Team 16

NIMS Suite v1.0

High Level Design Document

Overview

This is the high 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 overview which covers information regarding the application environment and the hardware and software requirements. The fourth part is the system design which contains relationship design, and database design, use case diagrams and the design architecture.

Target Audience

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

Revision History

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

Table of Contents

1. Introduction

A software design document (SDD) is a written description of the software product, that a software designer writes in order to give a software development team an overall guidance of the architecture of the software project.

1.1 Purpose

The purpose of this document is to outline the technical design of the NGO Information Management Suit and provide an overview for its implementation.
Its main purpose is to –

- Provide high level design of our system
- Provide the link between the Functional Specification and the detailed Technical Design document
- Document the functionality provided by each module or group of modules and show how the various components interact in the design
- This document is intended to help the coding team to build our system.

1.2 Document Overview

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 Overview: Describes User Characteristics, User Objectives, Hardware and software Requirements, interfaces and Design Constraints
- System Design: Contains use case diagram, E-R diagram, Data-flow Diagram.
- Architecture: Describes architecture design and briefly functionality of various modules and how group of modules interact to provide common functionality.
- Glossary

1.3 Scope

This document contains the high level design which shows the various modules and how they interact with each other thus enabling our system to work. The Application Design outlined in this document builds upon the scope defined in the Requirements phase. This document will serve as a link between design team and coding team and will be frequently referred by coding team to build our system.

1.4 References

- Software Requirements Specification
- Project Plan

1.5 Methodology, Tools, Techniques

- Dia
- Microsoft Word

1.6 Key Stakeholders

- VSSM (NGO)
- Application Developers

1.7 Intended Audience

This document is meant for the technical team for reference during the implementation phase.

2. Design Overview

We aim to develop a web based information management system that takes care of the basic needs of cataloguing their work, evaluating the NGO and its employee(and community) and coordinator'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, which's relational and ER

diagram are depicted underneath. The architecture of the website and automatic error detection and information retrieval system on Android is also provided.
Business Angle: The NGO maintains all its data by far, on paper and it takes time for it to reach to NGO from volunteers working in remote locations. Also, the form in which the data comes is very weird (as people there don't have much knowledge of providing information). To solve this problem we develop an android application, which takes data in very tabular form, as to be filled on server side database and transfers it in real time to NGO server PC.

2.1 Background Information

At system level we need to understand the user requirements very clearly before designing the database and the information which the NGO generally requires from its volunteers. For the same we have analysed the various forms, they ask their volunteers to fill, like collecting family information, children studying in school, various charity items given to people, information about various cards and government schemes for tribal denotified tribes, etc. We will also integrate in the database information needed by various government run schemes, for tribal people in Version 2 as per extended need of the NGO, as told in interview with them.

2.2 Current Process

The current work done for designing includes:

1.) Logic Design: Understanding the requirements and dividing the project into 3 modules - server, client android app and website.

2.) Technical Design: Further we have done 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.

2.3 Constraints/ Risks

1.) Used “Dia” as software tool for E-R Diagram.

2.) Assuming that requirements are as the client has informed us and only what they work on till now. We have not added new features, except that of social mapping.

3.) To add new information collection forms, the database has to be added with more fields at later stage, which may cause some problems.

2.4 Guiding Principles

- Scalability - It can be scaled for more than one NGO, but as different firms have different demands, this project is only for a particular NGO. The database can however be scaled within the framework of one NGO, by adding new fields in tables.

- Maintainability - It is easy to maintain, as only the system requirements to handle data and proper entry of data from android mobile phones has to be ensured, rest is all automated updation and data extraction.

- Portability - It's a system tested for only Windows OS.

3. System Overview

3.1 User Characteristics

1. Literacy- Yes.
2. Education- About 10th grade
3. Language Proficiency- Gujarati speak/write/read
4. English language

Initially the NIMS build will be shipped in with an interface in English language only. Until local language support is provided the system might be of very less use. On the other hand, as we know typing in Gujarati/Hindi can be very cumbersome even a local language support may create user interaction issues if the user interface is not developed taking into consideration this issue.

5. Computer literacy

The users have no prior issue in handling a computer or a computer like device. At the most, they have used java based mobile phones. So adapting to an android OS based device will take time if a proper interface is not provided which helps them adapt to the new system quickly.

