Team 16

NIMS Suite 1.0

Low Level Design Document

Overview

This is the low level design document for NIMS, an information management suite for an NGO.

The first part gives a brief description of the project topic. The second part is the overview of the design of this application. The third part is the system components which contain the database table and the detailed description of the modules.

Target Audience

This document is intended for the technical team. This document contains detailed information regarding implementation procedures.

Version Control History

Version	Primary Author(s)	Description of Version	Reviewed By	Date Completed
1.0	Anshul, Lalit, Megha, Parth	Low Level Design Document for Phase 1	Anshul	13th Feb, 2012

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the design of the NGO Information Management Suit.
Its main purpose is to –

- Provide low level design of the system.
- Document the algorithm used in building each module or group of modules and shows how the various components are implemented
- This document is intended to help the coding team to build our system.

1.2 Document Overview

This document contains detailed design of the system like detailed description of Modules, Database tables.

1.3 Scope

This document contains the design of the actual program code which is designed based on the High Level Design Document. Logic design done for every program is documented here. Hence the developers can code directly by referring to this document with minimal effort of debugging and testing.

1.4 Document Organization

This document is organized into the following sections:

- Introduction: - Provides information related to this document (e.g. purpose, term definitions etc.)
- Design Overview: - Describes the approach and guiding principles.
- System Components: - Describes in detail functionality and implementation details of various modules and group of modules which interact to provide common functionality.
- Glossary

1.5 References

- Software Requirements Specification
- Project Plan
- High level design document

2. Design Overview

2.1 Approach

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. The IMS will be equipped with several relevant statistical analysis capabilities. The IMS is going to consist of 3 parts (both software and hardware wise):

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

2. **Server application** (at main centre of the NGO) for database management and analysis of the collected data.

3. **Generic public website** (which will be test-implemented for the client NGO) to let public view their activities and their progress via social maps; the generic format of the website gives options to customise the settings and information displayed on the website according to an organisation's specific needs.

For the server end, data coming from Android phones used by coordinators/volunteers will be stored in a database. The architecture of the website and automatic error detection and information retrieval system on Android is also provided.

Technical Design: We have depicted the flow of processes which is described in section 4 "Use case Diagrams", the database by using "E-R Diagram" and "Relational Schema". The final top level architecture is explained in section 5, of top level design. The section 3 of detailed design has various tables for the server end database, all along with primary keys, attributes and foreign keys identified. It has the use case diagrams as well, which depicts in detail the flow of information collected by volunteers using android phones, which finally leads to information updation in server database.

3. System Components

3.1 Database tables

Family_Info Table

Fam_id	Fam_no_of_members	Vil_id	Fam_Community
Fam_Ration_Status	Fam_Ration_category	Fam_House_benefit_scheme	Fam_Traditional_Occup
Water_conn.	Coordinator_name(or no.)	Survey_Date	

Member Table

Fam_id	**Mem_id**	Mem_Name	Mem_birth_year	Mem_Occupation
Mem_Relation_with_Head	Mem_Gender	Mem_Job_Card	Mem_Voter_Status	Mem_Adhar

Village Table

Vil_Id	Vil_Name	Vil_location	Vil_Head	Vol_Id

Children Table

Ch_id	Ch_Name	Fam_id	Ch_Birth_Year

School_Children_Info Table

Fam_id	**Ch_id**	Sch_id	Standard	Ch_joining_date	Ch_Performance

School_Info Table

Sch_Id	Sch_Name	Sch_Address	Sch_Head	School_or_Balghar

Social Map Table

SM_Id	SM_Name_Place	SM_Category	Village_id	SM_latitude	SM_longitude	Vol_id

Distribution Table

Item_Name	Item_Id	Vil_id	Fam_id

Volunteer Table

Vol_Id	**NGO_ID**	Vol_Name	Vol_Phone	Vol_joining_date

Donors

Donor_ID	**NGO_ID**	Donor_Name	Donor_Contact	Donor_Don_Date

Admin Staff

Admin_ID	**NGO_ID**	Admin_Name	Admin_Contact	Admin_Join_Date

NGO

NGO_ID	Name	Office_Location	Type	Contact_No.

Campaign_Info

Camp_ID	**Vill_ID**	Coord_ID	**Start_Date**	End_Date

3.2 Detailed level Description of Modules

3.2.1 Client Side Modules

Login Module: This module does the following things**:**

- Loads Login Form for the coordinator.
- Coordinator enters his/her coordinator_id.
- When the submit button is clicked, validate_coordinator function is called which validate the coordinator by his/her coordinator id and returns to login form in case he/she is not an authenticated coordinator.
- In case the coordinator is valid coordinator, then the campaign module is called.

