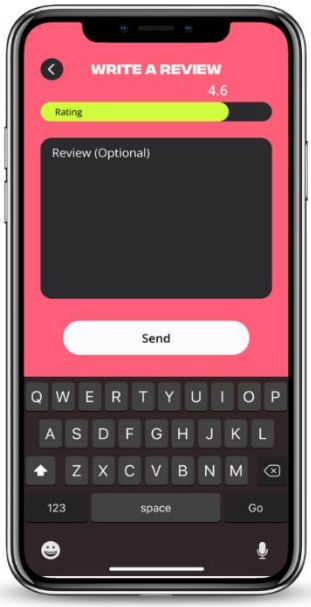


Figure9.6.2.11 this figure shows the review page for Uqla

Figure9.6.2.13 this figure shows the video demonstration page for Uqla

Figure9.6.2.14 this figure shows the workout schedule page for Uqla

9.6.3. Coach Interface:

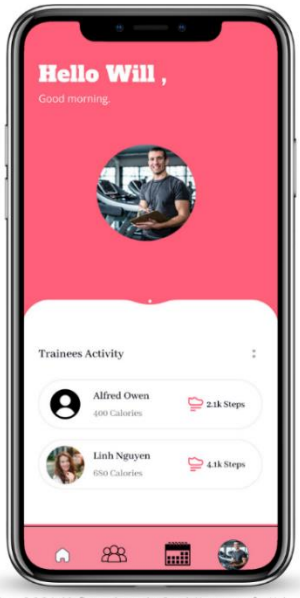


Figure9.6.3.1 this figure shows the Coach Home page for Uqla

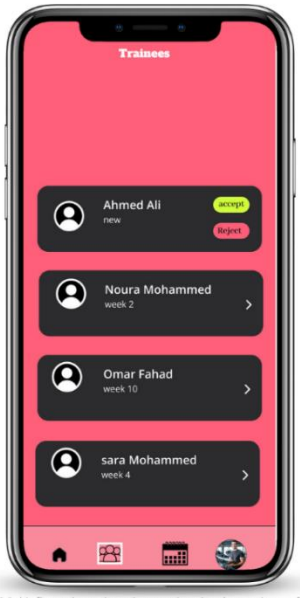


Figure9.6.3.2 this figure shows the trainees assigned to the coach page for Uqla

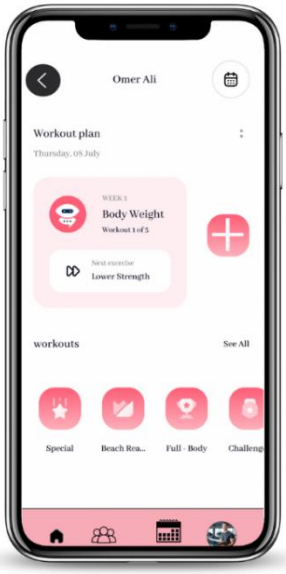


Figure9.6.3.3 this figure shows the trainee workout plan page for Uqla

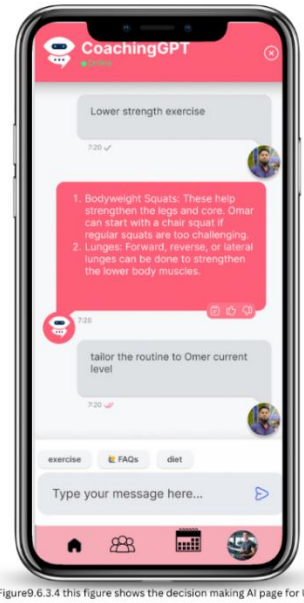


Figure9.6.3.4 this figure shows the decision making AI page for Uqla

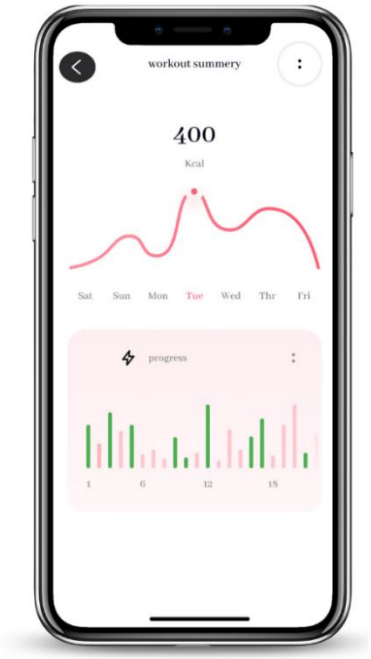


Figure9.6.3.5 this figure shows workout summary page for Uqla

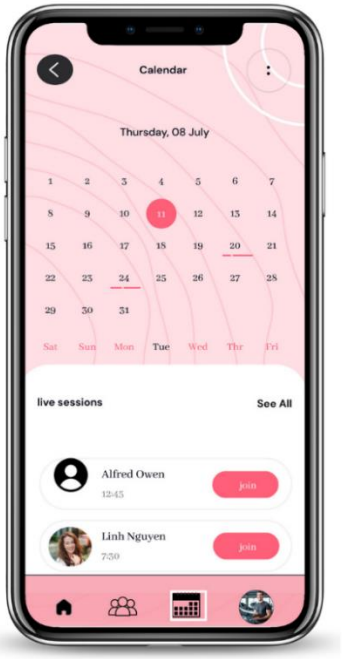


Figure9.6.3.6 this figure shows the sessions calendar page for Uqla

10. Non-Functional Properties:

We choose to implement the MVC, Blackboard and batch sequential architectural styles to uphold essential quality factors such as:

10.1. Reliability: Considering the high demand for users to access health information and maintain proper exercise form, the system shall be available 99.0% of the time, as any downtime could lead to perform the exercise incorrectly which could lead to injuries.

10.2. Usability: Uqla will provide an easy interface for users to interact with the application since individuals vary in levels of technological familiarity, so the user shall learn how to use the system's core features in less than 10 minutes.

10.3. Correctness: Given that one of the core functionalities is the form recognition, which is used by trainees to do their exercise, the system shall display red light indicating incorrect form. Also, Incorrect prediction by form recognition system during exercise can lead to potential injury, so the ML supervised learning model accuracy rate shall be at least 80%.