3.2 User Problem Statement

1. Currently in the pen and paper based approach the user (field coordinator) is expected to keep track of his activities over the month and then submit a set of documents detailing his monthly activities. Since there is no constraint on keeping the information objective enough and to the point, the user ends up sending lots of data that may be of very less help to the NGO and a wasted effort for the coordinator.

2. The NGO personnel on receiving the data from the field coordinators have to first go through the process of transferring the pen-paper work to their computers before doing any analysis task. This is an overhead.
3. The field coordinators have to come to Ahmedabad from various places to submit the documents as of now. There is travel money that is spent in addition to one person day lost.
4. The social maps currently made are on chart paper and purely based on coordinator’s estimation of a place which can be totally incorrect at time. Again, effort is wasted along with inefficient work.

3.3 User Objectives

Outline the users’ objectives and requirements for the new system. Where appropriate, include a "wish list" of desirable features.

1. Seamless interface
2. Social Mapping on the go
3. Real time data feed from field to office headquarters
4. Data Visualization for the NGO personnel

3.4 Hardware Requirements

The project would require a basic android device (due to inbuilt GPS) and a personal computer with Windows operating system. The android device would be required for NGO coordinators and personal computer for the NGO to run a server.

3.5 Software Requirements

The project would require an Eclipse IDE to build the Android app later tested on the Android Emulator plugin within Eclipse, PHP for server side programming, MySQL to maintain and manage database at server side, WAMP server to provide a local server for running and testing PHP scripts at sever side and Google Map API or Jquery plugins like craftman to implement the social mapping feature on the public website of the NGO. The project would require a domain to make the NGO server live. Only then the coordinator would be able to submit his forms into the NGO server database.

3.6 Interfaces

There will be three interfaces:

- **Coordinator Interface**: In this interface, first the coordinator needs to login into the application and then he/she can submit the family information, can check the status of a family and can update the information of the family. The coordinator can also submit the information of any place. The distribution campaign information, tent school information can also be submitted though this interface.
- **NGO Admin Interface**: In this interface, first NGO admin would login and then he/ she can see the collected data and analyse the data. The data would include the village information (family information), the social map of the village, the volunteer’s performance, the distribute campaign and the housing project information.

- **Public Website**: The NGO would have a public website. The public website would display the working of the NGO, its projects, its events, its media coverage. Donor’s information and NGO contact details. This interface is for public.

4. System Design

4.1 Use Case

Users of the system will be the coordinator (Client Side) and the administrator (Server side) of the NGO. Each type of user has different needs and hence performs different functions as mentioned in the Software Requirements Specification.

1.

Use Case	1. Client Login on Android Device
Description	Logging into a android device
Actors	1. NGO Coordinator 2. Software Maintenance personnel
Assumptions	1. Login screen is always available first on opening the client application. 2. Login password and username are known
Steps	1. Enter Username 2. Password

Variations	1. If username and password have been set to “remember me” than click login straightaway. 2. Wrong Password and Wrong Username 3. Wrong password or Wrong Username
Non-Functional	Registration not to be provided at client side.
Issues	What if coordinator forgets password?

2.

Use Case	2. Client Logout on Android Device
Description	Logging out from an android device
Actors	1. NGO Coordinator 2. Software Maintenance personnel
Assumptions	1. logout option is always available across all screens in a session
Steps	1. Click “Logout”
Variations	None
Non-Functional	Inform actor about logout success by showing a login screen and a message-”You have logged out successfully”
Issues	1. GPRS/3G connection of the phone is lost. 2. Sudden Battery down 3. Request Timeout

3.

Use Case	3. Server Login
Description	Logging in at the Server Side.
Actors	1. NGO Admin. 2. Software Maintenance personnel
Assumptions	1. Web Interface is reachable. 2. Login screen is loaded on reaching the server side web interface. 3. Login username and password are known
Steps	1. Enter Username 2. Password
Variations	1. If username and password have been set to “remember me” than

	click login straightaway. 2. Wrong Password and Wrong Username 3. Wrong password or Wrong Username
Non-Functional	None
Issues	1. What if Password or username is/are forgotten?

4.

Use Case	4. Server Logout
Description	Logging out at the Server Side.
Actors	1. NGO Admin. 2. Software Maintenance personnel
Assumptions	1. logout option is always available across all screens in a session
Steps	1. Click Logout
Variations	None
Non- Functional	None
Issues	None

5.

