

Team 16

NGO Information Management Suite 1.0

Test Plan

Overview:

The test plan is to refer to once the coding on small individual modules of the project is done, so that it is easy to integrate the system with least bugs later on. This is to assist in coordinating software and test versions within configuration management.

Intended Audience:

Mentors

Developers / Coders

Testing Team

Revision History:

Version	Primary Author(s)	Description of Version	Reviewed By	Date Completed
1.0	Megha, Palashi, Jatan	1 st Test Plan Report with test features	Aakash	23 Feb 2012

Table of Contents:

1. References	4
2. Introduction	
2.1 Overview of NIMS	4
2.2 Purpose of Document	5
2.3 Formal Reviewing	5
3. Test Items	5
4. Software Risk Issues	6
5. Test Case Format	6
6. Features to be Tested	7
7. Features not to be Tested	13
8. Approach	13
9. Item Pass/Fail Criteria	14
10. Test Deliverables	14
11. Hardware and Software Needs	14
12. Responsibilities	15
13. Schedule	15
14. Planning Risks and Contingencies	16
15. Glossary	17

1. References

The following documents are needed to proceed with the development of an efficient test plan for the product:

- Project Plan - To complete the product development in timely and cost effective manner
- SRS - To clearly comprehend what features are needed by the client/user of the product
- Design Document - High level + Detailed
- Test process standards
- IEEE 829 template

2. Introduction

2.1 Overview of the NIMS:

We aim to develop a web based information management system that takes care of the basic needs of cataloguing their work, evaluating the ngo's and its employee(and community) volunteer's performance and aid in creating social maps of the population and the area in which the volunteer is working. This IMS is specifically being designed for an NGO that works for the identity crisis of the Notified and Denotified Tribes of India. They have several projects like low level education of the tribes, acquiring domestic land, voting rights, health of the settlements, acquisition of several national identification cards, projects to enhance tribal art, etc. The IMS assists the organisation to keep a permanent and a well managed record of their data which is right of being taken down on paper, it will also be equipped with several relevant statistical analysis capabilities. The IMS is going to consist of 2 parts (both software and hardware wise) :

1. Client application (which will be with the volunteers/coordinators) for easy data collection and updation.

2. Server application- Database + CMS (at main centre of the NGO) for database management and analysis of the collected data.

- Social Maps (accessible by both server and client app) will be a source of visual information and a dynamic social mapping technique which will help the organisation keep a track of their target locations and their statistics related to ongoing or completed projects.

Note: The term volunteer and coordinator mean the same thing for all documents for our project.

The above functionalities will be delivered in a phased manner. Phase 1 will cover the

implementation of the following features:

2.2 Purpose of the document:

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Testing also provides an objective, independent

view of the software to allow them to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Objectives for writing a system test plan are as follows:

- Satisfy the allocated system requirements
- Solve the right problem (e.g., correctly model physical laws, implement business rules, and use the proper system assumptions)
- Satisfy its intended use and user needs.
- Identify which features of the system will be tested.
- Define the pass/fail criteria for each item to be tested.
- Define any suspension criteria and subsequent resumption techniques Identify the test deliverables.

2.3. Formal Reviewing:

There will be several formal review points before and during system test. This is a vital element in achieving a quality product.

1.3.1. Formal Review Points

1. Design Documentation
2. Testing Approach
3. Unit Test Plans
4. Unit Test Conditions & Results
5. System Test Conditions
6. System Test Progress
7. Post System Test Review

3. Test Items

We intend to test the following:

- Android application - We need to check that the android application “takes in and stores” data from volunteers, validating volunteer login details, not taking in any

wrong format of data (e.g: should not take characters in contact numbers/ number of members in the family), and that it provides the ability for several volunteers, around 10 to login and do data entry simultaneously. The GPS locator of Android should correctly store location and time details. The application should send the stored data without corruption over the GPRS link.

