<Customer Name>

ECHO

Test Strategy and Plan

1.0

	Prepared By / Last Updated By	Reviewed By	Approved By
Name			
Role			
Signature			
Date			

Table of Contents

Purpose of This Document

The purpose of this document is to decide and define the testing approach for Echo Social Media Application. This document is intended for project stakeholders, namely Development team, Testing Team and Trainer(s).

This document will aim to:

- Define the processes, approach & Methodology to be followed for this project.
- Define and communicate the scope and boundaries of testing.
- Outline the definition and entry and exit criteria, assumptions, dependencies, schedule and also roles and responsibility of the resources.
- Provide Reference on Definition of Ready and Definition of Done as applicable.
- Define deliverables across all Sprints / Iteration/ Releases and phases within it.
- Identify the Infrastructure, Tools and Test Environments for all types of testing.
- Identify how defects should be classified and handled throughout defect lifecycle.

1.0 Introduction

The purpose of this QA Testing Strategy document is to outline the approach and practices that will be adopted to ensure the successful delivery of a high-quality social media application—Echo. Echo is designed to provide users with a seamless and engaging platform for sharing content, discovering trends, and building meaningful connections. Given the complexity and user-centric nature of the application, a robust testing framework is essential to validate that all features function as intended and meet the defined business requirements. This document details the scope of testing, methodologies—including both manual and automated testing using Selenium with Java—tools, resources, and timelines that will guide the QA process throughout the development lifecycle. The strategy aims to identify and resolve defects early, ensure system stability, and deliver a reliable and intuitive user experience upon release.

1.1 Application Overview

1.1.1 Purpose:

Echo is designed to be a user-centric social media platform that promotes real-time interaction, community engagement, and content discovery. It aims to reduce digital friction and enhance user experience through intuitive design and rich features.

1.1.2 Target Audience:

- Development Team
- QA Team
- Business Analysts
- Database Team

1.1.3 Key Functional Modules

- Login & Registration
- Home Page
    - Left Sidebar: Navigation and user profile.
    - Top Bar: Search, profile, messages, settings, logout, and theme toggle.
    - Main Feed: Stories, new post creation, and timeline.
    - Right Sidebar: Suggestions, trending topics, and notifications.
- Search Feature
- Suggestions Box
- Explore Box
- Notifications Box
- Followers/Following
- Profile Page
- Follower/following counts and user posts with engagement metrics.
- Update Profile
- Upload options for cover and profile pictures

2.0 Testing Objectives

The primary objective of this QA strategy is to validate that all functional and non-functional requirements are implemented correctly and accepted prior to release. This includes ensuring that the system behaves as expected under various conditions, meets usability standards, and is free from critical defects. The strategy outlines the testing scope, methodologies, tools, and resources that will be employed to achieve high-quality delivery.

2.1 UI/UX Validation

- Ensure all visual elements (logos, icons, buttons, profile pictures, etc.) are displayed correctly.
- Validate layout consistency across different screen sizes and devices.
- Confirm that placeholder texts and labels are accurate and user-friendly.

2.2 Functional Accuracy

- Verify that all interactive elements (buttons, toggles, input fields) perform their intended actions.
- Ensure that validations (e.g., character limits, empty input restrictions) are enforced correctly.

- Confirm that non-interactive sections (e.g., hashtags, notifications placeholder) remain static and do not respond to user actions.

2.3 Data Integrity

- Ensure that user-generated content (posts, images, tags) is correctly saved and displayed.
- Validate that user suggestions and feed content are fetched and rendered accurately.

2.4 Navigation and Flow

- Confirm that clicking on icons (Profile, Messages, Settings, Logout) leads to the correct pages or actions.
- Ensure smooth transitions between different sections of the homepage.

2.5 Accessibility and Usability

- Validate that the application is usable with keyboard navigation and screen readers.
- Ensure that color schemes (especially in dark mode) maintain readability and accessibility standards.

2.6 Error Handling

- Check that invalid inputs (e.g., empty post, overly long text) trigger appropriate error messages.
- Ensure that the application handles missing data (e.g., no notifications) gracefully.