Use Case	5. Submit Social Map datum
Description	Sending geographic information of a place type and name to server.
Actors	1. NGO Coordinator 2. Software Maintenance personnel
Assumptions	1. GPRS/3G connection already established 2. Coordinator has logged into a session.
Steps	1. Fill in name of the place. 2. Fill in type of the place from the drop down list. 3. Submit information. (GPS information fetched by the device and embedded into the outgoing packet appropriately.)
Variations	1. If information is already present in the database then return a message informing about it. 2. Type and Name are both required fields so submission not possible without their entries.
Non-	Inform actor about logout success by showing a login screen and a

Functional	message-”You have logged out successfully”
Issues	1. What is GPS data is not reachable? 2. What if GPRS connection is lost in between activity?

6.

Use Case	6. Submit family information
Description	Posting a Form filled for family information to the server
Actors	1. NGO Coordinator 2. Software Maintenance personnel
Assumptions	1. GPRS/3G connection already established 2. Coordinator has logged into a session. 3. Village and Program information already filled.(via GPS)
Steps	1. Fill in the details of a family 2. Submit.
Variations	1. If information is already present in the database then return a message informing about it.
Non-Functional	Inform actor about logout success by showing a login screen and a message-”You have logged out successfully”
Issues	1. What if GPS data is not reachable? 2. What if GPRS/3G connection is lost in between activity?

