Test Plan
For
Admin360

Prepared By:	**Amaan Saiyed**	**Reviewed by:**	
Version No:	**V1.1**	**Issue Date:**	**19/11/2024**

1.1 Document Revision History*

Document Version	Submitted By	Date	Reviewed & Approved by	Notes
V1.0	Amaan Saiyed	20/11/2024		
V1.1	Amaan Saiyed	26/11/2024		

Table of Content

Test Plan – Admin360	

1. Introduction

1.1 Purpose

The purpose of this test plan is to make sure the **Admin360 Application** works smoothly and securely, especially in key areas like **Login**, **Dashboard**, and its features for managing **guest houses** and **tracking remote office attendance**. This document serves as a guide for stakeholders to align testing efforts with project goals.

1.2 About this document

This document serves as high level test planning document with details on the scope of the project, test strategy, test schedule and resource requirements, test deliverables and schedule. All testing activities and provides detailed information on how the testing process will be executed for the Admin360 Application.

1.3 Related Documentation

- SRS Document – System Requirement Specification.
- Screenshots - Visual aids for application features

2. Scope

2.1 Application Scope

The scope of the project includes testing the following features of Admin360.

Inclusion

- **Login**
- **Dashboard**
    - **Guest House System**
    - **Remote office attendance System**

From our understanding, we believe above functional areas need to be tested.

2.2 Test Scope

Testing will cover:

- Functional Testing: Login, Dashboard, Guest House, and Remote Office Attendance.
- Integration Testing: Interaction between modules.
- Usability and GUI Testing: Verifying interface usability.
- Security, Performance, and Compatibility Testing.

2.3 Exclusions to the Scope

- All the features except that are mentioned under ‘Inclusions.
- Any third-party features or Payment gateways.
- Test Automation

3 Approach

3.1 Traditional Testing

We need to perform traditional testing of all the functionalities mentioned in the above Scope section. It will include

- Test case design
- Test Execution
- Defect tracking.

3.2 Functional Testing

We need to perform Functional Testing of all the functionalities mentioned in the above Scope section. As part of Functional Testing, we will follow the below approach for Testing:

We will apply several Test Designing techniques while creating Test Cases

- Equivalence Class Partition
- Boundary Value Analysis
- Decision Table Testing

We also use our expertise in creating Test Cases by applying the below:

- Positive Testing – Validate by using valid input.
- Negative Testing – Validate by using invalid input.
- Exploratory Testing - To ensure new changes do not impact existing functionalities.
- Retesting - For failed test cases

3.3 Integration Testing

In integration testing how components of the application interact with each other.

- End-to-End Scenarios- Test end-to-end dataflow of multiple modules after integration.
- Data Consistency- Verify that data flows correctly between integrated components.

3.4 Look and Feel Testing of Web pages (GUI aspects)

A graphical user interface includes all the elements such as text box, menus, checkbox, buttons, colors, fonts, size, icons, content, and images.

Our main focuses will be how attractive and user-friendly website is then only user will use this website and application.

3.5 Browser Compatibility Testing

We will do Browser Compatibility Testing to make ensure website functions or web Applications working properly across various browsers, versions, and configurations.

Browser Compatibility Testing Scope:

1. **Browser List**

- Google Chrome
- Edge
- Internet Explorer

2. **Operating System**

- Window (10, 11)

3. **Test Environment**

- Hardware configuration

- Software configuration
- Network configuration

3.6 Application Testing on Mobile Device

We will test the mobile version of Admin360 to ensure functionality and compatibility on smartphones and tablets.

1. **Devices: Include devices to be tested:**

- Mobile Device (Android, OS)
- Tablets

3.7 Security Testing

Security Testing ensures that the application is protected against system's vulnerability to threats, focusing on Login security, data protection, and user role management.
It will include:

1. **Penetration Testing** - Simulating attacks to exploit vulnerabilities and evaluate the system's defense mechanisms.
   Tools: **Burp Suite**, **Wireshark**, and **Metasploit** for automated vulnerability scanning and manual attack simulations.
   Environment: Conducted in a dedicated **staging environment** that mirrors production, with appropriate permissions for detailed logging and monitoring.
2. **Data Encryption** - Any sensitive data being sent or received must be encrypted.
   Tools: **Wireshark** for analyzing network packets and **OpenSSL** for verifying encryption algorithms and SSL/TLS configurations.
   Environment: Testing will be performed on environments configured with HTTPS.
3. **Multiuser support** - without interfering with the data between them.
   Tools: **JMeter** for simulating concurrent user activity and database auditing tools to ensure data isolation.
   Environment: Scenarios tested in the **staging environment**, with seeded test accounts mimicking real-world user roles and permissions.
4. **Access control -** Validate that users can only access data and features based on their roles and permissions.
   Tools: **Postman** for testing role-based API access, and manual testing for UI validation.
   Environment: A staging environment pre-configured with predefined user roles (e.g., Admin, Editor, and Viewer).

