Team 16

NGO Information Management 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
2.0	Anshul, Lalit, Megha	Low Level Document for Phase 1 and 2, changes in the database, new modules added	Lalit	1st April, 2012
1.0	Anshul, Lalit, Megha, Parth	Low Level Design Document for Phase 1	Anshul	13th Feb, 2012

Table of content

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 update.

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

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.

3. System Components

3.1 Database tables

Family_info:

Fam_id	Fam_Head	Fam_no_of_members	Fam_no_of_children
Fam_last_migrated_from	Fam_traditional_occupation	Fam_no_of_children_school	Fam_daily_income
Fam_settlement_date	settlement_ID	Fam_water_connection	Com_ID
Fam_janani_support_status	Fam_widow_pension_scheme	Fam_vraddh_pension_scheme	Fam_housing_support
Fam_plot_card_Status	Fam_ration_card_category	Fam_ration_card_status	Fam_loan_application_status
Fam_electricity_status	Fam_no_of_handicapped	No_of_items	

Member Table:

Mem_ID	Fam_id	Mem_Name	Mem_birth_year	Mem_Occupation
Mem_Relation_with_Head	Mem_Gender	Mem_Job_Card	Mem_Voter_Status	Mem_Adhar_card

Values for this table:-

Mem_job_card-

  NOT_APPLIED(default), ISSUED, APPLIED_NOT_ISSUED ,
NOT_APPLICABLE

Mem_voter_status-

  NOT_APPLIED (default), ISSUED, APPLIED_NOT_ISSUED ,
NOT_APPLICABLE

Mem_adhar-

  NOT_APPLIED (default), ISSUED, APPLIED_NOT_ISSUED ,
NOT_APPLICABLE

NGO Table:

NGO_Name	NGO_Type	NGO_Contact_No	NGO_Office_location
NGO_username	NGO_password		

Settlement Table:

Set_Id	Set_Name	Vil_ID

Village Table:

Vil_Id	Vil_Name	Taluka_ID

Taluka Table:

Tal_ID	Tal_Name	Dist_ID

District Table:

Dist_ID	Distr_Name

Coordinator Table:

Coor_Id	Coor_Name	Coor_Contactno	Coor_joining_date	NGO_ID	username	password

Survey Table:

Fam_ID	Coor_ID	Survey_Date

Community Table:

Com_ID	Com_Name

Campaign Table:

Camp_ID	Set_ID	Camp_Name	Last_survey_date

Social Map Table:

SM_latitude	SM_longitude	SM_type	SM_Name	SM_village

Admin_info Table

Admin_id	Admin_name	Admin_contactno	Admin_join_date	NGO_id

Item_info Table

Item_id	Item_name

Projects Table

PROJ_ID	PROJ_NAME	PROJ_ST_DATE	PROJ_END_DATE
PROJ_MILESTONES	PROJ_CATEGORY	PROJ_DONORS	**PROJ_SETTLEMENT_ID**

3.2 Detailed level Description of Modules

3.2.1 Client Side Modules

Login Module: This module validates if it is an authenticated coordinator.

- When coordinator start the 'NIMS' application from the android phone, nims activity starts. In nims activity, login view sets and coord_login function called. The coordinator enters his username and password. The username and password validates in coord_login and on successful validation, 'CampaignInfo' activity starts.

Campaign Module: This module is used to take the information of the campaign by the coordinator.

- In 'CampaignInfo' activity,campaign_info view sets and get_campaign_information() called. The coordinator enters settlement name and village name. These 2 fields gets validated and receive_families_info() called. In order to receive family's information of the entered settlement on 'submit' button click, the current activity sends a HTTP request to the given URL with settlement name and village name as parameter. The HTTP response is an xml containing family's information. In order to parse the xml, parse_get_families_info() called. This function parses the xml and save the data into the local android database. After this, SelectProject activity starts.
- SelectProject activity sets the view and coordinator can select any project out of 3 NGO's on-going projects. The 'check_existence' activity starts on selection of family information project or 'SocialMap' activity starts on social map project or 'ItemDistributionName' activity starts on item distribution project selection.

Family Information Module: This module is used by the coordinator to maintain and manage the family information and its member information.

