**High-Level Design Document**

Mini Project

Contents

[1 Project Overview 2](#_Toc137748514)

[1.1 Brief introduction of the project. 2](#_Toc137748515)

[1.2 Purpose of the project. 2](#_Toc137748516)

[1.3 Goals of the project: 2](#_Toc137748517)

[2 Project Scope 3](#_Toc137748518)

[2.1.1 Platforms and devices the mobile app will support (e.g., iOS, Android). 3](#_Toc137748519)

[2.1.2 Key features and functionalities of the mobile app. 3](#_Toc137748520)

[3 Architecture and Technology Stack 3](#_Toc137748521)

[3.1.1 Overall architecture of the mobile app. 3](#_Toc137748522)

[3.1.2 Architecture 4](#_Toc137748523)

[3.1.3 Technology stack (e.g., programming languages, frameworks, libraries). 4](#_Toc137748524)

[3.1.4 Rationale behind the chosen technology stack. 5](#_Toc137748525)

[4 Mobile App Components 6](#_Toc137748526)

[4.1.1 Main components of the mobile app. 6](#_Toc137748527)

[4.1.2 Purpose of each component. 6](#_Toc137748528)

[5 User Interface Design 8](#_Toc137748529)

[5.1.1 User interface (UI) design approach. 8](#_Toc137748530)

[6 Testing and Quality Assurance 12](#_Toc137748531)

[6.1.1 Testing approach for the mobile app. 12](#_Toc137748532)

[6.1.2 Types of testing to be performed. 13](#_Toc137748533)

[6.1.3 Quality assurance processes and tools to ensure app reliability. 14](#_Toc137748534)

[7 Project Timeline and Resources 15](#_Toc137748535)

[7.1.1 Estimated project timeline, including major milestones. 15](#_Toc137748536)

[7.1.2 Roles and responsibilities of the project team members. 15](#_Toc137748537)

[8 Risks and Mitigation Strategies 15](#_Toc137748538)

[8.1.1 Potential risks and challenges associated with the project. 15](#_Toc137748539)

# Project Overview

## Brief introduction of the project.

The project focuses on the development of an Employee Training Management System (ETMS), aimed at revolutionizing the training process for interns and employees within an organization. The envisioned platform will serve as a centralized hub for accessing, tracking, and managing training activities, with the overarching goal of enhancing learning outcomes and streamlining administrative processes.

## Purpose of the project.

The purpose of this project is to empower patients and enhance their experience by providing them with a comprehensive instrument tracking solution.

The key goals of the project include:

* Streamline Training Processes: Employee Training Management System (ETMS) will centralize and automate training administration, making it easier for interns and employees to access, enroll in, and track training activities.
* Enhance Learning Outcomes: User-friendly platform with personalized training plans and assessments to improve employee engagement, knowledge retention, and skill development, ultimately driving organizational growth and success.

## Goals of the project:

* Develop an intuitive user interface: Create a user-friendly interface that is easy to navigate and understand, ensuring interns and employees can access training materials and track progress without confusion.
* Ensure cross-platform compatibility: Build a responsive web application that works seamlessly across different devices and screen sizes, allowing users to access training resources from desktops, laptops, tablets, and mobile phones.
* Enhance accessibility and availability: Ensure the web application is accessible from any location with internet connectivity, enabling interns and employees to engage in training activities at their convenience, promoting continuous learning.Deployment to App Store and Play Store.
* Facilitate real-time progress tracking: Implement features for tracking training progress and performance metrics in real-time, enabling users to monitor their learning journey, identify areas for improvement, and stay motivated.

# Project Scope

## Platforms and devices the mobile app will support (e.g., iOS, Android).

|  |  |
| --- | --- |
| Platforms | Version |
| iOS | 11 and above |
| Android | 10 and above |

## Key features and functionalities of the mobile app.

**User Registration and Authentication:**

Streamlined registration process for interns and employees to create their accounts.

Secure authentication mechanisms, including username/password

**Personalized User Profiles:**

Customized profiles for each user, containing details such as name, email, role, and department.

Ability for users to update their profile information and preferences as needed..

**Interactive Training Dashboard:**

Tailored dashboard for each user, providing an overview of their training progress, upcoming sessions, and recommended modules.

**Responsive Design:**

Mobile-friendly design to support various devices, including iPads and smartphones

Responsive user interface for optimal viewing and usability across different screen sizes

# Architecture and Technology Stack

## Overall architecture of the application.

The Employee Training Management System follows a robust and scalable architecture to ensure optimal performance, security, and usability. The architecture consists of the following components:

**Client-Side Interface:**

The client-side interface serves as the user-facing component of the Employee Training Management System (ETMS) application. It is designed to provide interns and employees with an intuitive and responsive user interface, enabling them to access their personalized dashboards, track training progress, and receive notifications.