10.4. Interoperability: As coaches require extensive information to create personalized plans for trainees, the system shall interoperate with Health app to retrieve health data, and with MyFitnessPal to retrieve diet data.

10.5. Testability: To ensure that Uqla is developed accurately and securely, the system shall be able to pass the testing in less than 6 weeks.

10.6. Integrity: To guarantee the security of the users, the system shall send an OTP for user password reset.

10.7. Maintainability: To guarantee the long-term maintenance of Uqla, the software documentation shall adhere to the IEEE documentation guidelines, as team members frequently change, allowing them to get to speed as soon as possible. Furthermore, it could make the process of diagnosing and resolving errors faster.[9]

10.8. Efficiency: To ensure optimal efficiency, the Uqla application shall not use more than 10% of battery power in two hours' time during active use.

In Summary, we will develop a system that will increase the user experience satisfaction, it will be available in both Arabic and English, it will also support portability through running on the iOS platform.

11. Quality Assurance:

11.1. Reviews:

In our system "Uqla" development team will conduct **Walkthrough** as well as **Software Audit Review**. Walkthrough is a type of **Peer Review** in which a work product is examined by the author of work product along with one or more colleagues to evaluate technical parts of software product. As for Software Audit Review, which is an external review led by managerial level individuals who are not part of the development team. It will ensure that every phase is checked properly from an outside view perception.
The reviews will be conducted at various stages of the software development lifecycle, we decided to conduct them in every sprint.

