**Test plan for:**

**<< Lumi Express Team 2>>**

Changelog

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| 0.1 | 31.10.2020 | JK | Initial document creation |
| 0.2 | 08.12.2023 | Team2 | Added testing information for React components |
| 0.3 | 13.13.2923 | Team2 | Added the API test information |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Terms/Acronyms 3](#_Toc85035757)

[2. Introduction 4](#_Toc85035759)

[2.1. Scope 4](#_Toc85035760)

[2.1.1. In scope 4](#_Toc85035761)

[2.1.2. Out of scope 4](#_Toc85035762)

[2.2. Quality Objective 4](#_Toc85035763)

[2.3. Roles and Responsibilities 4](#_Toc85035764)

[3. Test methodology 5](#_Toc85035765)

[3.1. Overview 5](#_Toc85035766)

[3.2. Test levels 5](#_Toc85035767)

[3.3. Bug triage 5](#_Toc85035768)

[3.4. Suspension Criteria and Resumption Requirements 5](#_Toc85035769)

[3.5. Test Completeness 5](#_Toc85035770)

[4. Test Deliverables 7](#_Toc85035773)

[5. Resource & Environment Needs 8](#_Toc85035774)

[5.1. Test Environment 8](#_Toc85035775)

[5.2. Testing Tools 8](#_Toc85035776)

1. Terms/Acronyms

A mention of any terms or acronyms used in the project

|  |  |
| --- | --- |
| TERM/ACRONYM | DEFINITION |
| API | Application Program Interface |
| Jest | JavaScript Testing Framework for React applications |
| RESTful | Representational state transfer |
| NPM | Node Package Manager |
| HTTP | Hypertext Transfer Protocol |
|  |  |
|  |  |



# Introduction

The project involves a React and Express application. The Express part doesn't have any tests due to its simplicity. The React part consists of three separate applications, with the "consumer app" being the most complex. The testing focus is primarily on base components like Input, Button, and Modal, using the Jest testing framework.

The main objective is to establish a testing approach with thorough coverage for both API (backend) and React (frontend) components. The strategy involves utilizing a mix of unit tests and potentially integration tests.

## Scope

### In scope

Testing for React components (e.g., Input, Button, Modal) in the consumer app.

Testing API endpoints for user login, ensuring that user authentication is secure and reliable.

### Out of scope

Testing for Express application (not performed due to simplicity).

## Quality Objective

Ensure the consumer app's base components conform to functional requirements.

Identify and fix any bugs or issues in the tested React components before deployment.

## Roles and Responsibilities

Ashif Moon: Ashif Handles the development of fundamental components for the backend driver app, and refining the UI design of the consumer app.

Lufei Wu: Lufei Wu takes responsibilities for the frontend development of the consumer App, getting password for the cosumer app and API tests.

Minyi Zhang: Minyi Zhang is charge in the frontend development of the touch screen App, Authentication for the consumer app and React tests.

Ziqi Li: Ziqi Li holds the reins in backend development, the dashboard page for both the driver and consumer.

In the development of our team's project, each member has assumed primary responsibilities. Simultaneously, everyone is engaged in tasks such as searching and discussing details taking on distinct functions in apps. We collaborate to aid the main responsible person in problem-solving when necessary. Everyone has been active equally.

# Test methodology

## Overview

The project follows an Agile methodology to meet the tight development schedule.

The development process involves short one-week sprints to quickly deliver starter-kits for various services in both frontend and backend.

## Test levels

Component Testing (focused on React components in the consumer app).

## Bug triage

Bugs will be triaged regularly during sprint reviews to determine resolution types.

Prioritization of bugs will align with sprint goals and client requirements.

## Test Completeness

100% test coverage

All Manual & Automated Test cases executed



# Test Deliverables

Here mention all the Test Artifacts that will be delivered during different phases of the testing lifecycle.

**Test Plan** (this document filled)\*

\*Documents that are in bold are minimum requirement for this course!

**Test Plan: Button and Modal components**

Testing is vital for web applications' reliability and functionality. This plan details a comprehensive strategy using Jest, a JavaScript testing framework, to examine button and modal components. Jest excels in unit and integration testing, offering a solid foundation for thorough testing.

We set the testing environment setup by installing jest first: “npm install --save-dev jest”. For button component testing, we confirm the button renders correctly to the rendering test. Verify the button's click event to click event test.

Ensure the button behaves correctly in different states to test button states. for the Modal Component, we also Confirm the modal renders for the Rendering Test.

Then we Test modal opening and closing functionality. Finally, we test content display. We also need Verify the interaction between the button and modal components to test integration.

To enhance efficiency, set up automation to run Jest tests automatically on code changes. Additionally, generate detailed test reports encompassing results and coverage.

In conclusion, this concise Jest-based test plan ensures an effective approach to button and modal component testing. Regular updates accommodate changes, and thorough documentation ensures reliability. Adhering to these procedures instills confidence in the web application's essential elements.

**Test Plan: User Login Endpoint**

4.1 Objective:

Ensure the user login endpoint functions correctly, securely, and meets the specified requirements.

* 1. Tools:

Jest (JavaScript testing framework)

4.3 Test Scenarios:

**Successful Login**: a. Test Valid Credentials: Verify that a user can successfully log in with valid credentials.

**Invalid Login Attempts**: a. Test Invalid Username: Verify that login fails with an invalid username.

b. Test Invalid Password: Verify that login fails with an invalid password.

**Security and Validation**: a. Test Input Validation: Ensure the login endpoint properly validates input data (e.g., username and password).

b. Test Rate Limiting: Verify that the login endpoint has rate limiting to prevent brute force attacks.

**Session Management**: a. Test Token Generation:

Verify that a secure authentication token is generated upon successful login.

b. Test Token Expiry: Ensure that authentication tokens have a reasonable expiration time.

**Execution:**

Run the Jest test suite using the appropriate commands.

Monitor test results and address any failures or errors.

Regular updates accommodate changes, and thorough documentation ensures reliability.

# Resource & Environment Needs

## Test Environment

React development environment.

Jest testing framework.

## Testing Tools

Jest for React component testing.