- Server Side Database and CMS- We will check the data updation happens within in a stipulated time after android application collects data. It needs to check if there are manipulated data for same families/children and inform the same, without letting the wrong data entry happen. The CMS with access to Admin only, will allow to make changes and show details - in tabular form, in graphical form, in form of social maps for various works.

4. Software risk issues

The various risks associated with the testing of this project are

- System hang at server side
- System hang at client side
- bug in the android application
- Unwanted results on integration modules
- No receiving of the data packets sent by client side
- Data out of memory exception due to overflow in the data base
- Power cut at the server side
- Bugs in the admin website
- Log in error
- No updating of the database done from the client side at the server side database
- Bugs detected in the final system deployment within NGO environment as the theoretical picture of the environmental needs do not match exactly to those in reality.
- Security issues like Database Hacking on the server computer.
- Problematic internet connection between android and server application and hence data updating is slow and not assured.

5. Test Case Format

We intend to present the test cases in the format specified below assuming that each module has several test conditions to be taken into consideration:

Test ID - Feature Name

Purpose

Pre-requisite
Test data
Steps
Expected Output

6. Features to be tested - Megha, Palashi

1.1 LOGIN

Purpose: To be able to login successfully.

Pre-requisite : No pre-requisite

Test Data : {valid user name, valid password} , {valid user name, invalid password} ,
{invalid user name, valid password} , {invalid user name, invalid password}

Steps:

- In the login details section, enter user name
- Enter password
- Click on login button
- If user name or password is wrong, you will get a prompt "Please Enter again".
- Login again

Expected Output:

If the user name and password are valid, the user should be able to login, or else there should be a prompt to inform about the wrong entry.

1.2 ADMIN INFO ENTRY

Purpose: For the main administrator to enter his/her details

Pre-requisite : The admin should have become a registered user.

Test Data : {Admin ID} , {NGO ID} , {Name} , {Contact number} , {NGO Joining Date}

Steps:

- Login with the valid user name and password.
- Click the personal information button.
- A form appears, with all fields.

- Fill the fields correctly
- Choose the field to be public or private (public - other users can read the information)
- Save the form.
- Logout

Expected Output:

The administrator details will be saved for reference of other users, with the option of making the personal information public/private.

1.3 VOLUNTEER INFO ENTRY

Purpose: For the volunteers to enter their details

Pre-requisite: Volunteer should be a registered user.

Test Data: {Volunteer ID} , {Android Phone number} , {NGO ID} , {Name} , {NGO Joining Date}

Steps:

- Login with the valid user name and password.
- Click the personal information button.
- A form appears, with all fields.
- Fill the fields correctly
- Choose the field to be public or private (public - other users can read the information), but this can always be read by the administrator.
- Save the form.
- Logout

Expected Output:

The volunteer details will be saved for reference of other users.

1.4 CAMPAIGN ENTRY

Purpose: To enter details about various campaigns in village/taluka/district

Pre -requisite: The volunteer should have logged in.

Test Data: {Existing campaign buttons} , {Village} , {Submit button}

Steps:

- Choose either "Start new campaign button" or "Existing campaign button".
- If new campaign, enter village name and further family info as told in 1.5.
- If existing campaign, choose the campaign from drop down list of all available.
- Then give family ID whose information to be seen in the campaign

- Click on submit button

Expected Output:

All campaigns to be added in the database at server and properly updated.

1.5 VILLAGE FAMILY INFO ENTRY

Purpose: To collect and store information about tribal families as needed by the NGO.

Pre-requisite: The user(here, volunteer)should have a login user name and password, to be able to make any data entry.

Test Data: {Family ID} , {Member detail Fields} , {Card Details Field}

Steps:

- Login as volunteer with user name and password
- Click on login
- Clcik on the family info. entry button.
- Enter family ID in the tab asking for the same.
- Fill the entries in the form that opens, with all family details fields - number of members, village ID, family community , family ration card status, etc.
- Then a next form pops out about member information as per the number of members in a family. Fill their details like, name, age, gender, occupation, etc.
- Click on save button.