- The coordinator chooses whether he/she wants to create new family (then 'InsertUpdateFamilyInfo' activity starts) or update the information of an existing family (then 'familiesList' acitivty starts).
- In case of the new family, the coordinator enters all the family information. This family information gets validated and saved in the local android device's database. After successful submission, 'FamilyMenu' activity starts. In case of existing family, the coordinator selects one of the existing families on the basis of the family head name.
- The 'FamilyMenu' activity sets view of 4 buttons. One button is to view family information ('ViewFamilyInfo' activity), one to see the list of member in this family ('memberListInfo' activity), one to edit the status of the identity cards ('IdentityCardsMenu' activity) and one to edit the status of government schemes ('GovtSchemeMenu' activity).
- 'ViewFamilyInfo' activity sets the view to display the family information. On 'edit' button click,'InsertUpdateFamilyInfo' activity starts. The coordinator can now edit the family information. On 'submit' button click, the family information gets updated in the local android device database.
- The 'memberListInfo' activity displays the list of the name of all members of a family. The coordinator can view the information of any member. This starts 'ViewMemberInfo' activity.
- The 'ViewMemberInfo' activity sets the view to display the information of the selected member. The coordinator can edit the member information by clicking the edit button. This starts 'InsertUpdateMemberInfo' activity.The coordinator can now edit the member information. On 'submit' button click, the member information gets updated in the local android device database.
- The 'IdentityCardsMenu' activity sets the view of 5 buttons. The coordinator can select 'VoterCard' button (starts 'memberListStatus' activity),'RationCard' button (starts 'RationCardMenu'activity),'PlotCard' button (starts 'PlotCardType' activity), 'JobCard' button (starts 'memberListStatus' activity) and 'AdharCard' button (starts 'memberListStatus' activity).
- The 'memberListStatus' activity displays the list of all members with their respective status. The coordinator can edit the status and click on submit button. This results in the update of the local android device database.
- The 'RationCardMenu'activity displays the options of apply and issue. The coordinator can select either of these. This would result in the start of 'RationCardCategory' activity which displays the categories of ration card. The coordinator can select any one category and on 'submit' button click,the family information gets updated in the local android device database.

- The 'PlotCardType' activity displays the types of the plot card. The coordinator can choose any type and on 'submit' button click,the family information gets updated in the local android device database.
- The 'GovtSchemeMenu' activity sets the view of 4 buttons of various government schemes. On clicking any button, the 'GovtSchemeStatus' activity starts and displays the options of apply and issue.The coordinator can select any one and on 'submit' button click,the family information gets updated in the local android device database.

Social Map: This module is used to take the name of a place, type of place as an input by the coordinator. The latitude and longitude of the place is saved using Internet/GPS of the android device.

- The 'SocialMap' activity sets the view for coordinator to enter name of place, type of place. On Submit button click, the android device finds the coordinator's location and save it into the local android device database.

Item Distribution Module: This module is used to manage and maintain the information of item distribution by the coordinator to the families.

- The 'ItemDistributionName' activity sets the view for the coordinator to enter the item name. On clicking the submit button, 'familiesListItemDistribution' activity starts. The coordinator can enter the number of items corresponding to each family.

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_user 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 user is directed to the home page.

Community Module: This module is used by the administrator of the NGO to view and analyse the various information relating to the various communities.

- This module initially displays the analytical data of all the communities in the form of bar graphs using get_all_communities() function.
- When the user selects a particular community whose information has to be viewed, the get_one_community() function is called which displays the information regarding sex ratio, ration card status etc. in the form of pie charts. The districts where the community is present, average family income of the families belong to that particular community are also displayed.

Social_map Module: This module displays the social map of the corresponding district selected by the user.

- The user selects the district whose social map he/she wishes to view.
- The social map of the district selected is displayed.

Project Module: This module is used to display the information of all the projects undertaken by the NGO on a timeline which can be viewed by the user.

Tabular_data Module: This module displays the information of various districts, taluka, village etc. in a tabular form.

- The information of all the districts covered is displayed initially.
- The user can select the table whose information he/she wants to view from the side menu. The corresponding table is displayed based on the selection.
- Also, when a table is displayed the user can select a particular field to display further information relating to that particular field only. For example, if a user selects the district "Sabarkantha", information related to only the talukas present in that district are displayed.

Settings Module: This module allows the user to view, edit, add and update information of the coordinators, projects, village etc.

change_password: The user can change the password by providing his/her old password. The old password is checked whether it is correct or not. In case the old password entered by the user is correct, the password is updated.

add_new_coordinator: TheNGO admin can add a new coordinator to the database by providing all the required information.

edit_coordinator: TheNGO admin can edit any of the coordinator's information. The NGO admin can select the coordinator from the list displayed whose information he/she wishes to edit. Complete information of the selected coordinator is displayed which can be edited by the admin. Once the edited information is submitted by the admin, it is tested for validation. In case the information submitted is valid, the information is updated in the database.

add_new_project: TheNGO admin can add a new project to the database by providing all the required information.

edit_project: TheNGO admin can edit any of the existing project's information. The NGO admin can select the projects undertaken by the NGO from the list displayed whose information he/she wishes to edit. Complete information of the selected project is displayed which can be edited by the admin. Once the edited information is submitted by the admin, it is tested for validation. In case the information submitted is valid, the information is updated in the database.

add_village: The NGO admin can add a new village to the database by selecting the appropriate taluka where the village belongs.

add_taluka: The NGO admin can add a new taluka to the database by selecting the appropriate taluka where the district belongs.

Help Module: This module provides a brief description of each of the above functionalities to the user.

Logout Module: This module facilitates the user to logout of the system.

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.