ISS 496- Senior Project

Final Report

Attendance tracker for AWS

In collaboration with: Orange digital center

Elaborated by:

Med Amine Karoui, Med Rayan Belaam

Khadija Yasmine Mzabi, Skander Lahbaiel

Supervised by:

Ms. Turki, Mariem - Orange

Dr. Amdouni, Asma - MedTech

Tunis, 2022-2023

Table of Contents

List of Tables

List of Figures

1- Project Overview

Describes the problem domain and the specific problem to be resolved, reviews existing solutions and their inadequacies, and how the design project, by the use of modern design theory and methodology, adequately resolves the specific problem.

1.1- Introduction

The manual process of taking attendance leaves room for human error and is susceptible to inaccuracies, making it unreliable and inefficient. Automating the attendance process using technology, such as facial recognition, helps mitigate these issues and saves valuable time for the administration and learners. With a system that is able to accurately and quickly track attendance, the administration can focus on more important tasks and learners can spend more time on their education.

This report provides an overview of our project, including a description of the project idea and its implementation process. It outlines the methodologies, tools, and frameworks used, and culminates in a final version of the project, complete with all the features discussed and documented in the subsequent sections.

1.2- Project Description

The Orange Digital Center provides a range of resources and services to support digital innovation, including training programs. The Orange Company has established numerous Orange Digital Centers around the world that function as dynamic hubs for networking and collaboration to share knowledge, ideas, and best practices. [1]

Our project aims to improve the efficiency of attendance tracking at the Tunisian Orange Digital Center. It consists of a mobile app with a facial recognition system for Orange AWS learners to track attendance. The attendance tracker project not only saves time but also provides a more secure and accurate solution for tracking attendance. By using facial recognition, the app can detect the learner's face and match it with their profile, ensuring that the correct individual is being marked present. The system also eliminates the need for manual exchanges of badges, reducing the risk of lost or stolen badges, and making the process more secure. The use of image metadata to extract time and location adds an extra layer of accuracy to the attendance tracking process, ensuring that the learners are indeed at the Orange ODC when they are marked present. The daily attendance report and cohort management features provide the administration with valuable insights and control over the attendance process, streamlining the overall workflow and making it easier to track attendance.

1.3- Problem Definition

The ODC management has faced various challenges in taking attendance for its programs. One of the most significant issues is the risk of false data, as there is a possibility of learners manipulating the

badging system. Additionally, instructors may overlook taking attendance, which can lead to incomplete records. Another concern is that full-day absence may occur if an attendee arrives late or leaves during attendance time. Furthermore, the attendance data is not readily available at the ODC level, and the manual attendance reports are often incomplete, further complicating the process.

1.4- Vulnerabilities and Existing Solutions

One major potential risk is the precision of this technology. The pictures based on which learners will be recognized and marked present throughout the year may differ greatly from those the model was trained. This will affect the certainty of our system and we might have errors when taking attendance.

An existing solution is the automation of attendance tracking via badges. This solution is currently used by the ODC, but it has many related problems, regarding the storage of data which is centralized and cheating.

Other digital solutions exist (INVOLVIO), where learners can scan a QR code, proving that they are at that location at that time.[2]

The first screen is the QR code screen that the learner should scan to mark their presence.

Figure 1: QR code scanner

The second screen present the attendance report.

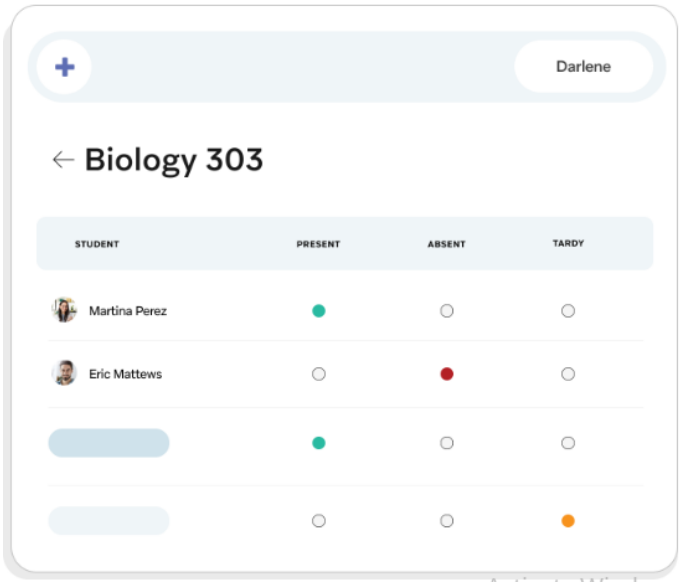

Figure 2: Attendance Report

Then, the last screen consists of the admin dashboard.

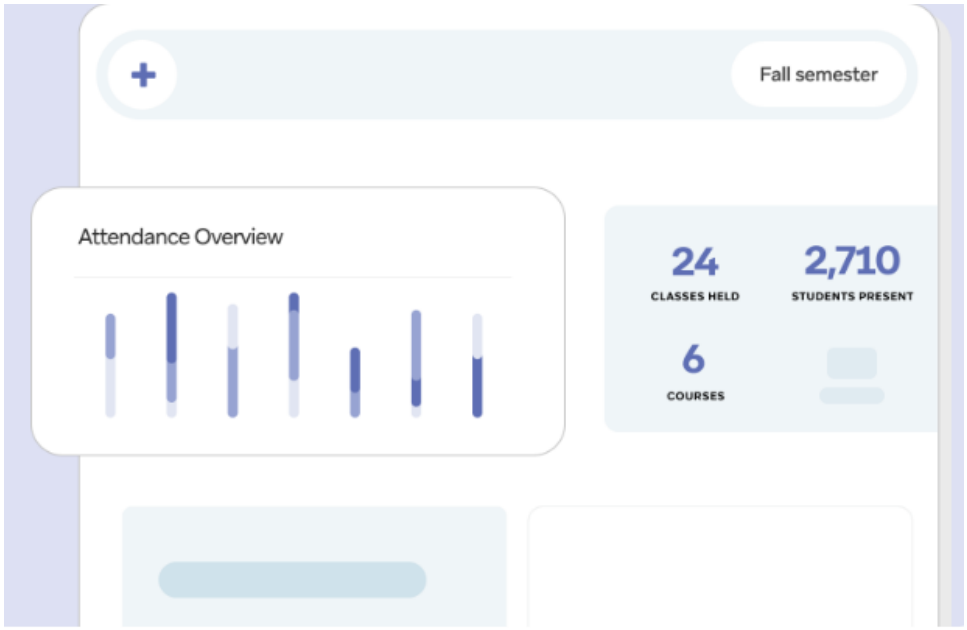

Figure 3: Admin Dashboard

1.5- Proposed Solution

The proposed system will automate the process of taking attendance by recognizing learners based on image processing, sending metadata containing date, time and location to the server and marking the learner either absent or present.

It will facilitate the job of the administration by providing the admin with a detailed report after each session and providing him with a dashboard enabling him to add and remove cohorts set start and end times.

The system will solve the problem of tardiness and early leaving by taking the attendance each time a learner arrives to class.

1.6- Literature Review

Before starting to work on our project, we searched for similar projects in order to get an idea about the architecture, the design and the implementation of our system. Two such systems are the Smart Attendance with Real-time Facial Recognition and the Automatic Attendance System, which were developed in response to the limitations of manual attendance recording methods. Both utilize facial biometrics to quickly and accurately identify participating students in a classroom setting. [3] [4]

During the implementation phase, we found some helpful source code on Github that enabled us to work more effectively on the frontend and backend.

1.7- Conclusion

In this part, we have introduced the main idea and the main goal of our project. Automating the attendance process using facial recognition saves valuable time for both the administration and the learners. In the next part we will present the steps needed to achieve the implementation of our app.

2- Product Specifications

2.1-Introduction

In the "Product Specifications" section, the focus will be on defining the needs of the end user and converting them into technical requirements for the solution. This section will clearly outline the measurable specifications, using explicit metrics and values that will be used to test the solution's effectiveness. Additionally, the section will address the standards and constraints that are relevant to the problem being addressed and the proposed solution. The reader can expect to understand how each specification and its value was determined, providing a thorough explanation of the thought process behind the design of the solution.

2.2- Project Management

The management of a software project is a crucial phase as it plays a vital role in maintaining a balance between the three fundamental aspects of the project: time, cost, and quality.

Our team utilized an agile process for project management to achieve this balance effectively.

2.2-1. Methodology

For our project, we have opted to use the Scrum development process, which falls under the Agile framework. The Scrum methodology is characterized by incremental and iterative development practices and is well-suited for our project. In accordance with Scrum, our project team will be composed of distinct roles, including the Product Owner who is responsible for defining and prioritizing the features and functionality of the product, the Scrum Master who is responsible for facilitating the Scrum process and helping the team to continuously improve, and finally the Development Team that have to design, build, and test the product increment. [5]

2.2-2. Timeline

We have divided the project into features, where each feature has a set of user stories assigned to it. We did this to have a logical representation of the project and to define its priorities.

We split the development period into X 2-weeks sprints. In this subsection, we present how we conducted the Scrum ceremonies.