Campaign Module: This module is used to manage the information of the campaigns.

- Loads the Check New campaign form which allows the coordinator to start a new campaign or restart an already existing campaign through the check_new_campaign function.
- In case, the coordinator wants to start a new campaign, post_new_campaign function is called and a form is generated which allows the coordinator to enter the information of the new campaign.
- The coordinator can also re-start and work on his existing campaign by selecting the campaign from the list of campaigns already available in the database. This is done through the select_my_campaign function.
- This will be followed by calling the display_projects function which would display the available projects for that campaign.
- The coordinator would select a project from the list of the displayed projects and the corresponding module would be called.

Family_info Module: This module is used to maintain and manage the family information.

- Loads the Check New Family form which allows the coordinator to enter information of a new family in the database or edit the information of an already existing family in the database through check_new_family function.
- In case, information of a new family is to be entered, get_full_family_info function would be called and would generate a form to get the information of

the new family including the family head and the other members of the family. On submission of this form, a new family id will be generated.

- The coordinator can also edit the information of an existing family either by entering the family id or any other information like family head, village name etc. The family information entered will be validated against the database and the corresponding matched entries will be displayed from which the coordinator can select the appropriate family whose information is to be edited.
- The identity cards will be displayed through the display_identity_cards function.
- When the coordinator selects any of the cards, validate_previous_cards function would be called to check whether the previous cards have already been issued or not. This will ensure the cards are issued in correct sequence. In case the previous cards have not been issued, the coordinator will not be allowed to edit the data for the selected card.
- After the previous cards validation is completed and the function verifies that the identity card is in correct sequence, the coordinator can edit the information for this card in the database.
- If the current card exists, then the coordinator updates the database with a value 1 implying that the card has been issued to the family with the given family id. If neither the current card exist nor any application has been submitted for issuing the card, the coordinator on the behalf of the family submits an application and corresponding changes are made in the database. In such a situation, the database is updated with a value of 2, implying that the application for the identity card has been submitted.
- If the family has already applied for the identity card, the coordinator manually checks with the government officials regarding the application status of the identity card and updates the database accordingly.

3.2.2 Server Side Modules

Login Module: This module does the following things**:**

- Loads Login Form for the administrator.
- The administrator enters his/her username and password.
- When the submit button is clicked, validate_coordinator function is called which validate the username and password returns to the login form in case he/she is not an authenticated administrator.
- In case he/she is an authenticated administrator, then the campaign module is called.

Family_Info Module: This module is used by the administrator of the NGO to view and analyse the various information regarding the families in a village.

- The get_family_info function takes the information of the village which the admin wants to view the data of as an input.
- The validate_village_info validates the village info and if the information is valid displays the all the information of the required village including the location of the village and other demographic details.
- The admin can either view a graphical analysis of various projects in the village or can view a tabular analysis of the individual families of the village. The check_selection function checks the option selected by the user and calls the appropriate sub module.
- If the admin wishes to do an analysis of the various projects conducted in the village, the sub module will display a graphical analysis of the results of the various projects conducted in the village by the conductor in the form of pie charts, bar graphs etc. This is accomplished through the display_graphic_data function.
- In case, the admin selects the other option, the get_family_info function takes as input the family information of which the admin wants the view the details.
- After the family info is validated, the display_tabular_data function displays the information of this family in form of tables.

4. Glossary

- E-R Diagram (Entity Relationship diagram):- An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database.

- Use Case Model: - Use Case Model describes the proposed functionality of a new system. A Use Case represents a discrete unit of interaction between a user (human or machine) and the system.

- Coordinator: - User type who voluntarily work with an NGO and help in collecting various types of information of a remote village.

- Administrator: - User type who manages the working of the NGO, keeps a track on the working of the coordinators, maintain the information of the donors as well as view and analyse the information collected by the coordinators.

- Social Mapping: - It is designed to enhance the impact of such social investment by showing the extent of investment in relation to socio economic needs by geographic region and development sector. The map depicts this information in an easily accessible form on a Google Earth platform.

- Model: - The model manages the behaviour and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).

- Views: - The view renders the model into a form suitable for interaction, typically a user interface element. Multiple views can exist for a single model for different purposes.

- Controller: - The controller receives user input and initiates a response by making calls on model objects. A controller accepts input from the user and instructs the model and views to perform actions based on that input.