**Server-Side Layer:**

The server-side layer hosts the application logic and processes user requests received from the client-side layer. It includes server-side programming languages (e.g., Node.js, Python) and frameworks responsible for handling business logic, authentication, authorization, and data processing. This layer interacts with the database layer to fetch or store data and responds to client requests with appropriate data or actions.

**Database Layer**:

The database layer stores and manages the application's data, including user profiles, training modules, session schedules, and assessment results. It uses database management systems (e.g., MySQL, PostgreSQL, MongoDB) that store structured or unstructured data according to the application's requirements.

## Architecture

### 

## Technology stack (e.g., programming languages, frameworks, libraries).

The technology stack for the Web Application Employee Training Management System will include the following components:

**React:**

React is a JavaScript library for building user interfaces, known for its component-based approach. It simplifies UI development by efficiently updating the DOM based on application state changes, resulting in fast and responsive web applications.

**React Navigation:**

React Navigation is a navigation library for React Native applications. It provides a flexible and customizable solution for handling navigation between screens, including stack navigation, tab navigation, and drawer navigation.

**Axios:**

Axios is a widely used JavaScript library for making HTTP requests from the application to the server. It simplifies the process of sending and receiving data, handling API calls, and managing network requests in the mobile app.

**Material-UI:**

Material-UI is a popular React component library that implements Google's Material Design principles. It provides a vast collection of pre-designed and customizable UI components for building modern and visually appealing web applications. Material-UI simplifies the development process by offering ready-to-use components such as buttons, cards, and menus, allowing developers to create consistent and responsive user interfaces with ease.

## Rationale behind the chosen technology stack.

The selection of the technology stack for the Employee Training Management System (ETMS) web application is based on several key considerations:

**Cross-Platform Compatibility:**

React, chosen for its ability to build user interfaces, ensures cross-platform compatibility by enabling the creation of web applications that can be accessed seamlessly across different devices and operating systems. This eliminates the need for separate development efforts for various platforms, optimizing user accessibility and reducing development complexity.

**Time and Cost Efficiency:**

The selection of React and associated libraries such as React Navigation and Material-UI contributes to time and cost efficiency. React's component-based architecture streamlines development processes, allowing for rapid prototyping, iterative development, and efficient code maintenance. Additionally, the availability of pre-designed UI components in Material-UI accelerates development cycles, reducing time-to-market and overall development costs.

**Native-Like Performance:**

React Native allows us to develop mobile applications with native-like performance and user experience. It achieves this by utilizing native components and APIs specific to each platform, resulting in smooth animations, fast response times, and seamless integration with device features. This native performance enhances the overall usability and responsiveness of the instrument tracking application.

**Large Developer Community and Ecosystem:**

React Native boasts a vast and active developer community. This thriving community provides extensive support, resources, and libraries that accelerate development and problem-solving. The availability of a wide range of open-source libraries and components allows us to leverage existing solutions and integrate additional functionalities efficiently.

**Code Reusability:**

The chosen technology stack, centered around React, promotes code reusability through its modular architecture and component-based design. Components developed using React can be easily reused across different parts of the application, minimizing redundant code and enhancing development efficiency. This not only reduces development time but also ensures consistency and scalability in code management.

**Flexibility and Customizability:**

React and its associated libraries offer a high degree of flexibility and customizability, allowing developers to tailor the ETMS application to specific requirements and user preferences. React's ecosystem provides a wide range of third-party libraries and tools that enable developers to extend functionality, integrate with external services, and customize user interfaces according to project needs.

# Mobile App Components

## Main components of the mobile app.

* react
* Material-UI
* React Router
* React Date Picker
* React Icons
* moment
* axios
* Formik
* Big Calendar

## Purpose of each component.

### React

Purpose: React is the core library used for building user interfaces in the ETMS web application. It facilitates the creation of reusable UI components and manages the rendering of these components efficiently.

### Material-UI:

Material-UI provides a library of pre-designed and customizable UI components following Google's Material Design principles. It ensures a consistent and visually appealing user interface across the ETMS web application.

### React Router:

React Router is responsible for managing navigation within the ETMS web application. It enables routing between different pages and components, ensuring smooth transitions and a seamless user experience.

### React Icons:

React Icons offers a collection of icons for use in the ETMS web application. It enhances the visual elements of the user interface by providing a variety of icons for buttons, menus, and other UI components, improving accessibility and user interaction.

### Moment

The moment package is a widely used and powerful JavaScript library for handling and manipulating dates and times. It provides an extensive range of functionalities to parse, format, manipulate, and display dates and times in various formats.