2.7 Performance and Responsiveness

- Verify that UI elements load quickly and interactions are responsive.
- Ensure that the feed and suggestions sections update without noticeable lag.

3.0 Testing Scope

3.1 In Scope

3.1.1 Program/Application / Module

The scope of testing for this phase of the Echo social media application is limited to three primary modules:

- Login Module: Validates user authentication, input validations, error handling, and navigation to the home page upon successful login.
- Registration Module: Ensures correct user account creation, field validations, error messaging, and redirection to the login page.
- Home Page Module: Covers post creation, feed rendering, basic UI elements, and navigation links within the home interface.

These modules represent the foundational user journey and are critical to the initial release of the application.

3.1.2 Type of Testing

The following types of testing will be conducted:

- Manual Testing: Used for functional validation, UI/UX checks, and exploratory testing across the in-scope modules.
- Automation Testing: Implemented using Selenium WebDriver with Java to automate critical workflows such as login, registration, and post creation on the home page.
- Basic Non-Functional Testing: Includes minimal checks for responsiveness, browser compatibility, and basic performance under normal usage conditions.

3.1.3 Other In-Scope Items>>

- Validation of input fields, error messages, and form behaviors.
- Navigation between login, registration, and home page.
- Verification of UI elements and layout consistency across supported browsers.
- Execution of automated test scripts for regression and smoke testing.
- Logging and tracking defects using the designated bug tracking tool.

3.2 Out of scope

3.2.1 Program/Application / Module

The following modules of the Echo social media application are excluded from the current QA testing scope:

- Search Feature

- Suggestions Box
- Explore Box
- Notifications Box
- Followers/Following
- Profile Page and Update Profile Functionality

These modules will not be tested in this phase and are planned for future iterations or separate validation cycles.

3.2.2 Type of Testing

The following types of testing are out of scope for this phase:

- Advanced Non-Functional Testing: This includes performance benchmarking, load testing, stress testing, and scalability assessments.
- Security Testing: Penetration testing, vulnerability scanning, and data protection validation are not included.
- Accessibility Testing: Compliance with accessibility standards (e.g., WCAG) is not part of the current scope.
- Localization/Internationalization Testing: Multilingual support and region-specific content validation are excluded.

3.2.3 Other Out-of-Scope Items>>

- Integration testing with third-party services or APIs.
- End-to-end testing across all modules.
- Mobile app testing on physical devices (only basic browser-based checks are performed).
- UI/UX feedback cycles beyond basic layout and navigation validation.

4.0 Testing Approach & Strategy

This section elaborates on the overall test approach to be followed for the Echo social media application, covering various aspects of the testing lifecycle.

The overall test approach for the Echo application will be a hybrid strategy, combining comprehensive **manual testing** with targeted **automated testing**. We maintain a detailed manual testing document that provides structured views for various test scenarios, test cases, Requirements Traceability Matrix (RTM), and mappings to defect IDs. This ensures thorough coverage and traceability for all functional and non-functional requirements

4.1 Feature Testing Approach & Coverage

Our feature testing approach relies on a **comprehensive manual testing document**. This document serves as the central repository for:

- **Test Scenarios**: High-level descriptions of functionalities to be tested.
- **Test Cases**: Detailed, step-by-step instructions for executing tests, including preconditions, inputs, expected outcomes, and postconditions.
- **Requirements Traceability Matrix (RTM)**: A matrix linking each functional requirement from the FRD to one or more test cases, ensuring 100% coverage of specified requirements.
- **Required Mappings**: Links between test cases and specific modules or features.
- **Defect IDs**: Direct mapping of test case failures to reported defects for clear tracking.

Manual testing ensures in-depth exploration of features, user experience validation, and handling of complex scenarios that might be difficult to automate. Coverage targets all functional areas outlined in the FRD, including:

- Login Page
- Registration Page
- Home Page (Left Sidebar, Top Bar, Main Content Area, Right Sidebar)

4.2 Test Automation Approach & Coverage

Overall Automation Strategy and Approach:

Our automation strategy focuses on building a robust and maintainable automated test suite using **Selenium with Java**. This approach aims to accelerate regression testing cycles and provide quick feedback on the health of the application after code changes.