Expected Output:

The volunteer should be able to make the entries if he logs in correctly and if the family ID is unique, else it will pop up a box saying "This family exists already".

1.6 CHILDREN INFO ENTRY

Purpose: To collect and store information about tribal children as needed by the NGO for the education campaign

Pre-requisite: The user(here, volunteer)should have a login user name and password, to be able to make any data entry.

Test Data: {Child ID} , {Family ID} , {Age} , {Name}

Steps:

- Login as volunteer with user name and password
- Click on login.
- Click on the children info. entry button.
- Enter child ID in the tab asking for the same.
- Enter family ID of that child.

- Fill the entries in the form that opens, with all details fields like name, age, etc.
- Click on save button.

Expected Output:

The volunteer should be able to make the entries if he logs in correctly and if the child ID is unique, else it will pop up a box saying "This child exists already".

1.7 FAMILY INFO UPDATION

Purpose: To update some information about members or card status of the tribal families

Pre-requisite: The volunteer should be registered, or the admin can overwrite the database once registered.

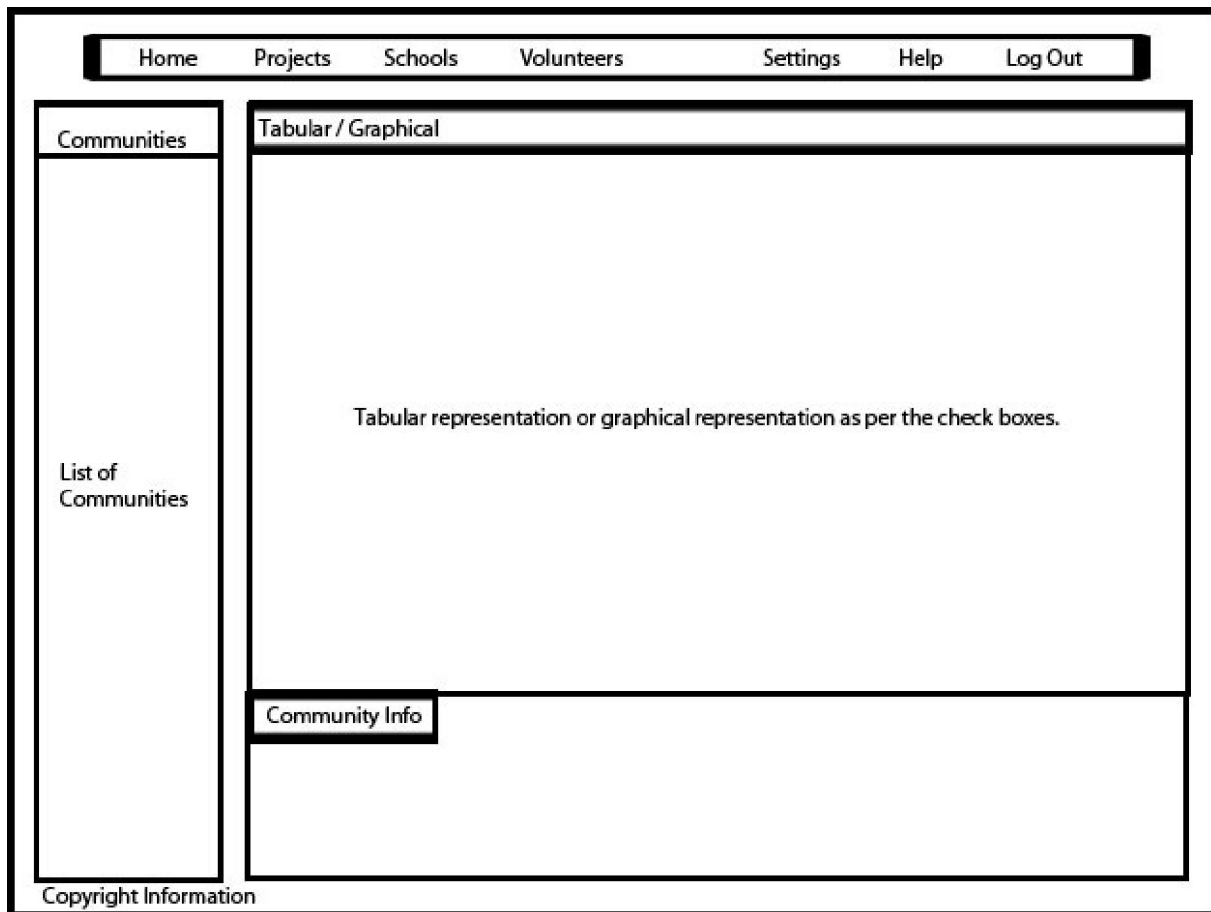
Test Data:{Family Member Details} , {Card Type} , {Card Status}