### React Native Axios

The react-native-axios package is a lightweight JavaScript library that provides an easy-to-use API for making HTTP requests in React Native applications. It is based on the popular axios library and offers a simplified syntax for performing HTTP operations such as GET, POST, PUT, DELETE, and more.

### Formik

The Formik npm package is a popular and flexible form management library for React applications. It simplifies the process of building and handling forms by providing a set of powerful features and utilities.

# User Interface Design

### User interface (UI) design approach.

|  |  |
| --- | --- |
| **Splash Screen:**  A screen shot of a cell phone  Description automatically generated with medium confidence | **Walk Through Screen:**  A screen shot of a cell phone  Description automatically generated with medium confidence |
| 1. The mobile splash screen serves as the initial visual introduction to the app, representing its branding and visual identity. 2. It provides feedback to the user that the app is loading and initializing, improving the perception of a smooth transition. 3. The duration of the splash screen should be optimized to balance loading time and user experience, avoiding unnecessary delays. 4. Incorporating brand messaging and a clear call-to-action on the splash screen can enhance user engagement and guide them towards key app features. 5. Designing the splash screen to adapt to different screen sizes and orientations ensures a consistent and visually pleasing experience across devices. | 1. Walkthrough screens provide a guided introduction to the app's features, functionality, and user interface. 2. They help new users understand how to navigate and interact with the app, reducing the learning curve. 3. Walkthrough screens can showcase key app benefits, highlight important features, and set user expectations. 4. Interactive elements, such as swiping or tapping, can be incorporated into walkthrough screens to engage users and encourage exploration. 5. Designing visually appealing and intuitive walkthrough screens can leave a positive first impression and increase user retention. |

# Testing and Quality Assurance

Testing and quality assurance are essential processes in software development that aim to ensure the reliability, functionality, and overall quality of a software product. These processes involve systematically examining and validating the software at various stages to identify defects, bugs, and any deviations from the expected behavior. By conducting thorough testing and quality assurance, developers can uncover and rectify issues early in the development cycle, enhancing user satisfaction, minimizing risks, and ultimately delivering a stable and dependable software product to the end users.

### Testing approach for the mobile app.

When it comes to testing a mobile application, several approaches can be employed to ensure its quality and functionality. Here is the testing approach for a mobile application:

1. Requirement Analysis: Understand the requirements of the mobile application thoroughly, including its intended functionality, target audience, supported platforms (iOS, Android, etc.), and any specific device requirements.
2. Test Planning: Develop a comprehensive test plan that outlines the testing objectives, scope, test environments, test cases, and testing techniques to be used. Determine the types of testing to be performed.
3. Test Environment Setup: Prepare the necessary test environments, including physical devices, emulators, simulators, or cloud-based testing platforms. Ensure that the test environments closely match the actual user environment.
4. Functional Testing: Verify that the mobile application meets the specified functional requirements. Test various features and functionalities of the application, such as user interface, navigation, user input validation, data processing, and integration with backend services.
5. User Interface Testing: Test the mobile application's user interface (UI) to ensure consistency, responsiveness, and adherence to platform-specific design guidelines. This involves checking for proper alignment of elements, accurate rendering of fonts and images, correct color schemes, and intuitive navigation across different devices and screen sizes.

### Types of testing to be performed.

The following types of testing should be carried out in the project:

**Functional Testing:**  
In functional testing, each function in the application tested by giving the input value, determining the output, and verifying the actual output with the expected value. This testing examines the

**Outcome**:

* It ensures that the customer or end-user is satisfied.
* It ensures the all the requirements should be met.
* It ensures the proper working of all the functionalities of an application/software/product.

1. **Usability Testing:**  
   Conduct thorough compatibility testing across different mobile platforms, screen resolutions, and device configurations to identify and address compatibility issues proactively.  
     
   **Outcome**:

* User satisfaction: Measure the level of user satisfaction and identify any areas where users may face difficulties or frustrations while interacting with the mobile application.
* Navigation and intuitiveness: Evaluate the ease of navigation within the mobile application and assess how intuitive the user interface is for performing tasks and accessing different features.
* Learnability: Assess how quickly users can understand and learn to use the mobile application, especially for first-time users, by evaluating the clarity of instructions and the simplicity of the interface.
* Error frequency and severity: Identify the frequency and severity of user errors encountered during usability testing, noting any critical or recurring issues that may hinder user experience.

1. **System Testing:**  
   System Testing (ST) is to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.   
     
   **Outcome**:

* It covers end to end testing
* System tests is used to specify how the application should behave

1. **Retesting:**  
   Re-testing is executing a previously failed test against new software to check if the problem is resolved. After a defect has been fixed, re-testing is performed to check the scenario under the same environmental conditions.   
     