4.3 Test Data Management

The overall Test Data Management (TDM) approach will focus on ensuring the availability of relevant, realistic, and reusable test data for all testing phases.

Test Data Provisioning Source:
Test data will be provisioned from a combination of sources:

- **Manual Creation**: For specific, complex scenarios or edge cases.
- **Database Dumps/Subsets**: Anonymized production data subsets where appropriate, to ensure realism.

Excel Files: Test data will be stored in Excel files, serving as a structured source for test inputs.Testing Phases, Scope, Deliverables, Acceptance Criteria, Impacted Applications

4.3.1 Testing Phases & Scope:

- **Smoke / Sanity Testing**: Initial build verification.
- **UI Testing**: User interface and responsiveness validation.
- **Functional Testing**: Feature-by-feature validation against FRD.
- **E2E Testing**: Critical business flow validation.
- **Regression Testing**: Ensuring no new defects are introduced.
- **Positive / Negative Testing**: Validating expected and error behaviors.

4.4 Test Deliverables:

The following deliverables will be created and maintained throughout the project lifecycle:

- **Project Level Test Strategy and Test Plan (this document)**: Defines 'what' will be done from a testing perspective for the entire project and 'how' the testing strategy will be implemented across the project. It governs the test plan for each test type.
- **Test Cases**: Functional test cases based on user story acceptance criteria/requirements. Test Cases would be developed covering all business flows and test scenarios to enable higher coverage for testing, ensure traceability from requirements to release, and facilitate automated regression. These are maintained in a comprehensive manual testing document.
- **Test Suites**: During test execution, test cases will be grouped into logical suites for efficient execution and reporting.
- **Test Execution Reports**: Provides the status of test execution at both the sprint and release level, detailing pass/fail rates and executed test cases.
- **Defect Reports**: Filter defects by their priority and status. Provides insights into defect discovery by type and resolution across various sprints.
- **Test Summary Report**: A final report which describes the results of all test types during the project. It defines the health of the final build of the sprint, in terms of requirements met, outstanding defects, and test cases status. The frequency of this report will be at the end of each major sprint and before each release.

5.0 Overall Testing approach

The overall testing approach for the Echo social media application is a blend of manual and automated testing, designed to ensure comprehensive coverage and efficient defect detection.

We leverage a **comprehensive manual testing document** that provides detailed views for test scenarios, individual test cases, the Requirements Traceability Matrix (RTM), and mappings to defect IDs. This document is the backbone for our manual testing efforts, ensuring that all functional and non-functional requirements are systematically covered.

In parallel, we are implementing **automated testing using Selenium Java**. This automation is specifically applied to the Home Page, Login, and Registration modules, utilizing the Page Object Model (POM) for structured and maintainable test scripts. The automation currently focuses on desktop browsers only, providing rapid feedback on critical functionalities within these modules.

When a defect is identified, a clear **defect management process** is followed:

1. **Defect Creation**: A defect is created and logged with all necessary details (steps to reproduce, actual vs. expected results, screenshots, logs).
2. **Developer Alert**: The relevant developer is immediately alerted about the new defect.
3. **Defect Discussion & Validation**: The developer and tester meet to discuss the defect. During this discussion, the test case that exposed the defect is validated to ensure its correctness and relevance.
4. **Solution & State Change**: If the defect is validated, the developer is tasked with creating a solution (fix). The defect's status is then updated to either "Fixed" (if a solution is implemented) or "Pending" (if the fix is in progress or awaiting retesting).

5.1 Test Design Approach

Test design is driven by FRD and user stories, focusing on clear acceptance criteria. We encourage ATDD principles for defining executable tests early. Manual test cases are detailed in a comprehensive document, ensuring full coverage of business flows. Automated test specifications are reviewed for accuracy and adherence to best practices. Before test case creation in a sprint, user stories must be finalized with clear acceptance criteria, and relevant documentation (FRD, designs, technical specs) must be available.

5.2 Test Execution Approach

Test execution combines manual and automated efforts. Manual tests are performed systematically using the comprehensive manual testing document, covering exploratory and usability scenarios, with iterative cycles per sprint and during regression. Automated tests, using Selenium Java and POM, run frequently on desktop browsers for Home, Login, and Registration modules, integrated into the CI/CD pipeline for rapid feedback. Interface systems are validated via manual testing, API testing, and leveraging test data/virtualization. All defects found are logged and discussed with developers for prompt resolution.