Sprints	From	To
Sprint 1	13/02/2023	28/02/2023
Sprint 2	28/02/2023	14/03/2023
Sprint 3	14/03/2023	30/03/2023
Sprint 4	30/03/2023	16/04/2023

Table 1: Sprint's Timeline

2.3-Product Backlog

You can find here our product backlog that consists of a prioritized list of features and requirements that need to be developed and implemented in the product.

ID	Feature	User Story			Priority	Sprint
		As a ...	I want to be able to ...	So that I can ...		
1		User	Send my picture	Take attendance	MUST	1

	Facial recognition		Send my picture	Add my face to database	MUST	
			Retake a picture	Send clear pictures	MUST	
2	Manage cohorts	Admin	Add new cohort	Add learners to it	MUST	2
			Add new user	Allow users to use the app	MUST	
			Edit user	Keep the system consistent	MUST	
			Delete user	Remove users who finished	MUST	
			Set start/end times of cohorts	Inform users	MUST	
			Assign instructors		MUST	
			Assign classrooms		SHOULD	
3	Password authentication	User	Login and logout		MUST	3
4	Attendance Reports	Admin	Have access to users' attendance reports	Analyze the data	SHOULD	
			Visualized attendance analytics	Have better presented data	SHOULD	
		User	Have access to my attendance report	Keep track of my attendance	COULD	
5	Biometric authentication	User	Upload pictures or videos of my face	Create an account	COULD	4
6	User Feedback and Support	User	To provide feedback	Receive support	COULD	

Table 2: Product Backlog

2.4-Sprint Backlog

The tables below detail our development team's sprint backlogs and offer a breakdown of the tasks handled by each team member.

	Rayan	Amine	Khadija	Skander
Implement facenet model using TensorFlow and Keras.				X

Develop a database to store face encryptions.			X	
Create a mechanism to compare new encryptions with the threshold and determine whether the face is recognized or not.				X
Develop a mobile application to capture pictures and send them to the backend.		X		
Implement the code that receives the picture from the mobile application, encodes the face, and compares it to the encryptions in the database to determine whether the user is known or not.	X			
Write code to update the user's attendance record if their face is recognized, or prompt them to retake the picture if their face is not recognized.			X	

Table 3: Sprint 1 Backlog

	Rayan	Amine	Khadija	Skander
Design and create a database schema for storing cohort information	X			
Develop a REST API endpoint for adding new cohorts to the database.			X	
Develop a REST API endpoint for adding learners to a specific cohort3			X	
Develop a REST API endpoint for adding new users to the system.		X		
Develop a REST API endpoint for editing user information	X			
Develop a REST API endpoint for deleting users from the system.				X
Develop a REST API endpoint for setting the start and end times of a cohort.		X		
Develop a REST API endpoint for assigning instructors to a cohort.				X
Develop a REST API endpoint for assigning classrooms to a cohort.			X	
Implement a UI for managing cohorts	X			
Develop unit tests for each REST API endpoint to ensure that it functions as intended.		X		

Table 4: Sprint 2 Backlog

	Rayan	Amine	Khadija	Skander
Implement a login form that prompts the user for their username and password.	X			
Verify the user's credentials by checking them against the stored user database.	X			
Implement a password hashing algorithm to securely store user passwords.			X	
Create a "forgot password" feature that allows users to reset their password if they forget it.				X
Implement a session management system to keep the user logged in until they choose to log out or their session expires.		X		
Create a logout function that clears the user's session and redirects them to the login page.			X	
Implement error handling for incorrect login credentials or other authentication-related errors.		X		X
Ensure that all password fields are properly protected and encrypted to prevent unauthorized access to sensitive user data.	X			

Table 5: Sprint 3 Backlog

	Rayan	Amine	Khadija	Skander
Develop a backend API for generating attendance reports.			X	
Develop a database schema to store attendance data and generate reports.	X			
Integrate data visualization tools to present attendance data in a visually appealing manner.		X		
Develop a feature to allow users to keep track of their attendance history.		X		
Create a user feedback form to capture user feedback.	X			
Implement a chat or messaging feature to allow users to communicate directly with support staff.			X	

Enable users to rate and provide feedback on support interactions.				X

Table 6: Sprint 4 Backlog

2.5-Requirements Specification

2.5-1. Functional Requirements

In this section, we listed our functional requirements which describe the specific behaviors and actions that the product must perform to meet the needs of the user and achieve its intended purpose.

Requirements	Description	Significance
Facial recognition	The system will provide facial recognition features for attendees. Each learner will be able to use the camera to take a picture, after recognition, the data will be sent to the server.	ODC Attendance tracker to optimize time management by eliminating the manual task of listing learners. ODC Attendance tracker to have better access to information and avoid the problem of centralization of data. Tensorflow or Keras to be used for facial recognition.
Cohort management	The dashboard will provide the admin with the ability to start/end time, add and remove cohorts. He will also be capable of adding learners to cohorts and upload images of new learners to the database in order to be recognized by the system.	ODC Attendance tracker to digitalize the administration's job by providing the admin with the power to easily manage and maintain the system, making it more efficient and effective. The front end will be implemented with React, nest.js will be used as API, and MongoDB as the database.

Presence verification	The system will extract metadata from the images, verify that the learner is in class at the right time and then mark him present. The system will provide a report, containing data about present and absent learners.	The system will extract metadata from the images, verify if the learner belongs to that cohort, if he has class at that time and mark him accordingly.

Table 7: Functional Requirements

2.5-1.1 Use case Diagram:

The provided use case diagram illustrates the architecturally significant use cases, while simpler and more straightforward use cases are not explicitly represented in the Software Architecture Document. Even though these simpler use cases are part of the system, their architectural approach and mechanisms will be the same as for the more complex use cases described above.

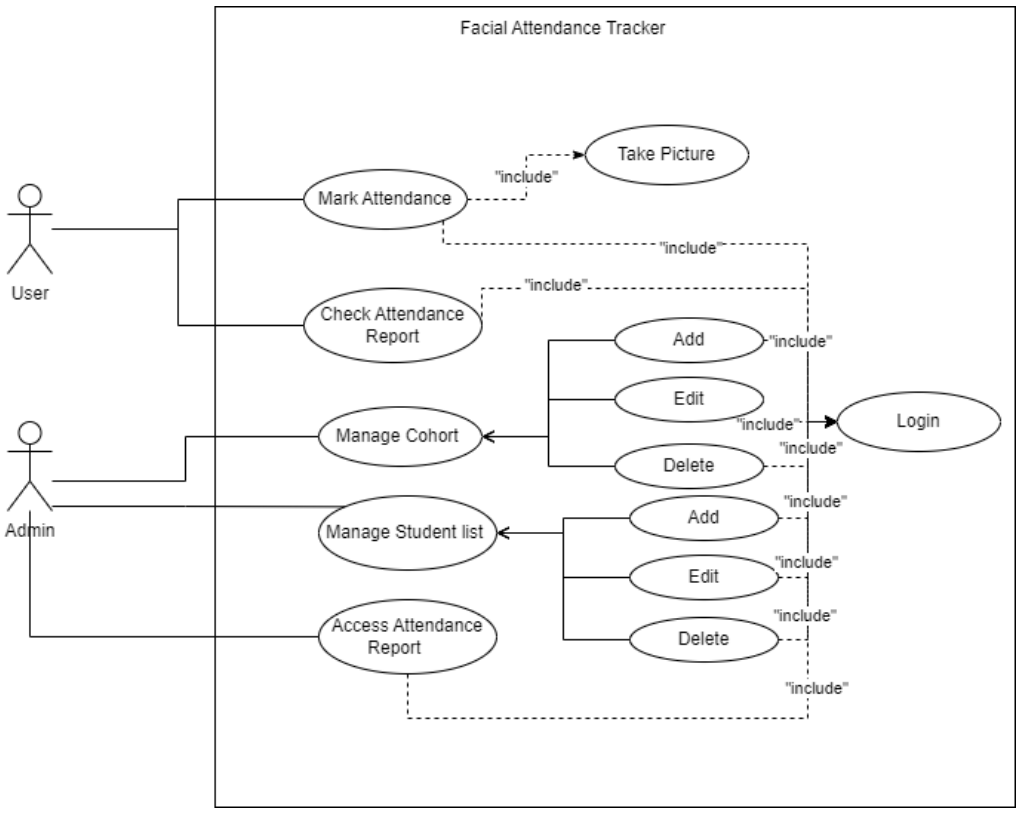

Figure 4: Use Case Diagram

2.5-2. Non-functional Requirements:

In this section, we listed our non-functional requirements, they specify the quality attributes and characteristics that the product must exhibit.

Non- functional requirements	User stories
Usability	• As a user, I should be able to navigate the ODC website easily and efficiently. • As a user, I should be able to access all relevant information about ODC in a clear and organized manner.
Security	• As a user, I should be confident that my personal information is protected when I use the ODC website. • As a user, I should be able to perform transactions on the ODC website securely.
Performance	• As a user, I should be able to load the ODC website quickly and without any delays. • As a user, I should be able to search for information on the ODC website quickly and effectively.
Maintainability	• As a user, I should be able to use the ODC website without encountering any bugs or glitches.