Steps:

- Enter the user name and password
- Click on login
- Click the family info entry button.
- Enter a family ID that exists.
- A pop up box says "Family exists already", "Do you want to update the information?".
- Click on "YES"
- It opens up the saved information for change.
- Make the appropriate changes.
- If the volunteer tries to make a change to apply for a card out of order, an error is shown.
- Select card applications for a family, one by one in order. (e.g: If the family does not have ration card, it can't apply for free house plan policy card by government.)
- Save the information again, with a timestamp

Expected Output:

The information should be successfully overwritten, and families card status, etc need to be correct and in order.



2.1 REGISTRATION

Purpose : To ensure proper new user registration on the web interface

Pre-requisite : No pre-requisite

Test Data : {unique user name, unique password, authority level} , {unique user name, common password, authority level} , {common user name, unique password, authority level} , {common user name, common password, authority level}

Steps :

- There are 2 authority levels - read/write (admin) , read only (volunteers) .By default the authority level is set to “read only”.
- The main admin creates an account with user name, password and authority level “read/write”.
- The main admin creates accounts for volunteers and donors with the authority level he wishes, in general as in step1.

- Enter the user name.
- Enter the password.
- Click on register button.

Expected Output:

When the user enters the correct user name and password, he/she must be able to login to the website and with the same user name and password on the android phone, for volunteers.

2.2 VILLAGE FAMILY/COMMUNITY INFO VIEW

Purpose: To collect and store information about tribal families as needed by the NGO.

Pre-requisite: The user(here, admin)should have a login user name and password, to be able to view or update family/community information

Test Data: {Family ID} , {Member detail Fields} , {Card Details Field}

Steps:

- Login as admin with user name and password
- Click on login
- Click on the communities button.
- Select community/district from the side pane asking for the same.
- Fill the filters in the form that opens
- As you filter click on tabular view/graphical view depending on which you want to see.
- Logout

Expected Output:

The admin should be able to see the data in tabular or graphical view as and when needed by him/her along with all updations with respect to incoming data from android application device used by volunteers.

2.3 SOCIAL MAP

Purpose: The social maps, should represent correct locations and statistics as per server database.

Pre-requisite: No pre-requisite

Test Data: {Social maps -Graphical display}

Steps:

- The admin should login with valid user name and password.
- The admin should click on graphical view of any tab - like communities, projects, schools, etc.
- If there exists a mapping of it on maps, it will shown, else pie charts, etc will be shown.
- It will be there for communities tab.
- Logout

Expected Output:

The statistics and locations mapped should match with database entries and be updated as and when new data is added on the server database for all tabs.

7. Features not to be tested (In v1)**1.1 SCHOOL INFO ENTRY**

Purpose: Store information about various schools and balghar's of villages

1.2 PROJECT INFO.VIEW

Purpose: To look at timeline and graphics of all ongoing projects simultaneously

1.3 PROJECT INFO.UPDATE

Purpose: To update the timestamps of completion of projects in areas and reflect that on timeline on server CMS.

