



Factual Note Test Plan Document

Table of Contents

1. Introduction	3
1.1. Test Plan Objectives	4
1.2. Purpose	4
1.3. Reference Documents	4
2. Scope.....	4
2.1. Factual Note - In/Out of Scope	4
2.2. Functional QA - In Scope	4
2.3. Non-Functional QA - In Scope	5
2.4. Non-Functional QA - Out of Scope	5
3. Test Strategy & Approach	5
3.1 Test Case Creation	5
3.2 Testing Methodology	5
4. Control Procedures	6
4.1. Reviews	6
4.2. Bug Review meetings	6
4.3. Change Request.....	6
4.4. Defect Reporting	6
5. Functions to be tested	7
5.1. Functions not to be tested.....	7
6. Resources and Responsibilities	7
7. Suspension / Exit Criteria	8
8. Resumption Criteria	8
10. Dependencies.....	8
10.1 Personnel Dependencies	8
10.2 Software Dependencies.....	8
10.3 Test Data & Database	9
11. Risks.....	9
11.1 Schedule.....	9
11.2 Management	9
11.3 Requirements	9
11.4 Training	9
12. Test Deliverables	9

1. Introduction

Factual Note:

FN is a web annotation engine, which helps the end-user to annotate (mark, comment) the specific text, element, page, video, etc. in a web page and share it to like-minded people. Recipients understand the sharing intentions efficiently as they are pointed to the Website data rather than a whole website, they respond to the shared data by starting a discussion, rating the information, flagging them (as spam, prohibited and miscategorized), etc.

FN is an “Advanced information retrieval engine” and also termed as “User perspective based search engine”. Between the birth of the world and 2016, there were Zettabyte (1 000 000 000 000 000 000 Bytes) of information created. It’s so painful to find the accurate information on a specific subject, Existing search engines are pointing us to many different websites on a similar subject. The only way to derive the accurate information from these huge data, is by involving the like-minded people feedbacks, ideas, discussions and their perspective on the entire website’s data on a specific subject, which **FN** does.

FN is an effective interface for knowledge sharing. All the information that is read by the users on the web evaporates after sometime. The time invested in reading is also wasted. The same amount of time can be saved for each individual who is in need of exact information.

FN establishes a *linear chain* of links using its algorithm that will lead the end user to a pool of knowledge on the subject. More users working on the same article will '*collaboratively*' tag the content.

FN analyzes the *semantics* of the *collaborative tagging* and the respective content behind to avoid information overload. By doing the *semantic analysis*, we also enhance the user experience and user participation.

FN allows end users to link multiple user-marked web pages or documents (related to 1 particular content). Any user on the web searching for a similar content is routed by **FN** to this **FN** 'tags'. **FN** is unique in a way pointing the user to the exact location of 'desired' data (**FNtags**) in a webpage using advanced *Machine Learning Algorithms* (supervised and non-supervised alike).

FN enables the user to store the information in the same way it was read thus making the information retrieval much easier for the user involved and for other parallel users at the same time. This is best achieved by **FN** by establishing a *linear relationship* between the user-marked content.

1.1. Test Plan Objectives

This document specifies the scope of functional testing, the overall approach to be adopted, the activities to be completed, the resources required and the methods and processes to be used to validate the functionality of 'Factual Note' web application that meets the established business requirements prior to release.

The objectives of the Test plan are as follows:

- To detail the project requirements in terms of the testing scope
- To identify the test strategy and test methods
- To identify the required resources and their responsibilities
- To list the deliverable elements and acceptance criteria of the test process.

The Test Plan provides a road map to the verification and validation process by providing a breakdown of the activities to be performed during testing.

1.2. Purpose

This document describes the approach and methodologies used by ZeptoH QA to plan, organize and manage the testing of Factual Note web application

2. Scope

This application is coded with plain JSP and Servlets. The ZeptoH QA team is responsible for testing the functionality product and ensuring it meets client's needs.