6.0 Test Environment

Test Environments	Application	URL’s	Owners	Comments
chrome	ECHO social media	https://echo1234.com		
edge	ECHO social media	https://echo1234.com		

7.0 Tools & Hardware / Software Requirements

This section outlines the essential tools, hardware, and software for the Echo social media application testing project.

- **Tools:** Selenium WebDriver, Apache POI, Java Development Kit (JDK), Test Management Tool, Defect Tracking System, Reporting Framework (e.g., TestNG Reports, Extent Reports), Integrated Development Environment (IDE), Version Control System Client.
- **Hardware:** Standard desktop/laptop workstations with sufficient RAM and processing power.
- **Software:** Windows and macOS operating systems, Google Chrome, Mozilla Firefox, Microsoft Edge, Safari web browsers, and Excel.

8.0 Dependencies

This section outlines the key dependencies required for the successful execution of the Echo social media application's testing process.

8.1 Documentation:

- Functional Requirement Document (FRD)
- Comprehensive Manual Testing Document (for test scenarios, test cases, RTM, defect mappings)
- Test Type Guide

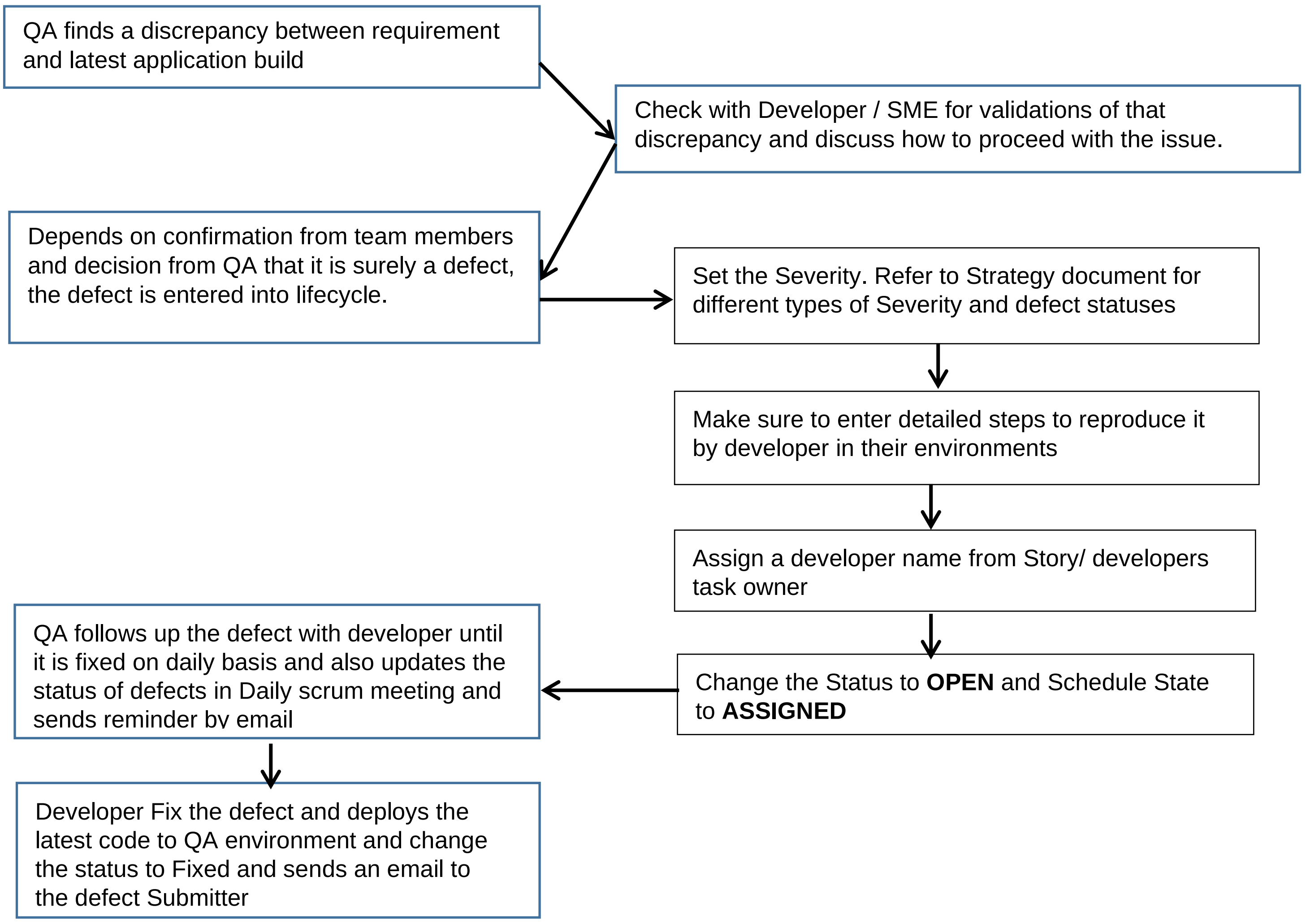
8.2 Tools & Technologies:

- Selenium WebDriver (for test automation)
- Java Development Kit (JDK) (for running Java-based automation)
- Apache POI (for runtime data injection from Excel files)
- Excel (for test data storage)
- Reporting Framework (e.g., TestNG Reports, Extent Reports)
- Defect Tracking System (for logging and managing defects)
- Test Management Tool (for managing test cases, test suites, and execution)

9.0 Defect Management

<<List down the defect process steps based on the project need. Sample provided below which needs to be modified based on the project need>>

Defect Process steps:



QA finds a discrepancy between requirement and latest application build

Check with Developer / SME for validations of that discrepancy and discuss how to proceed with the issue.

Depends on confirmation from team members and decision from QA that it is surely a defect, the defect is entered into lifecycle.

Set the Severity. Refer to Strategy document for different types of Severity and defect statuses

Make sure to enter detailed steps to reproduce it by developer in their environments

Assign a developer name from Story/ developers task owner

Change the Status to **OPEN** and Schedule State to **ASSIGNED**

QA follows up the defect with developer until it is fixed on daily basis and also updates the status of defects in Daily scrum meeting and sends reminder by email

Developer Fix the defect and deploys the latest code to QA environment and change the status to Fixed and sends an email to the defect Submitter

9.1 Defect Severity & Priority Definition or Classification Guidelines

Defect Severity	Definition / Classification Guideline
Critical	Fatal error or “Show Stopper” No Work Around - Cannot continue testing
High	Prevents the system from meeting business requirements Major Functionality Failure Workaround is complex and and/or time consuming, Testing can continue in another area and slow down the execution progress
Medium	Affects functionality (not critical) Simple Workaround is available Testing can proceed
Low	Cosmetic issue Does not affect any functionality Testing can proceed without interruption

Defect Priority	Definition / Classification Guideline
Critical	Blocks core functionality; no workaround; data corruption/loss.
High	Major functionality severely impacted; significant data issues; impacts many users.
Medium	Minor functionality affected; limited user impact; cosmetic issues with workaround.
Low	Minor cosmetic/trivial errors; no functional impact; With easy work around

10.0 Summary

This document outlines the comprehensive test plan for the Echo social media application, employing a hybrid approach of **manual and automated testing** (using Selenium with Java and Page Object Model for desktop browsers).

It details the **testing approach and strategy**, including agile team participation in defect triage and story grooming, handling scope creep, and defining Epic/Story prioritization. It also covers the **feature testing approach**, emphasizing a comprehensive manual testing document for scenarios, test cases, and RTM.

The plan specifies the **test automation approach**, focusing on Home, Login, and Registration modules, and utilizing Apache POI for test data injection from Excel. It defines the **test data management** strategy, covering provisioning, maintenance, and automation.

Furthermore, the document lists the **testing phases, scope, deliverables, acceptance criteria**, and impacted applications. It outlines the **overall testing approach**, including test design and execution strategies. Finally, it details the **tools, hardware, and software requirements** for the project and lists all **dependencies** and the **defect priority classification** guidelines.