4.2 E-R Diagram

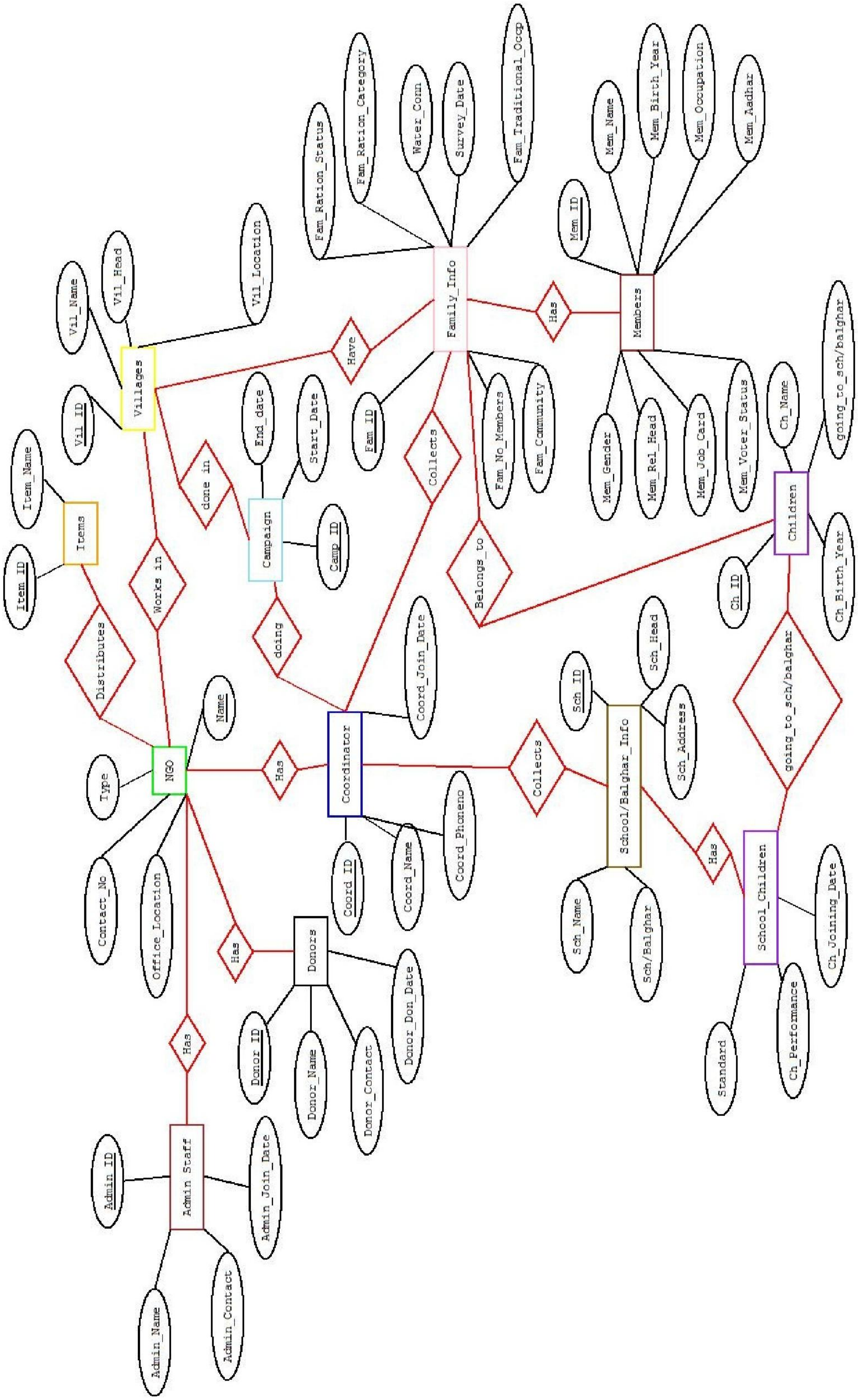
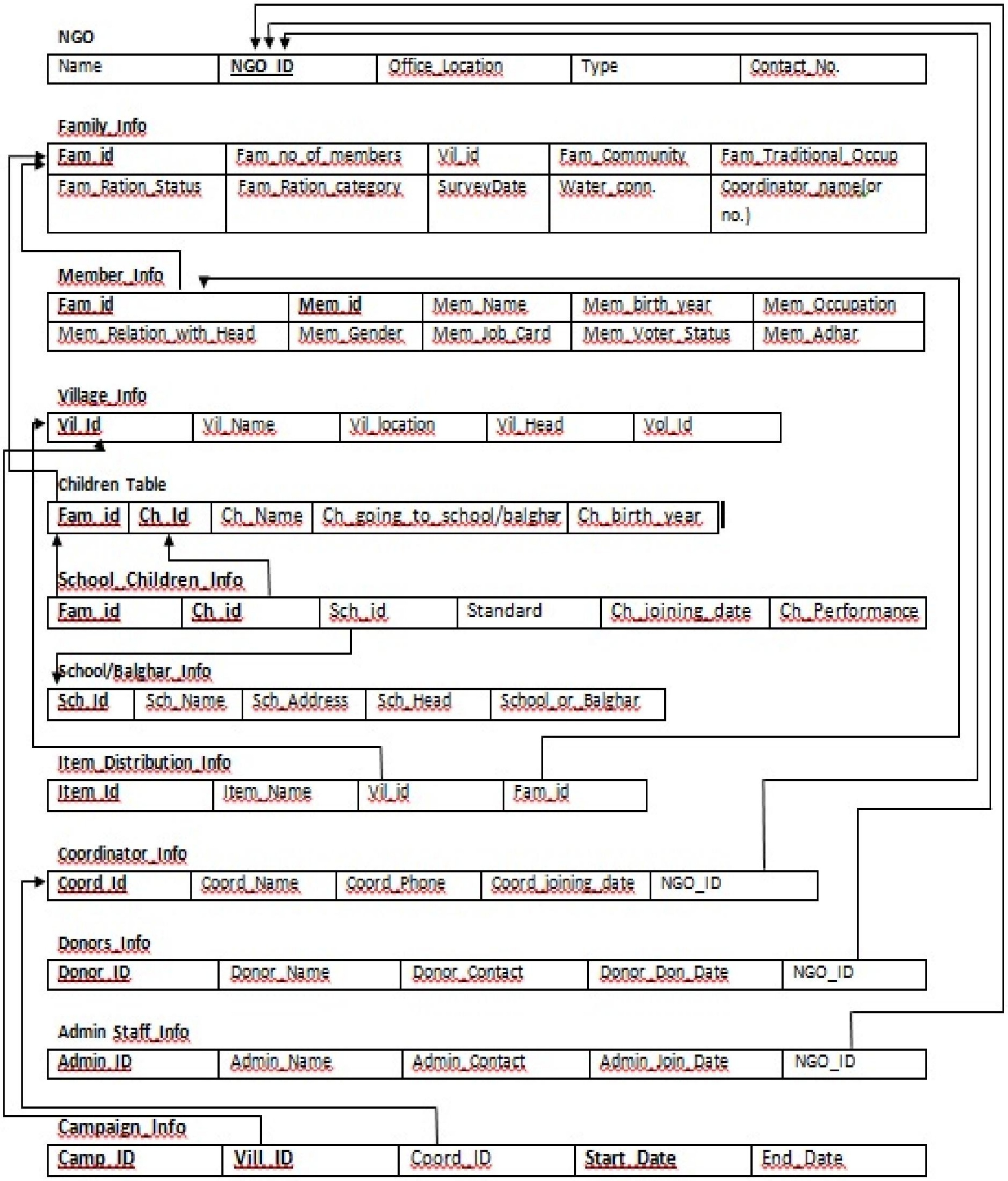