11.2. Verification:

Verification addresses the question, "Are we building the product right?". It will ensure that every step in building the software delivers the correct product. We intend to achieve that by conducting **Inspections**.
Inspections are the most structured and formal type of verification method; it includes two techniques: dynamic and statistic. Statistic techniques aim to find defects and problems mostly in documents, such as requirements, specifications, test plans, test cases, and coding (without actual execution). As for dynamic techniques, tested code is executed to find any defect in the code and helps in rectifying it.[17]
And by using Quality assurance procedures and work instructions, the inspectors will make sure that the standards and regulation are being followed properly.

11.3. Validation:

Validation testing is the process that answers the question whether the product that has been built is correct and right in order. It ensures that the product has been developed as intended based on requirements and specifications.
For us to cover the functional and non-functional requirements, we will use the Black Box two types testing which are:

- **Functional Testing:**
  This type of testing is concerned whether the system is meeting its functional requirements and specifications. Conducting this test needs to go throw 3 steps:[21]
  1. Unit testing: is the most 'micro' scale of testing where each unit of software or is tested.
  2. Integration testing: conducted when two units of software program are combined and tested together, for this test we will use Bottom-up approach which starts from low level units up to the high-level units.
  3. System testing: comes after the system has been integrated in which the system as a whole is tested by examining functions of software based on overall requirements and specifications.

- **Non-functional Testing:**
  This type of testing is concerned whether the system is meeting its non-functional requirements. We will use 2 Types:[21]
  - Stress testing: testing beyond the limits of normal operation, it emphasizes robustness, availability, stability, reliability and performance.
  - Penetration testing: simulates an attack by a malicious hacker, this ensures the security of the system.

By using Black box, we ignore internal working mechanisms and implementation details of the software and focus on what's the output.

11.4. Acceptance Criteria:

- Procedures:
  Rule-oriented acceptance criteria format is used to make sure all the conditions of the software product are accepted by the user.[19]

- Testing:
  Bata testing will be performed to help developers evaluate the software's performance and acceptance in real-world scenarios, it also helps in getting user's feedback.[20]

- Training:
  End user training will be focused on the form recognition feature as an **online tutorial** before using the feature to make sure all users use it properly. We will Also provide an **instructions video** to illustrate the application other functionality.

- Documentation:
  Uqla will provide **user documentation** that includes instructional guides and training manuals for end-users. As well as **Technical documentation** that includes system documentation and process documentation that helps developers and system administrators make changes to a product.

12. Future Considerations:

Uqla will use external systems and cloud computing in development to ensure future-proof software, as well as good documentation that Uqla will ensure by **treating the project as if it were open-source technique**, this will ensure that you always have solid documentation.[18]
Machine learning is a rapidly changing field, especially computer vision. Therefore, our system will be able to accommodate these changes and replace or add any new video recognition technologies that may help our form recognition feature.
An additional feature could be enhancing trainee and coach live workout classes, this could be done through Augmented Reality (AR), this would be useful for in-house trainees who wants to attempt live classes with coaches.
Using Blackboard for Data Analysis subsystem will be beneficial for scalability and reusability, it is useful for accommodating changes in ML and ensuring seamless integration of new methodologies. And by using MVC architecture style software Flexibility is ensured as components are independent, and changes to one component do not affect the others, allowing for easier updates and modifications. But as the system scales, increased intercomponent communication may lead to bottlenecks and decreased performance.
Uqla's key feature is form recognition using computer vision technology, it is a challenging feature as computer vision is rapidly improving and keeping up with this change might be costly. As well as plan management subsystem, it is considered as key functionality in our system but it might be a risky one, as most of its functionality is dependent on the data collected from the external systems.
proof of concept (POC) will be used to Check if the intended functionality can be implemented with the selected technology. As POC helps ensure that the external systems or APIs are feasible with the system, work as expected and can seamlessly integrated with our application.

