Contents

[Project steps 2](#_Toc168137197)

[Phase 1: Planning 2](#_Toc168137198)

[Phase 2: Development 2](#_Toc168137199)

[Phase 3: Testing 3](#_Toc168137200)

[Phase 4: Deployment 4](#_Toc168137201)

[Phase 5: Maintenance 4](#_Toc168137202)

[Documentation 5](#_Toc168137203)

[Outlines 5](#_Toc168137204)

[Suggested Order and Timeline 6](#_Toc168137205)

[Detailed Outline of Each Document 6](#_Toc168137206)

# Project steps

## Phase 1: Planning

1. **Requirement Analysis**
   * Define the features and functionalities of the application.
   * Identify user roles (researcher, admin).
2. **Project Setup**
   * Set up a project repository (e.g., GitHub, GitLab).
   * Configure Expo for React Native.
3. **Design**
   * Create wireframes and UI/UX designs for the application.
   * Define the data model for psychometrics and surveys.

## Phase 2: Development

**2.1 User Authentication and Authorization**

1. **User Authentication**
   * Implement user registration and login using AWS Cognito.
   * Integrate social media login options (optional).
2. **User Authorization**
   * Set up role-based access control (RBAC) for researchers and admins.

**2.2 Psychometrics Management**

1. **Create Psychometric Models**
   * Define the structure of a psychometric (questions, options, scoring).
   * Develop backend APIs to create, read, update, and delete psychometrics.
2. **Psychometric Creation UI**
   * Build frontend components for researchers to create and manage psychometrics.

**2.3 Survey Management**

1. **Create Survey Models**
   * Define the structure of a survey (linking multiple psychometrics).
   * Develop backend APIs to create, read, update, and delete surveys.
2. **Survey Creation UI**
   * Build frontend components for researchers to create and manage surveys based on psychometrics.

**2.4 Survey Distribution**

1. **Generate Survey Links**
   * Implement functionality to generate unique survey links.
   * Develop backend API to handle survey link generation.
2. **Survey Response Handling**
   * Create APIs to handle survey responses from applicants.
   * Develop frontend components for applicants to take surveys.

**2.5 Statistical Analysis**

1. **Data Collection and Storage**
   * Implement data storage solutions for survey responses (AWS DynamoDB or RDS).
2. **Statistical Analysis Tools**
   * Develop backend APIs for statistical analysis.
   * Integrate libraries for data analysis (e.g., Python with AWS Lambda).
3. **Statistics UI**
   * Build frontend components for researchers to view and analyze survey statistics.

**2.6 Payment Integration**

1. **Payment Gateway Integration**
   * Integrate a payment gateway (e.g., Stripe) for paid statistics features.
2. **Payment Handling UI**
   * Build frontend components to handle payments and subscriptions.

## Phase 3: Testing

1. **Unit Testing**
   * Write unit tests for frontend and backend components.
2. **Integration Testing**
   * Test the integration of different modules (authentication, survey creation, response handling).
3. **User Acceptance Testing**
   * Conduct UAT with a group of researchers to gather feedback.

## Phase 4: Deployment

1. **CI/CD Pipeline**
   * Set up a CI/CD pipeline using AWS services (CodePipeline, CodeBuild).
2. **Deploy to AWS**
   * Deploy the application to AWS (EC2, S3 for web, etc.).
   * Configure AWS Amplify or similar for seamless deployment.

## Phase 5: Maintenance

1. **Monitoring and Logging**
   * Set up monitoring and logging using AWS CloudWatch.
2. **Bug Fixes and Updates**
   * Regularly update the application based on user feedback and new requirements.
3. **Documentation**
   * Maintain comprehensive documentation for users and developers.

# Documentation

## Outlines

1. **Project Plan**
   * **Purpose**: To outline the overall project goals, phases, timelines, resources, and milestones.
   * **When to Start**: At the beginning of the project (Phase 1: Planning).
2. **Requirement Analysis Document**
   * **Purpose**: To detail the features, functionalities, and user roles of the application.
   * **When to Start**: Early in Phase 1 (Planning) after initial project planning.
3. **Design Documentation**
   * **Purpose**: To describe the architecture, components, interfaces, and data flow of the system.
   * **When to Start**: During Phase 2 (System Design).
4. **Development Documentation**
   * **Purpose**: To provide detailed instructions for developers, including coding standards, API documentation, and development guidelines.
   * **When to Start**: During Phase 3 (Implementation).
5. **Testing Documentation**
   * **Purpose**: To outline test plans, test cases, testing procedures, and test results.
   * **When to Start**: During Phase 4 (Testing).
6. **User Documentation**
   * **Purpose**: To provide end-users with instructions on how to use the application, including user manuals, help guides, and FAQs.
   * **When to Start**: Towards the end of Phase 3 (Implementation) and refined during Phase 4 (Testing).
7. **Maintenance Documentation**
   * **Purpose**: To keep comprehensive records of the system's updates, bug fixes, and maintenance activities. This includes change logs, issue tracking, and developer notes.
   * **When to Start**: During Phase 5 (Maintenance) and continuously updated.

## Suggested Order and Timeline

1. **Project Plan**: Start and complete during the initial project kickoff.
2. **Requirement Analysis Document**: Start early in Phase 1 and complete before moving to the design phase.
3. **Design Documentation**: Begin once requirements are finalized; complete before implementation starts.
4. **Development Documentation**: Create during the implementation phase, continuously updated as development progresses.
5. **Testing Documentation**: Prepare alongside development and complete before final testing.
6. **User Documentation**: Draft during implementation, finalize during testing, and update as needed.
7. **Maintenance Documentation**: Begin during the maintenance phase and update regularly.

## Detailed Outline of Each Document

**Project Plan**

* Introduction
* Objectives
* Scope
* Phases and Milestones
* Resources
* Timeline
* Risk Management

**Requirement Analysis Document**

* Introduction
* Features and Functionalities
* User Roles
* Use Cases
* Requirements Specification

**Design Documentation**

* System Architecture
* Component Design
* Data Flow Diagrams
* Interface Specifications

**Development Documentation**

* Coding Standards
* API Documentation
* Development Guidelines
* Database Schema

**Testing Documentation**

* Test Plan
* Test Cases
* Testing Procedures
* Test Results

**User Documentation**

* User Manuals
* Help Guides
* FAQs
* Troubleshooting Tips

**Maintenance Documentation**

* Change Logs
* Issue Tracking
* Bug Fixes
* Update Notes