Table 8: Non-functional requirements

Quality Attributes Scenarios:

This section aims to introduce the quality attributes scenarios that we have incorporated into our project. These scenarios serve to outline the essential qualities that the system must possess, described in a practical and accessible way. By providing concrete examples, these scenarios help to clarify the abstract concept of "quality".

Maintainability:

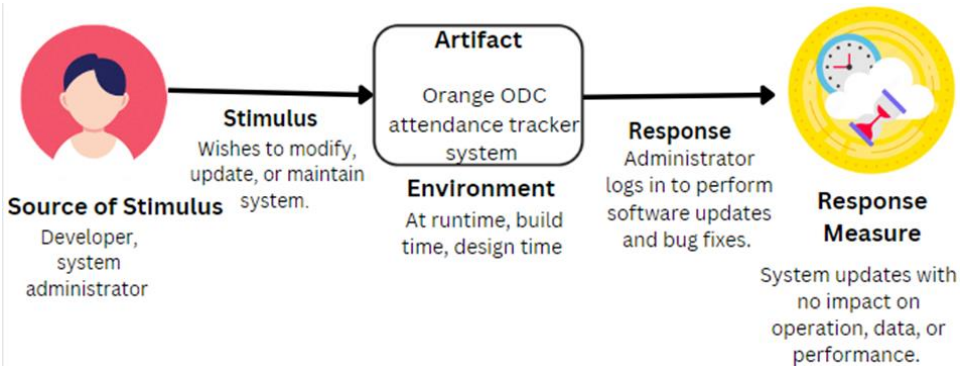

Figure 5: First Quality Attribute Scenario of the maintainability

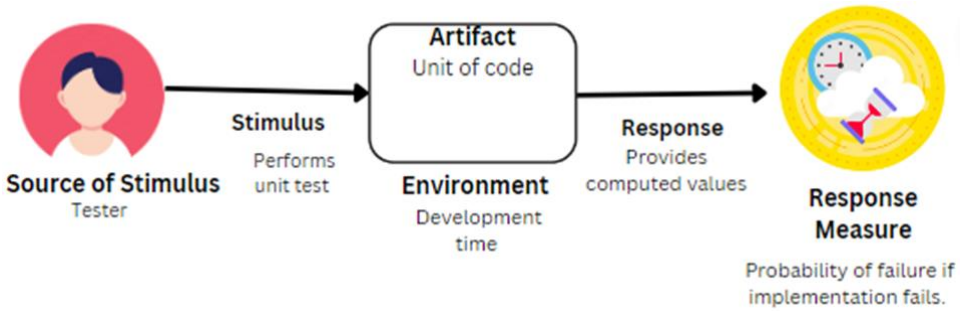

Figure 6: Second Quality Attribute Scenario of the maintainability

Usability:

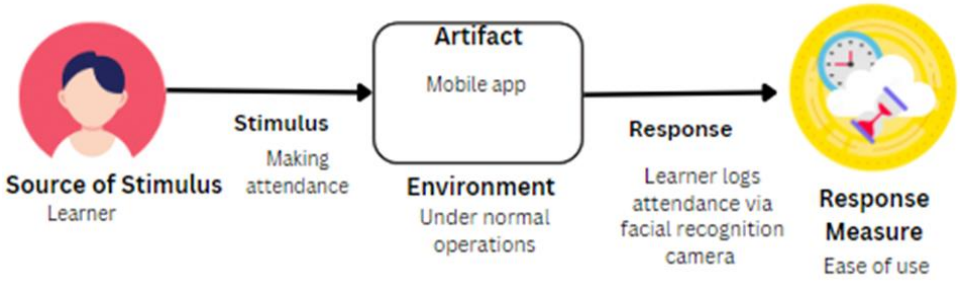

Figure 7: First Quality Attribute Scenario of the usability

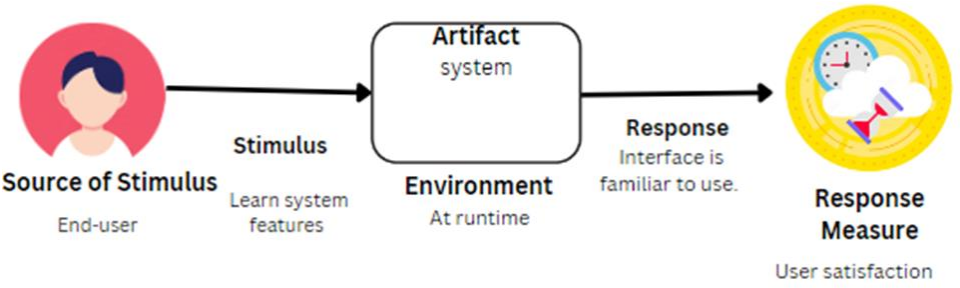

Figure 8: Second Quality Attribute Scenario of the usability

Security:

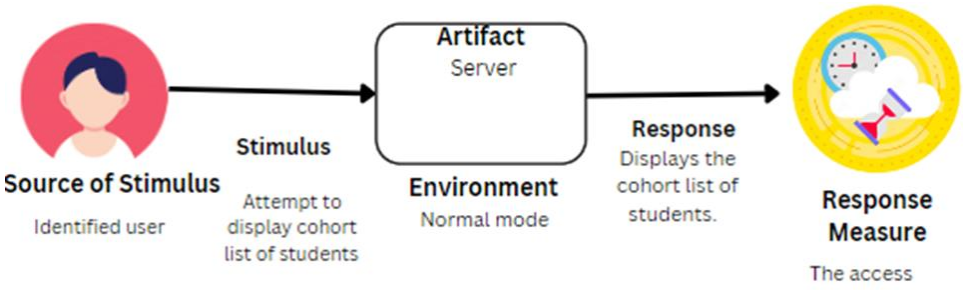

Figure 9: First Quality Attribute Scenario of the security4

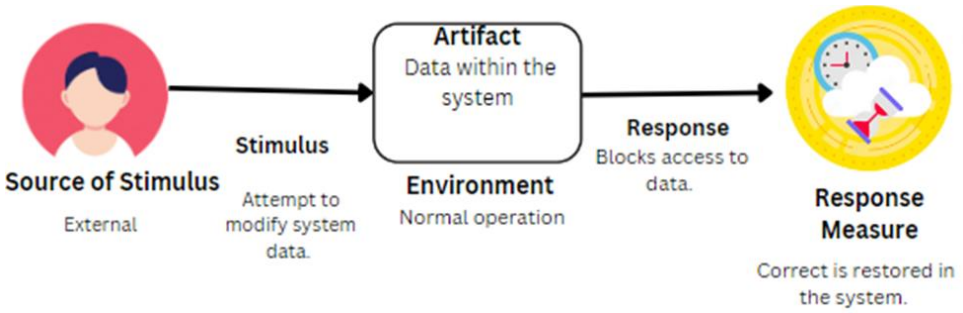

Figure 10: Second Quality Attribute Scenario of the security

Performance:

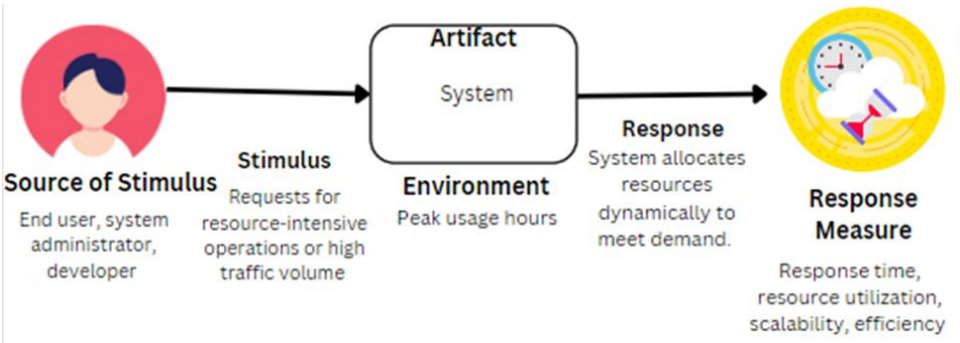

Figure 11: First Quality Attribute Scenario of the performance

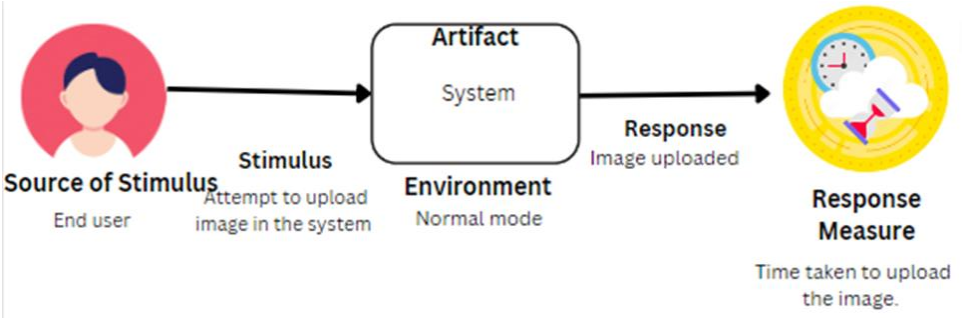

Figure 12: Second Quality Attribute Scenario of the performance

2.6-Project Considerations

2.6-1. Project Constraints

Constraints in a project refer to the restrictions or limitations that should be taken into consideration throughout the project's duration. In our case, we identified the following constraints to reach the project scope.