2.1. Factual Note – In/Out of Scope

Factual Note	In Scope Iteration#1-Sprint#1 (I1S1)
Functionality Testing	Yes
Data Base Testing	No
Compatibility Testing	Yes Sprint #1 will be tested in Window 10 & FF, Chrome, IE11 and Safari . we will test it with latest version of the browser for compatibility

2.2. Functional QA – In Scope

The areas which are in scope for functional QA during the duration of the Factual Note project are:

- Complete functional testing of the web application.
- New features (if any)
- Bug fixes (if any)

2.3. Non-Functional QA – In Scope

- User Interface / Usability Testing: This is not specifically in scope; however, QA will highlight any areas that are felt badly designed or inhibit the user experience. Every effort will be made by QA to make recommendations based on their experience, how to improve the appearance, layout and design of certain areas of the system.
- Automation Testing
- Mobile Testing.

2.4. Non-Functional QA – Out of Scope

- Localization/Internalization
- Stress
- Security
- Scalability
- Accessibility

3. Test Strategy & Approach

Different test activities will be performed by the QA team. QA will be responsible for Functional testing, Integration Testing, System testing, Adhoc Testing and Regression testing of each Factual Note iteration/sprint. Once the build has reached QA exit criteria it will be certified for production release.

Development → QA → Release → Production Release

3.1 Test Case Creation

The QA test design process for each new feature is as follows

1. Requirements and Wireframe Analysis
 - Requirement study
 - Raise queries for clarification
 - Discussion with Factual Note team
2. Test Case Preparation:
 - Test Case Preparation
 - Lead/Manager Review
 - Rework on TCs if applicable
 - Mailing the query to PM review
 - Rework on TCs if applicable

3.2 Testing Methodology

Smoke testing will be performed on each build delivered to QA. Smoke testing verifies the functionality at high level. The objective is to determine if further testing is possible. This testing emphasizes breadth more than depth. All components should be touched and every major feature should be tested briefly by the Smoke test. If Smoke test fails then QA rejects the build and further testing will not be carried out in that build.

Testing of application is in two phases: A) in the Test environment and B) limited testing in the Production environment.

A) Testing in the Test environment:

The following strategy is followed while testing the application in the test environment:
 Functionality testing: The functionality of the new features will be tested using the test cases.

Regression testing: Bugs will be verified after they are fixed in subsequent releases, in addition QA Team will also ensure that the existing functionality works as before.

B) Testing in the Production environment: Limited testing in the Production environment.

The table below outlines the Level of testing done by QA on each test environment on each iteration/sprint, along with the entry and exit criteria and mode of execution:

Test Level	Environment	Entry Criteria	Exit Criteria	Mode Of Execution
1	Testing	Final Build from the development	Testing is complete with all the bugs being raised in the Defect tracking tool. All the bugs raised are fixed and retested and closed.	Manual Functional Testing
2	Production	Tested build from Testing Environment.	No bugs exist except open issues	Manual Functional Testing

4. Control Procedures

4.1. Reviews

The QA team will perform reviews for each Phase. (i.e. Requirements Review, Test Plan Review, Test Case Review and Final Test Summary Review). A meeting notice, with related documents, will be emailed to each participant.

4.2. Bug Review meetings

Regular weekly meeting will be held to discuss reported defects. The development department will provide status/updates on all defects reported and the test department will provide addition defect information if needed. All member of the project team will participate.

4.3. Change Request

Once testing begins, changes to the Factual Note application are discouraged. If functional changes are required, these proposed changes will be discussed with the team. The team will determine the impact of the change and if/when it should be implemented.

4.4. Defect Reporting

When defects are found, the testers will complete a defect report on the Defect tracking system 'JIRA'. The defect tracking system is accessible by Testers, Developers & all members of the project team.

When a defect has been fixed or more information is needed, the developer will change the status of the defect to indicate the current state. Once a defect is verified as FIXED by the testers, the testers will close the defect.

Defect Categories:

Defects found during the Testing will be categorized as below:

List of Defect Categories

Severity	Impact
1 (Critical)	<ul style="list-style-type: none">This bug is critical enough to crash the system, cause file corruption, or cause potential data lossIt causes the application to hang and requires re-booting the system.
2.(Blocker)	<ul style="list-style-type: none">A major module is completely blocked
3 (Major)	<ul style="list-style-type: none">A major feature is broken
4 (Minor)	<ul style="list-style-type: none">Minor lose of function and there's an easy workaround function
5(Trivial)	<ul style="list-style-type: none">Cosmetic Issues
6 (Enhancement)	<ul style="list-style-type: none">Request for new feature or enhancement.