References:

[1] Lutkevich, B. (2023, June 27). *Artificial Intelligence Glossary: 60+ terms to know*. WhatIs. https://www.techtarget.com/whatis/feature/Artificial-intelligence-glossary-60-terms-to-know

[2] Rangari, Tushar & Kumar, Sudhanshu & Roy, Partha & Dogra, Debi & Kim, Byung-Gyu. (2022). Video based exercise recognition and correct pose detection. Multimedia Tools and Applications. 81. 10.1007/s11042-022-12299-z.

[3] What is an app? | definition from TechTarget. (n.d.). https://www.techtarget.com/searchmobilecomputing/definition/app

[4] Powerlifter noun - definition, pictures, pronunciation and usage ... (n.d.). https://www.oxfordlearnersdictionaries.com/definition/english/powerlifter

[5] Bodybuilder | English meaning - cambridge dictionary. (n.d.-b). https://dictionary.cambridge.org/dictionary/english/bodybuilder

[6] Beginner | English meaning - cambridge dictionary. (n.d.-a). https://dictionary.cambridge.org/dictionary/english/beginner

[7] Richards, K., & Wigmore, I. (2023b, December 28). *What is a one-time password (OTP)? definition from searchsecurity*. Security. https://www.techtarget.com/searchsecurity/definition/one-time-password-OTP

[8] Codecademy. (n.d.). *MVC: Model, view, Controller*. https://www.codecademy.com/article/mvc

[9] Quality Requirement Documentation Guidelines for Agile Software ... (n.d.-f). https://ieeexplore.ieee.org/document/9810243/

[10] What is an API? - application programming interface explained - AWS. (n.d.-h). https://aws.amazon.com/what-is/api/

[11] Researcher, V. K. A., Kanade, V., & Researcher, A. (2022, August 30). *What is machine learning? understanding types & applications*. Spiceworks. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/

[12] *What is supervised learning?*. IBM. (n.d.). https://www.ibm.com/topics/supervised-learning

[13] - He, H., & Garcia, E. A. (2009). *Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering,* 21(9), 1263-1284.

[14] The #1 human motion analysis for fitness & health: Vay: Switzerland. VAY. (n.d.). https://www.vay.ai/

[15] Architectural Design Decision. CS 410/510 - Software Engineering Class Notes. (n.d.). https://cs.ccsu.edu/~stan/classes/CS410/Notes16/06-ArchitecturalDesign.html

[16] Software Architecture & Design Guide. Quick guide. (n.d.). https://sceweb.sce.uhcl.edu/helm/SWEN5233%20Software%20Architecture/my_files/TableContents/Module-14/quick_guide.html

[17] *Faster Time to Market* | Agile methodology promotes shorter development cycles called sprints, which allow for faster delivery of software.
The Benefits & Advantages of Agile | Wrike Agile Guide
[18] Wilson, R. (2014, May 27). *Open Source And The Cloud*. OSSWATCH. http://oss-watch.ac.uk/resources/opensourceandcloud
[19] (n.d.). *Acceptance Criteria for User Stories in Agile: Purposes, Formats, Examples, and Best Practices*. Altexsoft. https://www.altexsoft.com/blog/acceptance-criteria-purposes-formats-and-best-practices/
[20] (n.d.). *Beta Testing*. GeeksForGeeks. https://www.geeksforgeeks.org/beta-testing-software-testing/
[21] (n.d.). *Types of Software testing*. GeekForGeeks. https://www.geeksforgeeks.org/types-software-testing/
[22] Heads-Up: Amazon S3 Security Changes Are Coming in April of 2023 | AWS News Blog Anon 2022 aws.amazon.com https://aws.amazon.com/blogs/aws/heads-up-amazon-s3-security-changes-are-coming-in-april-of-2023/
[23] Ardalis. (n.d.). *Handle requests with controllers in ASP.NET core MVC*. Microsoft Learn. https://learn.microsoft.com/en-us/aspnet/core/mvc/controllers/actions?view=aspnetcore-8.0