Design Constraints:

- **Quality:** The quality constraint is concerned with the attributes of the project deliverables. The project's quality will be assessed based on how well the final outcome aligns with the requirements established in the product backlog, which have been validated by the stakeholders and end-users.

Project Constraints:

- **Functional:** To create the end product, we adopted the Agile software development life cycle model which involves a combination of iterative and incremental process models. This approach places emphasis on customer satisfaction through the swift delivery of functional software products.

Course Constraints:

- **Time:** The team has a restricted amount of time to analyze and define requirements, design the architecture as well as implement the system, test, and deploy. “Attendance tracker for AWS” project must be completed within a specified timeframe: 4 months.
- **Limited Prior knowledge/ Experience:** Our team lacks prior knowledge regarding the proposed features and technologies (facial recognition using Tensorflow or Keras)

2.6-2. Project Limitations

One significant limitation of this project is the technological constraints that have been imposed by the product owner. Specifically, we have been instructed to use Python for the backend of the mobile application, Node.js for the API, and either React or Angular for the frontend. The second challenging aspect of the software development is the constrained time frame of approximately three months in which we had to be familiar with the technologies and the architecture of the project.

2.6-3. Project Standards

"A standard is a technical publication created to ensure the reliability of the materials, products, methods, and/or services. They establish the technical requirements, specifications, guidelines, characteristics, and/or procedures designed. Standards are recommend set of design tolerances, practices, operations, manufacturing methods, or uses of equipment within a specified environment." [6]

During this project, we considered the following engineering standard:

- **IEEE/ISO/IEC P15288 - ISO/IEC/EEE Draft Standard - Systems and Software engineering -- System Life Cycle Processes:** This International Standard establishes a common framework of process descriptions for describing the life cycle of created systems. It defines a set of processes and associated terminology from an engineering viewpoint. This standard defines a set of processes grouped into four categories: (1) Agreement processes, Enterprise processes, Project processes, Technical processes
- **IEEE 1008-1987 - IEEE Standard for Software Unit Testing:** An integrated approach to systematic and documented unit testing is defined. It uses unit design and unit implementation information, in addition to unit requirements, to determine the completeness of the testing.
- **ISO/IEC 25010:2011 Quality Model:** is an international standard for software quality that defines a quality model for evaluating software products and systems. The model is based on eight main characteristics, including functional suitability, performance efficiency, usability, reliability, security, maintainability, and portability.
- **ISO/IEC/IEEE 42010:2011 Conceptual Framework:** The standard pertains to the development, evaluation, and maintenance of system architectures by utilizing architecture descriptions.

2.6-4. Business, Social and Ethical Considerations

Business Considerations:

- The cost to the company hiring the students should be minimal, limited to the time spent on the project.
- The project should provide value to the company and contribute to their goals.
- The project should be completed within the specified timeline and budget.
- The project should adhere to all relevant laws, regulations and industry standards.

Ethical Considerations:

- The project should adhere to ethical standards and avoid conflicts of interest.
- The project should be transparent and accountable to all stakeholders.
- The project should respect the privacy and data security of all stakeholders.

2.7-Conclusion

In conclusion, the Product Specification section of this report outlines the comprehensive and well-planned approach to delivering a successful solution to this problem. It details the methodology, timeline, requirements, and project considerations that will guide the development process. The Product Backlog and Sprint Backlog ensure that all user needs, and technical requirements are captured, prioritized, and delivered iteratively. With the right mix of teamwork, collaboration, and attention to detail, the project is set to deliver a high-quality, innovative solution that meets the needs of the end-user.

3- Product Design

3.1- Introduction

Product design is a critical step in engineering where technical solutions are developed to solve problems. This involves identifying systems and subsystems, assigning specifications, and testing them appropriately. Alternative solutions are considered before arriving at the optimal design for the given problem.

3.2- High-Level Design

3.2-1. Architecture Overview

Our application architecture will be organized into separate facial recognition and cohort management attendance servers, along with a web app for the admin dashboard and a mobile app for users/learners:

We opted for an MVC (Model-View-Controller) architecture style, which separates the application into three interconnected components to ensure code reusability and maintainability. [7] For the model part, we utilized MySQL, ReactJS and React Native for the view part, and Node.js for the controller part, to ensure efficient and effective development of the system.

- **Mobile App** :

-Provides a user-friendly interface for users to capture and submit their images for attendance

-Sends captured images to the facial recognition server for processing

-Displays attendance report to the users

- **Web App:**

- Provides an admin dashboard for the management of cohorts, learners and attendance data

- Communicates with the cohort management attendance server via APIs

- Allows admin to view attendance reports, edit cohort information, add and remove learners from cohorts, and upload learner images for facial recognition

- **Facial Recognition Server:**

- Receives captured images from the mobile app

- Uses facial recognition technology to compare the captured image with stored images in a database

- Returns the identification result to the attendance server

- **Cohort Management Attendance Server:**

-Manages the attendance records and cohort information

-Communicates with the facial recognition server to verify attendance status of each learner

-Provides APIs for the web app and mobile app to retrieve attendance data and other relevant information

- **Database:**

-Records user's names, passwords and faces encoding.

-Cohorts data

-Users attendance reports

3.2-2. Component Diagrams

Below is the component diagram that illustrates the primary components and elements constituting the attendance tracker system. Each of these components represents a distinct entity within the system, and its interaction with the other entities.

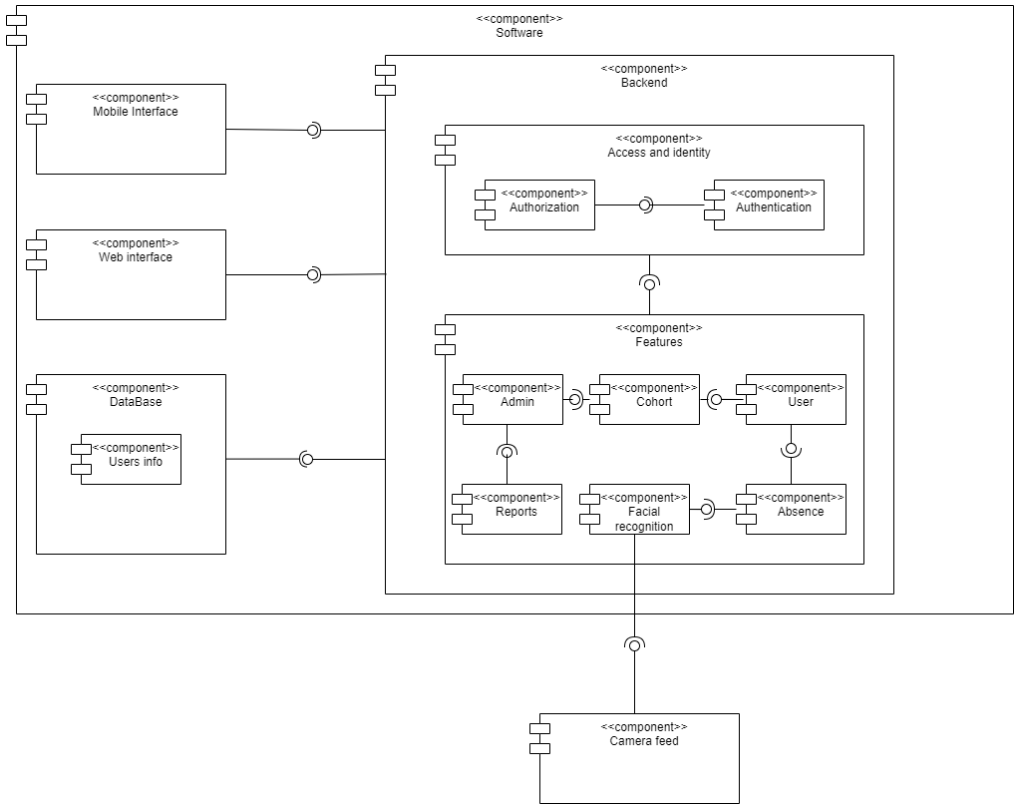

Figure 13: Component Diagram

3.3- Detailed Design

3.3-1. Sequence Diagrams and Use Case Scenarios

In this section, we will present a variety of use case scenarios related to the different use cases seen in section 2.5-1.1

3.3-1.1. Manage Cohort:

Delete	
System	Web application.
Actor	Admin.
Goal	To use the web app to delete cohorts.
Precondition	Admin should be logged in and have access to his dashboard.
Post-condition	Deleting has been completed successfully.