3.8 Performance Testing

We will do Performance Testing to ensure that **Admin360** will perform well under their expected workload.
Our goal for Performance Testing is not to find bugs but to eliminate performance bottlenecks.
We will test the **Admin360** performance like its response time, reliability, resource usage and scalability under the various conditions. This Includes:

1. **Load Testing** - To test **Admin360's** speed and efficiency when used by many people simultaneously.

Tools: Apache JMeter, Load Runner.
Environment: Conducted in a scaled staging environment mirroring production resources.

2. **Stress Testing** - To evaluate **Admin360's** stability under extreme or peak usage conditions.
   Tools: Locust, K6.
   Environment: Use stress-testing tools in the production-mimicking staging environment with increased load beyond capacity.
3. **Scalability Testing** - To assess **Admin360's** ability to handle increased workload, users, or data without compromising performance.
   Tools: Locust, K6.
   Environment: Use stress-testing tools in the production-mimicking staging environment with increased load beyond capacity.

3.9 User Acceptance Testing (UAT)

The objective of UAT is to validate the application against real-world scenarios, ensuring it meets the requirements and expectations of end-users.

- Verifying that Admin360 satisfies business requirements.
- Testing real-world scenarios to ensure the application behaves as expected.
- Gathering feedback from users to identify any issues or improvements.

Once UAT is complete, stakeholders sign off to indicate the application is ready for production.

4. Test Environment

4.1 QA/Staging Environment

The QA and staging environments is designed to support the testing activities, to ensure accurate validation of Admin360's functionality and performance**.** The QA Environment is a setup for the QA team to perform functional, integration, and regression testing.

Key aspects include:

- Tools and Resources - Equipped with necessary testing tools, monitoring tools, and resources for effective testing.
- Data Setup: Contains test data specifically created for functional, performance, and security testing.
- Stability: Ensures a stable environment to run repeated and consistent tests without interference.

Tools and Resources:

- **Testing Tools:** Includes tools for functional, integration, and regression testing, such as automation frameworks and defect management tools.
- **Monitoring Tools:** Supports performance and stability monitoring during tests, helping to identify issues under simulated workloads

Purpose of QA Environment:

- Validate individual features and workflows.
- Perform detailed regression testing after every sprint.
- Identify and log bugs for resolution by the development team**.**

4.2 UAT Environment

UAT environment is a final stage before the system goes live.

Key Aspects of UAT Environment:

- Provides access to end-users or stakeholders for conducting acceptance tests.
- The production environment means configuration, data setup, and integrations.

Tools and Resources:

- **Acceptance Testing Tools:** Tools for recording and tracking test cases, feedback, and results.
- **Collaboration Platforms:** Enable stakeholders to communicate findings efficiently with the project team.

Purpose of UAT Environment:

- Verify the system meets the defined business and functional requirements.
- Identify any usability issues in business workflows.
- Confirm system ready for deployment to production.

5. Defect Tracking Mechanism

All identified defects will be logged, tracked, and managed using a defect tracking tool (e.g., JIRA). Defects will be categorized by severity and priority, with regular updates shared with stakeholders. This defect tracking mechanism applies to all testing activities, including functional, integration, regression, performance, and user acceptance testing.

Assigned Roles:

- QA Engineers will log the defects.
- Development Team will fix the defects.
- QA Lead will verify and close defects after validation

Defect Reporting will include:

- Summary: A brief description of the defect.
- Steps to Reproduce: Detailed instructions to recreate the defect.
- Expected and Actual result: Description of what was expected and what occurred.
- Priority and Severity: Based on business impact.
- Environment Details: Including the browser, OS, and application version.
- Attachments: Screenshots, logs, or other supporting files.

6. Test Deliverables

These are the important documents and items we will create and share during the testing process for Admin 360:

1. Test Plan Document
2. Test Cases
3. Test Data
4. Defect Reports
5. Test Execution Reports
6. Traceability Matrix
7. Test Summary Report
8. User Acceptance Testing (UAT) Report
9. Release Notes
10. Test Closure Report

7. Assumption

1. **Test Environments Will Be Ready and Available**
   - The QA, staging, and UAT environments will be set up and fully functional before testing begins.
   - All necessary configurations, integrations, and data setups will be completed and verified

2. **Availability of Required Testing Tools**
   - Required tools for testing, such as automation frameworks and defect management systems, will be available and operational.
   - Suitable hardware and network resources will be allocated to support performance and security testing.
   -

3. **Stakeholder Participation**
   - Key stakeholders, including business users and developers, will be available for UAT and defect resolution.
   - Feedback cycles will be scheduled and adhered to during UAT phases.

4. **Risk Tolerance**
   - Any risks or blockers identified during testing will be documented and addressed promptly to minimize impact on testing schedules.

5. **Bug Resolution Process**
   - Defects identified during testing will be logged, prioritized, and resolved within agreed timelines.