   **Outcome:**

* It verifies that the problem has been resolved and that everything is operating as intended
* It raises the applications or product’s quality

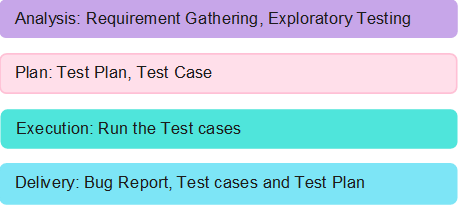
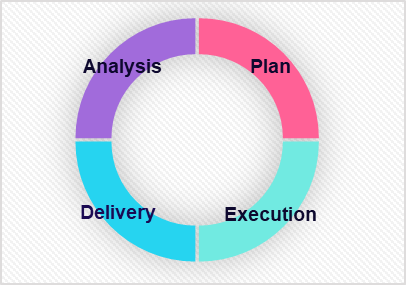
**Regression Testing:**  
Regression testing is responsible for the overall stability and functionality of the existing features. It can be performed on a new build when there is a significant change in the original functionality.  
  
**Outcome:**

* Regression testing ensures a fix does not adversely impact the existing functionality
* Regression testing outweigh the risks of skipping it

1. **Compatibility Testing:**  
   Compatibility Testing is a type of testing to check whether your software is capable of running on different hardware, operating systems, applications or Mobile devices.  
     
   **Outcome:**

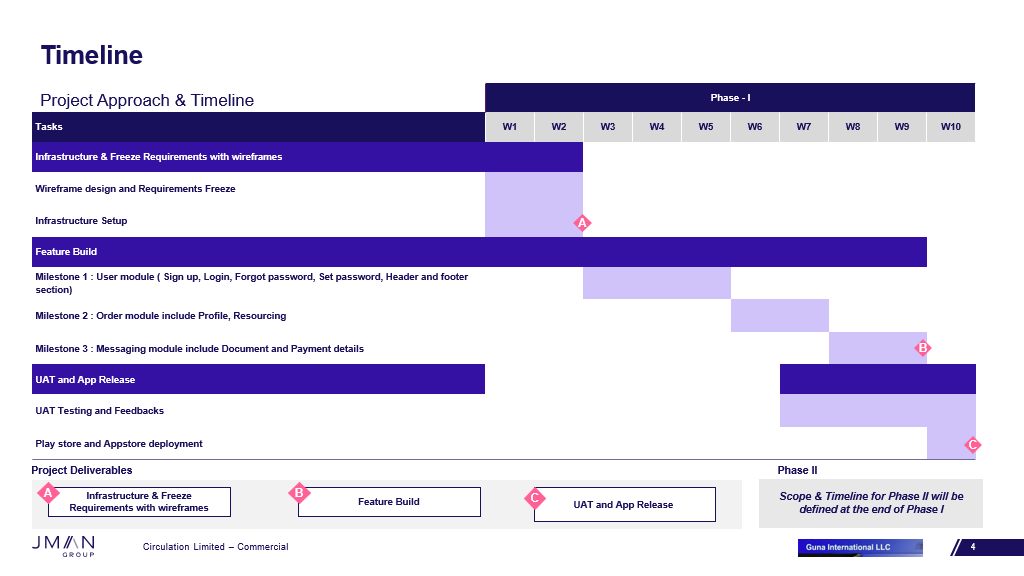
* It ensures that a mobile application functions correctly on different devices and screen sizes.
* It helps identify and resolve compatibility issues early in the development cycle, minimizing user complaints and negative reviews.
* By conducting compatibility testing, developers can ensure a seamless user experience across various platforms, enhancing customer satisfaction.
* The outcome of compatibility testing is a robust and reliable mobile application that works consistently across a wide range of devices, improving its market acceptance and user adoption.

### Quality assurance processes and tools to ensure app reliability.

The objective of the test is to define the goals and purpose of the testing effort. It aims to provide a comprehensive and focused statement of what is to be accomplished through testing. The test objectives serve as a guiding principle for the testing activities and ensure that they are aligned with the overall objectives of the project  
  


# Project Timeline and Resources

### Estimated project timeline, including major milestones.



### Roles and responsibilities of the project team members.

A screenshot of a computer

Description automatically generated with low confidence

# Risks and Mitigation Strategies

### Potential risks and challenges associated with the project.

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | Risk / Challenges | Impact | Mitigation Plan |
| 1 | If there is a delay in API | Project deliverables will be delayed | Need to have all the API ahead of development team |
| 2 | Application version limitation |  | App will support only versions which is above |

**Appendix Title**

Document Title