Basic Flow	1. Admin selects specific cohorts. System displays the cohort's list of learners 2. Admin selects the delete function. The system asks for confirmation to delete. 3. Admin confirms. 4. The system saves the changes and displays a success message. Use case ends.
Alternative Flow	**A1.** At 3, admin rejects confirmation to edit. The system returns to the main display. Use case ends

Table 9: Use case Scenario: Delete use case

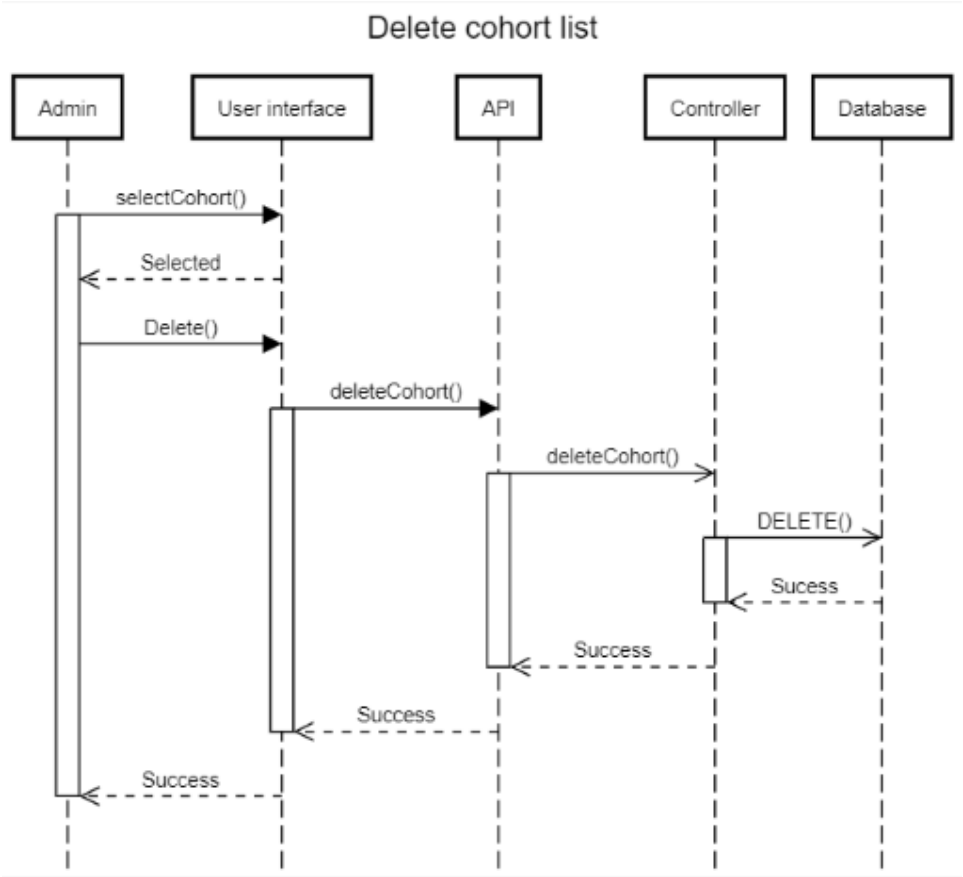

Figure 14: Sequence Diagram: Delete use case

Add	
System	Web application.

Actor	Admin.
Goal	To use the web app to add learners to cohorts.
Precondition	Admin should be logged in and have access to his dashboard.
Post-condition	Learner has been added successfully.
Basic Flow	1. Admin selects specific cohorts. System displays the cohort's list of learners 2. Admin selects the add function. The system asks for confirmation to add learner. 3. Admin confirms. 4. The system saves the changes and displays a success message. Use case ends.
Alternative Flow	**A1.** At 3, admin rejects confirmation to edit. The system returns to the main display. Use case ends

Table 10: Use case scenario: Add use case

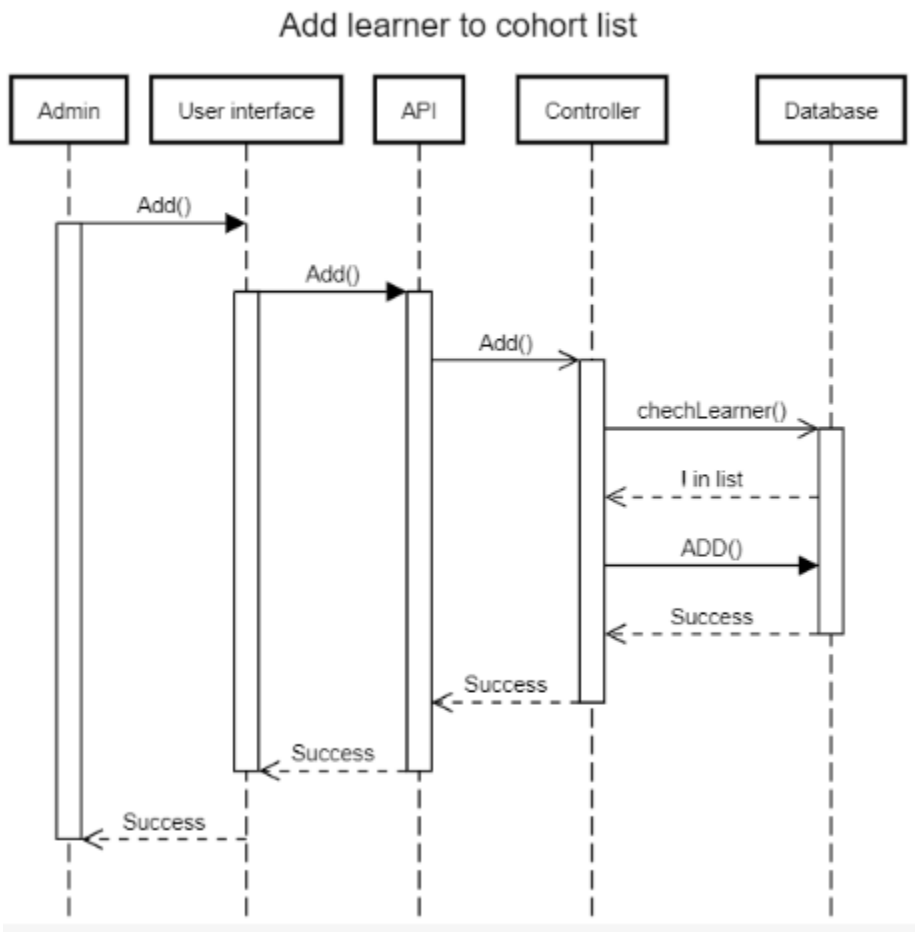

Figure 15: Sequence Diagram: Add use case

Edit	
System	Web application.
Actor	Admin.
Goal	To use the web app to manage cohorts.
Precondition	Admin should be logged in and have access to his dashboard.
Post-condition	Editing has been completed successfully.
Basic Flow	1. Admin selects specific cohorts. System displays the cohort's list of learners 2. Admin selects the edit function. The system asks for confirmation to edit. 3. Admin confirms. 4. The system saves the changes and displays a success message. Use case ends.
Alternative Flow	**A1.** At 3, admin rejects confirmation to edit. The system returns to the main display. Use case ends

Table 11: Edit cohort's use case scenario

3.3-1.2. Make attendance

Make attendance	
System	Mobile Attendance App with Facial Recognition.
Actor	Learner.
Goal	To use the mobile app with facial recognition to make attendance in a classroom.
Precondition	- The learner has the mobile attendance app installed on their smartphone. - The app is connected to the internet. - The learner has been enrolled in the app by the teacher and their face has been registered in the database.
Post-condition	The learner's attendance has been successfully recorded by the app.
Basic Flow	1- The learner opens the attendance app on their smartphone.

	2- The app accesses the smartphone's camera and displays the camera feed. 3- The learner holds their device up to their face, and the app uses facial recognition algorithms to match their face against a database of enrolled learners. 4- The app successfully recognizes the learner's face, their attendance is recorded, and a message appears confirming that their attendance has been taken.
Alternative Flow	**A1. At 4, if** the app is unable to recognize the learner's face, they can try again from step 2 of the Basic Flow.