5. Functions to be tested

The following are the list of requirements/screens that will be tested for Phase 1:

- FN_R1: Login Functionalities.
- FN_R2: Generate Notes Function.
- FN_R3: Generated a Note
- FN_R4: Reviewing the generated note
- FN_R5: Notes
- FN_R6: Analytics Page
- FN_R7: Search
- FN_R8: Feedback and Report a bug

5.1. Functions not to be tested

- (TBD)

6. Resources and Responsibilities

The Test Lead and Project Manager will determine when system test will start and end. The Test lead will also be responsible for coordinating schedules, equipment, & tools for the testers as well as

writing/updating the Test Plan, Weekly Test Status reports and Final Test Summary report. The testers will be responsible for Requirement analysis, writing the test cases, executing the tests and logging defects.

7. Suspension / Exit Criteria

The guideline criteria detailed below will be used as a reference to determine the level of quality expected from the software. If these criteria are not met, the software will be deemed un-testable, testing will be suspended and the software returned to the development team for improvement. Testing will resume once the quality reaches an acceptable level.

The QA team may suspend partial or full-testing activities on a given build if any of the following occurs:

- Application Fails the Smoke test.
- Show stopping defects, which prevent further testing along any path.
- There is a fault with a feature that prevents its testing and there is no suitable work around.
- Too many blocking / high severity bugs found too quickly
- Number of defects has reached a point where the subsequent testing has no value adds to the completion of the iteration.
- New build does not contain the specified change(s).
- A severe problem has occurred that does not allow testing to continue.
- Development has not corrected the problem(s) that previously suspended testing.
- Non-availability of the application
- Non availability of Test Environment(downtime)

8. Resumption Criteria

When a new complete build is deployed, the QA Lead needs to ensure that the team is able to proceed testing with showstoppers rectified.

This procedure includes and is not limited to:

- Availability of Application
- Stable environment up & running for testing
- Show stopper is rectified
- Deployment of fixes/Patches for reported defects
- Sanity test passed application.

10. Dependencies

10.1 Personnel Dependencies

The test team requires experience testers to develop, perform and validate tests.

10.2 Software Dependencies

The source code must be unit tested and provided within the scheduled time outlined in the Project Schedule.

10.3 Test Data & Database

Test data & database should also be made available to the testers for use during testing.

11. Risks

11.1 Schedule

A slip in the schedule in one sprint could result in a subsequent slip in the test schedule. If the schedule for each Sprint is aggressive it could affect testing.

11.2 Management

Management support is required so when the project falls behind, the test schedule does not get squeezed to make up for the delay. Management can reduce the risk of delays by supporting the test team throughout the testing phase and assigning people to this project with the required skills set.

11.3 Requirements

The Test plan, Testcases and test schedule are based on the Wireframes, User Stories and Query logs. Any changes to the requirement could affect the test schedule and the test cases need to be modified or redesigned based on the new requirements.

11.4 Training

Training from the Product Manager/Business Analyst on understanding the business rules and new requirements is required for QA Team to develop the test cases and testing the features.

12. Test Deliverables

The Deliverables from QA team will include the following:

No.	Title/ Section	Description
1	Test Plan	A document detailing the QA approach, including details of objective, scope & out of scope, environments and deliverables (this document)
2	Test Cases	These Testcases will be stored in Excel sheet, will contain Test objective, detailed steps along with expected results to test the functionality. This will map to the test coverage matrix.
3	RTM (Requirement Traceability Matrix)	The RTM will trace Test cases back to the requirements to ensure all requirements are sufficiently covered and tested.
4	WSR/DSR	Weekly/Daily status reports
5	Defects	All defects will be logged in JIRA
6	Defect List & Matrix	List of defects and matrix – this will be made available at any point during testing phase.
7	Test Summary Report	Test completion report details the test results for that particular cycle of testing x environment x build x features.