2.1 HELP TAB ON SERVER SIDE

Purpose: To give help to manage the CMS to admin

8. Test Approach

We will be using the bottom up approach for testing our entire project. We will build the smaller modules and do parallel testing on them functionally at an individual level. We will then integrate them in system and test the system again. The number of increments is three. In each increment the features will be unit tested in the existing system and then their integration testing will be performed with the other features added. This will help in rigorous testing for all the three modules. Regression testing will be conducted if some error/problem occurs in the features.

9. Item pass/fail criterion

An item will pass or fail based on the testing phase result. The pass criterion is:

- The output of the feature is exactly the same as the expected output.
- Execution occurs with no errors.
- Updation and execution happens in an optimal amount of time, that is small for the user experience to go bad.

10. Test deliverables

The following are the test deliverables:

Test Plan Document
Test Cases
Error logs and execution logs
Problem reports and corrective actions

As we proceed with the coding phase of the project, using the bottom up approach, we will do parallel test checks as per the test plan document.

11. Software and hardware needs

Hardware:

The minimum hardware requirements for the testing is:

- Android and GPRS supporting mobile phone
- Internet occupied server side computer

Our product has a client-server application and the client side (volunteer in the field) will have to be connected to the server side (NGO's server) using GPRS (or 3G if available).

The client side of the application will run on an Android based mobile device which will be used by the volunteers to collect and update data at the server side.

On the server side, there will be a server for database management and data analysis (for social mapping) which will have to function properly at all times to ensure the safety and security of the data collected.

Software:

Software	For

Eclipse IDE	Developing android application
Android Emulator plug-in	Android Application Testing
MySQL	Database Management
WAMP	Testing and implementing PHP scripts
Git	Version Control

12. Responsibilities

Name	Role	Responsibility
Aakash	Test Lead	System testing Test cases report Complete test phase report
Lalit	Test planner	System test plan Test case design
Jatan	Tester	Integration / system testing
Megha	Tester	Test case design System testing
Parth	Tester	Test case design

13. Schedule

Following is the list of tests scheduled to be performed on the version 1 of the intended product:

1. Unit Testing: This will be done by all the coders on the specific unit that they develop and hence will be done simultaneously and independently after each unit is coded.

2. Regression Testing: This will be performed for each module (like android app, server interface, etc.) when setup and run as a combination of several units. *The unit testing and regression testing should be exhaustively over by 8th March, 2012.*
3. Integration Testing: This will be performed over the entire system being run as a whole. This should be over and errors encountered mended by 14th March, 2012.

Following are the tests that are scheduled to be performed on the final version of the product that will be delivered as a finished and full proof system to the NGO:

1. Alpha and Beta Testing: This phase is intended to start a day after the integration testing of the final deliverable system is done.
2. Security Testing: Scheduled to start a day after beta testing is done.
3. Performance Testing: Scheduled to start a day after security testing is over to culminate the system testing procedure.

14. Planning Risks

The overall risks to the project are as under,

- Lack of personnel resources when testing is to begin.
- Lack of availability of required hardware, software, data or tools.
- Late delivery of the software, hardware or tools.
- Delays in training on the application and/or tools.
- Changes to the original requirements or designs.

Other possible risks can be,

1. Requirement definition will be complete by 13th February, 2012, but if the requirements change after that, any of the following may be the consequences:
 - The scope of the plan may change.
 - Cost of the project may change and hence cost reviewing and approving will be needed.
 - The number of tests performed might be reduced or number of allowable defects might be increased.
 - Test schedule and development schedule may be shifted accordingly.

2. Team member/s not able to deliver on time due to unforeseen reasons, then following consequences may imply:

- Other members might have to deliver and work more to fill in for the non performing members.
- Team co-ordination can become imbalanced.
- Quality of the product can suffer, so can delivery in time.

15. Glossary

Test No : Signifies the serial number of test case in a particular module.

1.1 - Signifies test case 1 for 1st module(Android Application)

2.1 - Signifies test case 1 for 2nd module (Server Database and CMS)

Pre-requisite : Condition that must satisfy before the test is conducted.

Test Data : Data that is processed during test run.

Expected Output : The result that is ideally expected out of test run.