Table 12: Use case scenario: Make Attendance

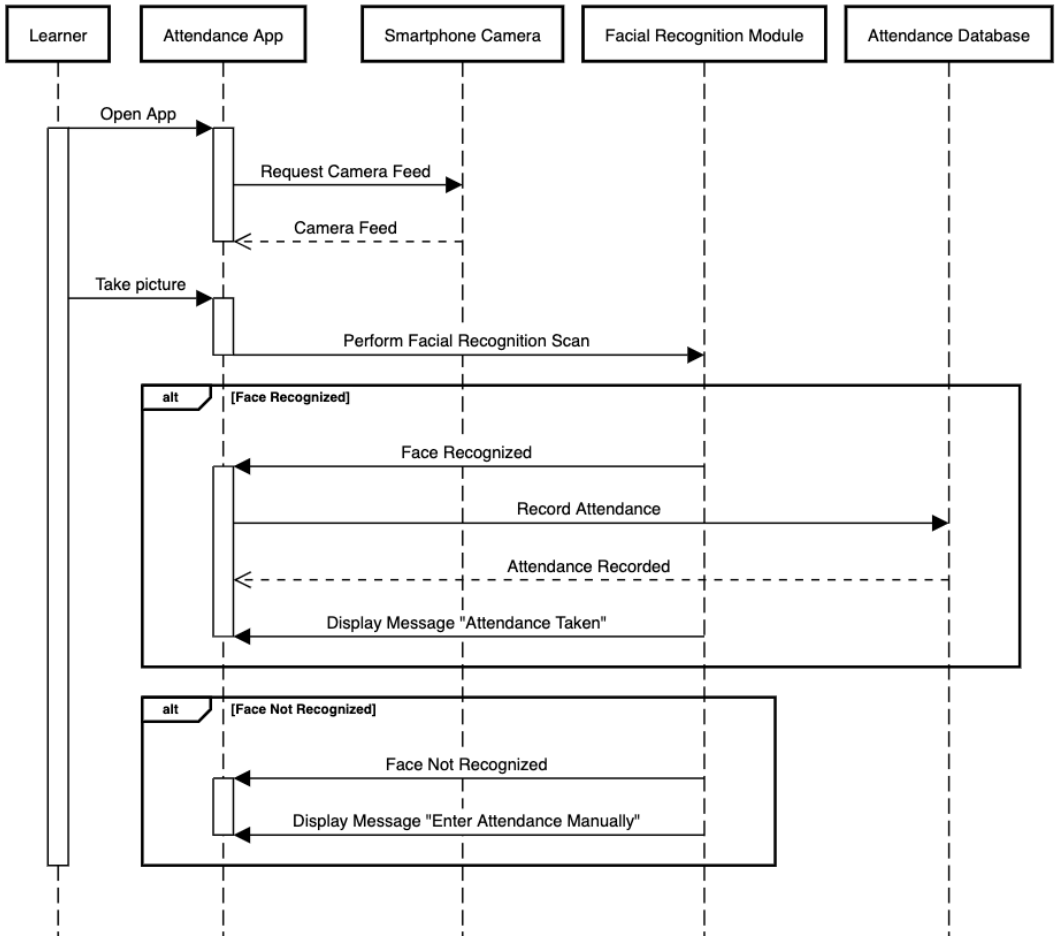

Figure 16: Sequence Diagram: Make Attendance use case

3.3-1.3. Check attendance reports:

Check Attendance Reports	
System	Web dashboard
Actor	Admin
Goal	To use the web dashboard with the aim of checking the reports of attendance.
Precondition	- The admin must have access to the attendance system and the required permissions to view reports.
Basic Flow	1. The admin logs into the attendance system. 2. The admin navigates to the reports section. 3. The admin selects the desired cohort for the attendance reports. 4. The attendance reports for the selected cohort are generated and displayed to the admin. 5. The admin can view the attendance reports for each member of the selected cohort. 6. The admin reviews the reports and ensures the accuracy of the attendance data.
Alternative Flow	**A1. At 1,** If the admin does not have sufficient permissions, they will not be able to access the reports.

Table 13: Use case Scenario: Check Attendance Reports

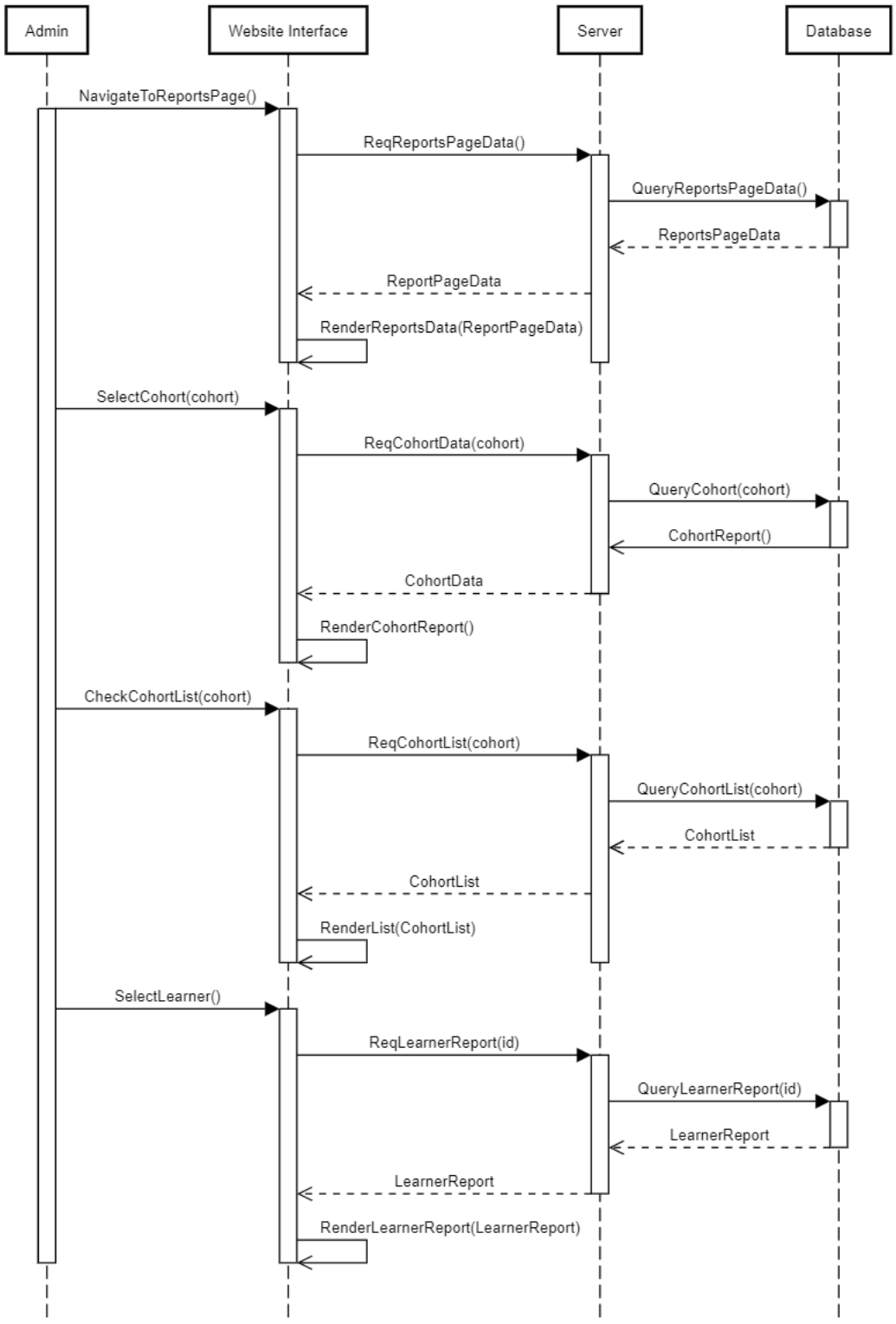

Figure 17: Sequence Diagram: Check Attendance Reports

3.3-2. Activity diagram

These diagrams illustrate the sequence of actions and the interaction between the actors and the system for both the "check attendance" and "take attendance" activities.

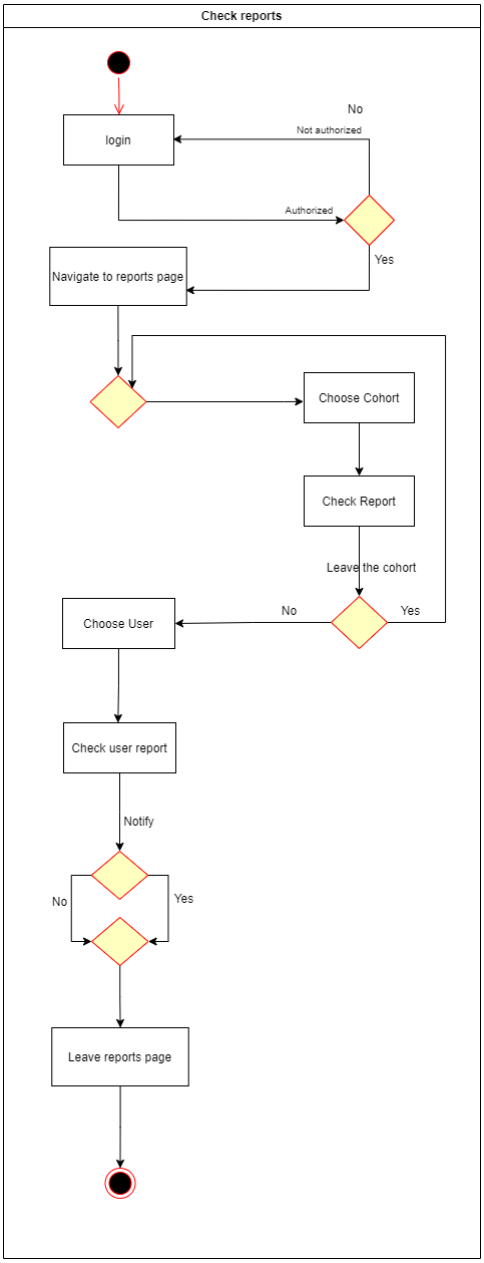

Figure 18: Activity Diagram: Check Attendance Reports

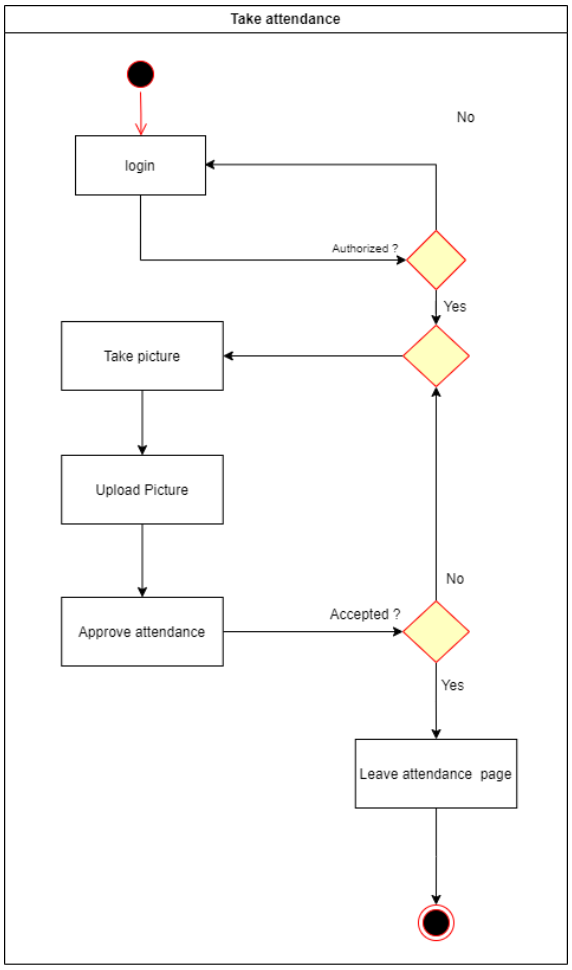

Figure 19: Activity Diagram: Take attendance

3.4- UI Design

These are the different user interface designs made for our website dashboard and mobile app.

Here is the Calendar Page of the webapp where the admin can choose a particular slot for a specific cohort.

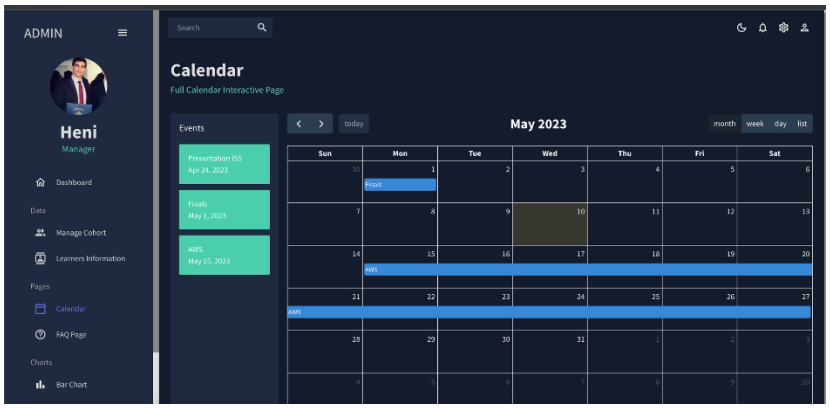

Figure 20: Calendar Page

Below, you will find the form to fill in order to create a new cohort.

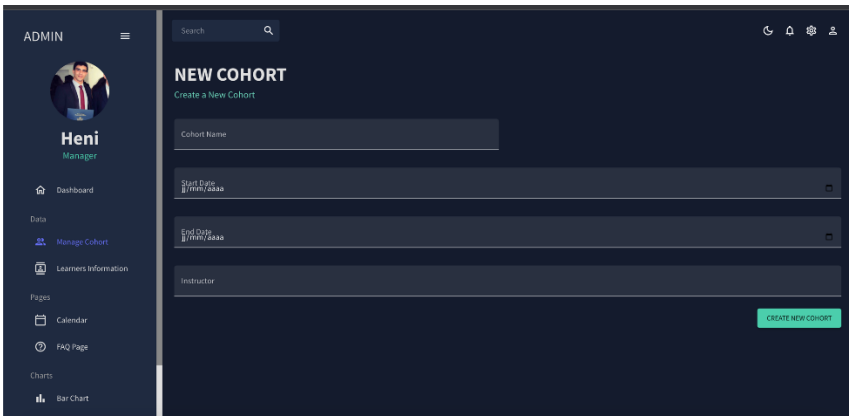

Figure 21: Create Cohort UI

Here is the list of cohorts from which the admin can edit, add and delete a cohort.

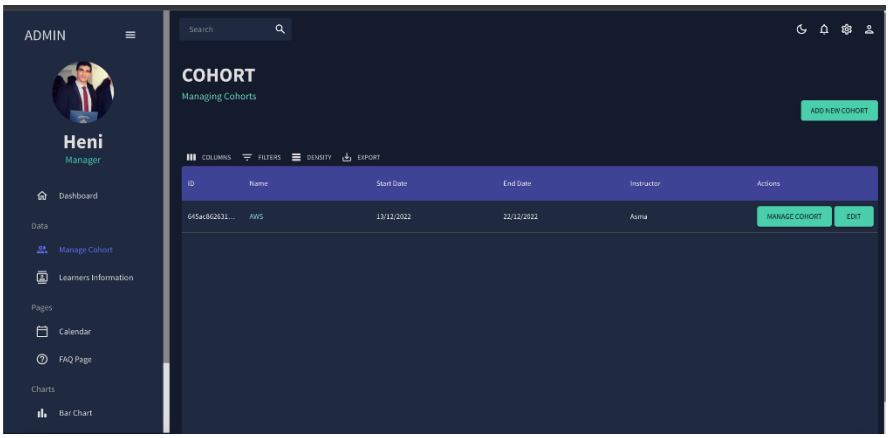

Figure 22 : Manage Cohorts

Then, you will find below the screenshots of the mobile app for the ODC learners.

First, after taking a picture the user will have to choose to either register or if he/she is already registered in the database, to upload the picture to mark his/her presence.

Figure 22: User picture

If the user is not yet registered, he will have to enter his name.

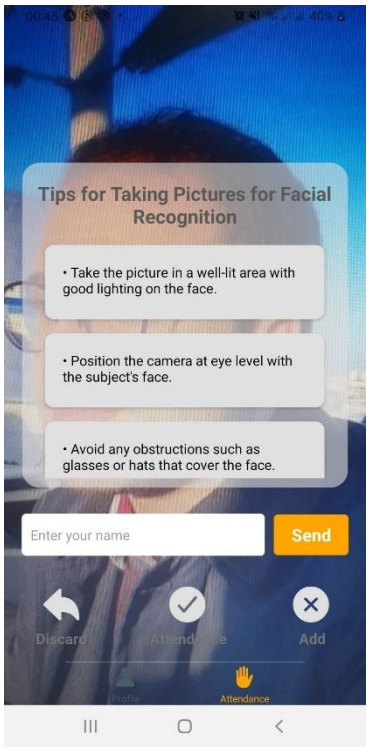

Figure 23: Sign up page

Figure 25: Loading screen

Finally, here is the confirmation page after taking the attendance.

Figure 24: Confirmation page

3.5- Conclusion

This part of the report covers a thorough study of the product design, including the usage of several diagrams. The report includes a use case diagram, a component diagram, multiple sequence diagrams, and use case scenarios. The report also contains interface drawings to help demonstrate the concept.

4- Development and Prototype Solution

4.1- Introduction

This portion of the report describes the product prototype's implementation and technology. It includes the frontend and backend technologies used, as well as the programming languages, frameworks, and libraries, as well as the development tools and database management system. The paper also looks at the integration of face recognition technology and the security elements that have been added, such as authentication, authorization, and data encryption.

4.2- Development Details

4.2.1- The Front-End:

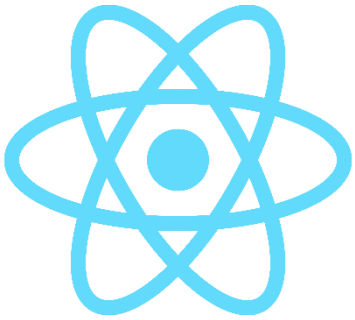

Figure 25: React Logo

We used React to build both the admin dashboard and the user interfaces for the mobile applications. We specifically picked React Native for the mobile app's design since it gives the native-like experience that our consumers anticipate. We found that the standard React library would be the most effective solution for the admin dashboard owing to its versatility and resilience. We can ensure that both interfaces are optimized for their respective platforms and provide a consistent user experience across devices by utilizing these two frameworks. [8] [9]

4.2.2- The Back-End:

Figure 26: NodeJS Logo

Figure 27: ExpressJS Logo

We used Node.js to create our server-side application, a popular runtime environment that allowed us to leverage JavaScript on the backend. We utilized Express, a popular Node web framework, to handle HTTP requests and verbs. This enabled us to quickly create a versatile API and link it with the frontend functionality. We picked Node.js and Express because of its dependability, scalability, and interoperability with a wide range of databases and third-party tools. We were able to optimize our backend development and ensure that our application could handle a high amount of requests with ease by employing these technologies. [10]

4.2.3- The Database:

Figure 18: MongoDB Logo

We choose MongoDB as our database. This utility provides a flexible data storage model that allows for modifications in document structure. We chose this database because it allows our developers to easily store structured and unstructured data while also leveraging a JSON-like format for document storage, which is excellent for our purposes. [11]

4.2.4- The Facial recognition algorithm:

In this project, we implemented a face recognition system that recognizes individuals based on their facial features. We used a pre-trained deep learning model called the Inception model, which was trained on millions of images and has achieved state-of-the-art performance on image classification and recognition tasks. Specifically, we used the Inception model to extract the features of the faces in the images, and then compared these features to those of known individuals in a database to determine who the person in the image is.

To extract the facial features, we used the pre-trained Inception model to process the images and extract a 128-dimensional encoding for each face. We then used these encodings to compare the faces in the images to those in a database of known individuals.

To implement the face recognition system, we used a distance metric called L2 distance, which calculates the Euclidean distance between two vectors. Specifically, we computed the L2 distance between the target encoding of the image and each encoding in the database. The encoding with the smallest L2 distance to the target encoding is considered the best match for the face in the image.

In addition, we added a feature that extracts the faces from an image, which involves using a pre-trained face detection model to detect the locations of the faces in the image, and then cropping and resizing the faces to a fixed size to ensure consistency in the encoding process.

Overall, our face recognition system uses a combination of deep learning models and distance metrics to recognize individuals based on their facial features. [12]

4.3- Development Support

During the development of this project, we had a dedicated team of developers, designers, and project managers working together to ensure that the application was built efficiently and met all the necessary requirements. The team utilized various tools and technologies to support the development process, including:

- **Version Control**: We used Git as our version control system, which allowed us to manage code changes effectively and collaborate on the project simultaneously.
- **Agile Methodology**: We employed Agile methodology to manage the development process, which allowed us to prioritize features and adjust the project's direction based on feedback from stakeholders and customers.
- **Project Management:** We used tools such as Trello to manage the project and track progress.
- **Communication:** We had regular meetings and communication channels established to ensure that everyone was up to date on the project's status and any issues were addressed promptly.

Overall, the development team received extensive support from the stakeholders and utilized various tools and methodologies to ensure that the application was developed efficiently and met all the necessary requirements.

4.4- Conclusion

In conclusion, our project successfully developed a robust and efficient face recognition system that leverages deep learning models and distance metrics to identify individuals based on their facial features. We used React and React Native for the front-end, Node.js and Express for the back-end, and MongoDB as our database. These technologies provided us with the tools to create a versatile and scalable application that can handle a high amount of requests with ease.

Overall, our project is a testament to the power of modern technologies and their ability to create innovative solutions to complex problems. We believe that our face recognition system can be useful in a wide range of applications, including security, surveillance, and access control, and we are excited to see how it will be used in the future.

5- Tests, Results, and discussions

5.1 Introduction

As we work towards implementing our minimal viable product, we have decided to use a black box testing technique to ensure the correctness of our features. Black box testing involves examining the functionality of the product without knowledge of its internal workings. This technique allows us to evaluate the output of the system against the expected outcomes without examining the internal code or design.

We believe that using a black box testing technique will allow us to test our product from the perspective of the end user. By focusing on the behavior of the product rather than its internal design, we can identify any functional defects that may exist. Additionally, black box testing can help us to identify any inconsistencies or issues with the user interface that may impact the user experience.

5.2 Results:

We chose to divide our testing approach across our system's modules. Each module has a distinct purpose. We created distinct test cases for each of them.

Modules	Functionalities	Test Cases	Expected Results	Result
Facial Recognition	Add new image	TC1: Upload clear face image to new user	Image successfully added to DB	Success
		TC2: Upload image without face	Invalid image	Success
		TC3: Upload blurred image	Invalid image	Success
		TC4: Upload image of existing user with other user.	User already exists	Fail
		TC5: Uploading different faces for the same user	The faces don’t match each other	Fail
	Make attendance	TC6: Upload clear image of existing user	Success in making the attendance	Success
		TC7: Upload image without face	Failing attandance and asking to retaking picture	Success
		TC8: Upload image of non-existing user	Failing attandance and asking to retaking picture	Success
		TC9: Upload image of another existing user	Failing attandance and asking to retaking picture	Fail

Authentification	Sign in	TC10:Enter valid credentials of an admin	Successfull login	Success
		TC11:Enter user's credentials in admin dashboard	Failed login	Success
		TC12: Enter invalid credentials	Failed login	Success
Cohort management	Create cohort	TC13:Enter all cohort's informations	Cohort successfully created	Success
		TC14: Keep some missing fields in the form	Unable to create cohort	Success
	Add learners	TC15: Upload CSV file of learners	Learners successfully added to the cohort	Success
		TC16: Add many learners at the same time	Learners successfully added	Success
		TC17: Add one learner	Learner successfully added	Success
	Delete learner	TC18: Select learner to be deleted and confirm	Learner successfully deleted	Success
		TC19: Select learner to be deleted and cancel	Maintain data as it was	Success
	Delete cohort	TC20: Select cohort to be deleted and confirm	Cohort successfully deleted	Success
		TC21: Select cohort to be deleted and cancel	Maintain data as it was	Success
	Edit learner	TC22: Edit learner and confirm	Learner's info successfully changed	Success
		TC23: Edit learner and cancel	Maintain data as it was	Success
	Edit cohort	TC24: Edit cohort and confirm	Cohort's info successfully changed	Success

		TC25: Edit cohort and cancel	Maintain data as it was	Success

Table 14: Black box Test Cases

We had 3 failed test cases which are:

TC4: The system would add image to the database even if it has high similarity rate with other images.

TC5: The system would add different faces to the same user.

TC9: The system would mark the other user as present.

5.3 Conclusion

Overall, we feel that black box testing is the best approach for ensuring the quality and accuracy of our minimal viable product. By using this technique, we can ensure that our product meets the needs of our users and is functional and reliable.

6- General Conclusion & Future work

In conclusion, the implementation of our attendance tracking mobile app with facial recognition technology is a significant step towards addressing the challenges faced by the Tunisian Orange Digital Center in managing attendance for its programs. By automating the attendance process and eliminating the possibility of human error and manipulation, our solution provides a more reliable and efficient way of tracking attendance. The use of image metadata to extract time and location adds an extra layer of accuracy to the attendance tracking process, ensuring that the learners are present at the Orange ODC when they are marked present. In addition, the admin dashboard web app will provide the administration with daily attendance reports and cohort management features that offer valuable insights and control over the attendance process. To develop this web application, we used various technologies; for the frontend we used ReactJS as a framework, for the backend, NodeJS as the development environment and ExpressJs as a framework then for the database, we used mongoDB. To develop the mobile app, we used React Native as framework, and for the face recognition system we used a combination of deep learning models and distance metrics to recognize individuals based on their facial features.

For our future work we would like to include a first-time login feature and liveness detection to enhance its accuracy and security. The first-time login feature will allow users to enroll by uploading their images and creating a biometric template for them. During enrollment, the system will capture multiple images of the user's face from different angles and store them in the database. This will help the system to recognize the user's face accurately when they log in to the system for the first time. The enrollment process will be quick and easy, and users will be guided through the process with clear instructions. To ensure that users upload live images, we will implement liveness detection. This feature will prompt the user to perform specific actions, such as raising their hand, smiling, or turning their head, to ensure that the image is a live image and not a static image. We will use advanced techniques such as detecting the user's pulse or asking the user to say a specific word to ensure that the image is a live image. The system will also be able to detect whether the user is wearing glasses, a hat, or a mask, and prompt the user to remove them if necessary. These features will help to improve the accuracy and security of our system, making it a reliable and efficient way to track attendance in various settings.

References

Ref.1- https://www.orangedigitalcenters.com/#

Ref.2- https://www.involvio.com/

Ref.3- https://ieeexplore.ieee.org/document/9952035?fbclid=IwAR0gW2dKdDH5UAxbgNTcrXgGxSVv3IZrmrSYBjjFVESMNmvn2Zt5ZOwrxyU

Ref.4- SMART ATTENDANCE & MANAGEMENT SYSTEM (2023)

Ref.5- https://www.redhat.com/en/devops/what-is-agile-methodology

Ref.6- https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:15288:dis:ed-2:v1:en

Ref.7- https://developer.mozilla.org/en-US/docs/Glossary/MVC

Ref.8-https://react.dev/

Ref.9-https://reactnative.dev/

Ref.10-https://nodejs.org/en

Ref.11-https://towardsdatascience.com/deep-learning-understand-the-inception-module-56146866e652

Abstract:

This end-of-year project is dedicated to improving the attendance system of Orange Digital Center through innovative solutions. Collaborating with Orange Digital Center, our team aims to address the challenges associated with traditional attendance tracking methods by implementing two primary features. The first feature involves enabling users to conveniently mark their attendance by uploading their images. Leveraging facial recognition technology, the system will accurately identify individuals and record their attendance. This eliminates the need for manual check-ins, streamlining the process and reducing errors. The second feature focuses on developing an administrative dashboard that empowers Orange Digital Center to efficiently manage cohorts. This dashboard will provide a centralized platform for cohort management, allowing administrators to track attendance records, add or remove users, and generate comprehensive reports. By implementing these features, our project aims to enhance the accuracy, efficiency, and convenience of Orange Digital Center's attendance system, ultimately improving productivity and streamlining administrative tasks.