Figure 1:- ER Diagram

4.3 Relational Schema



4.4 Dataflow Diagrams

4.4.1 Client Side

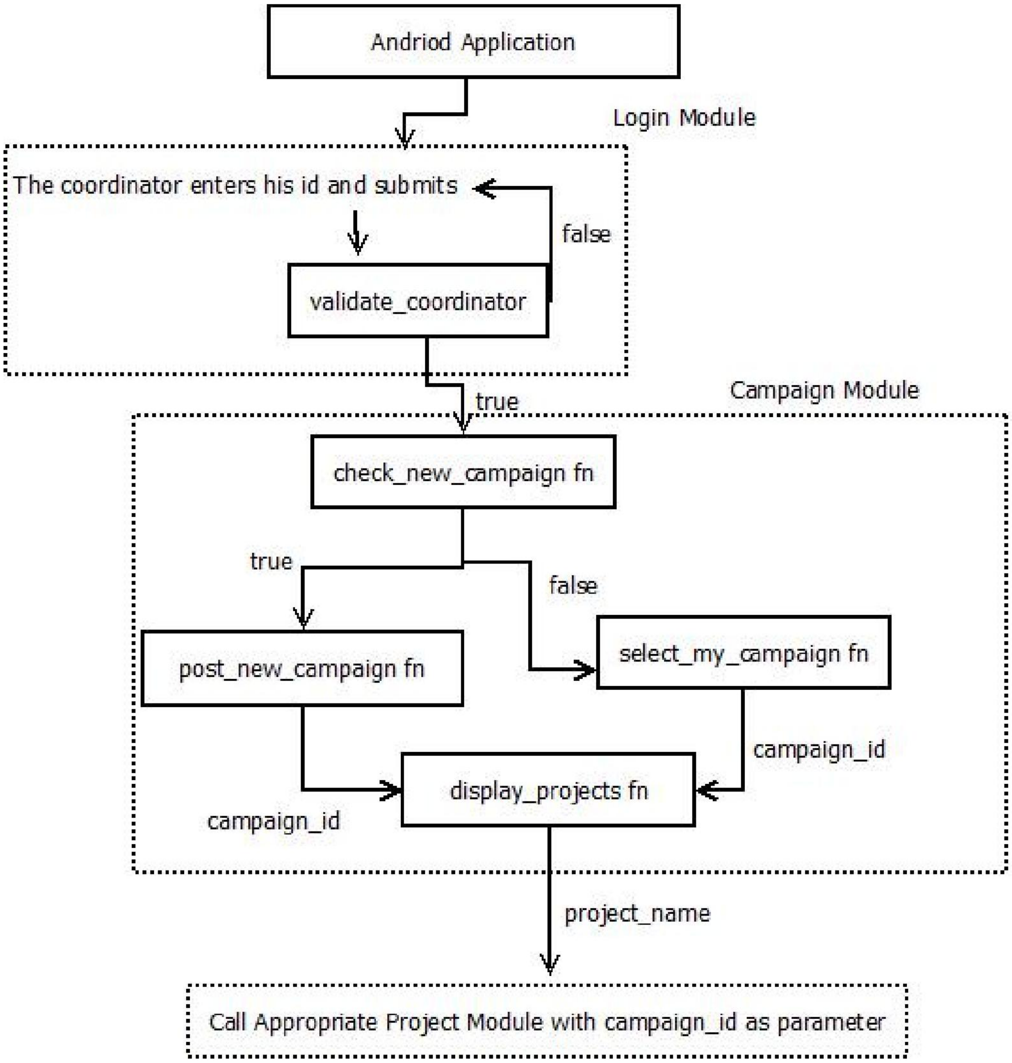


Figure 2:- Dataflow Diagram when the coordinator logins

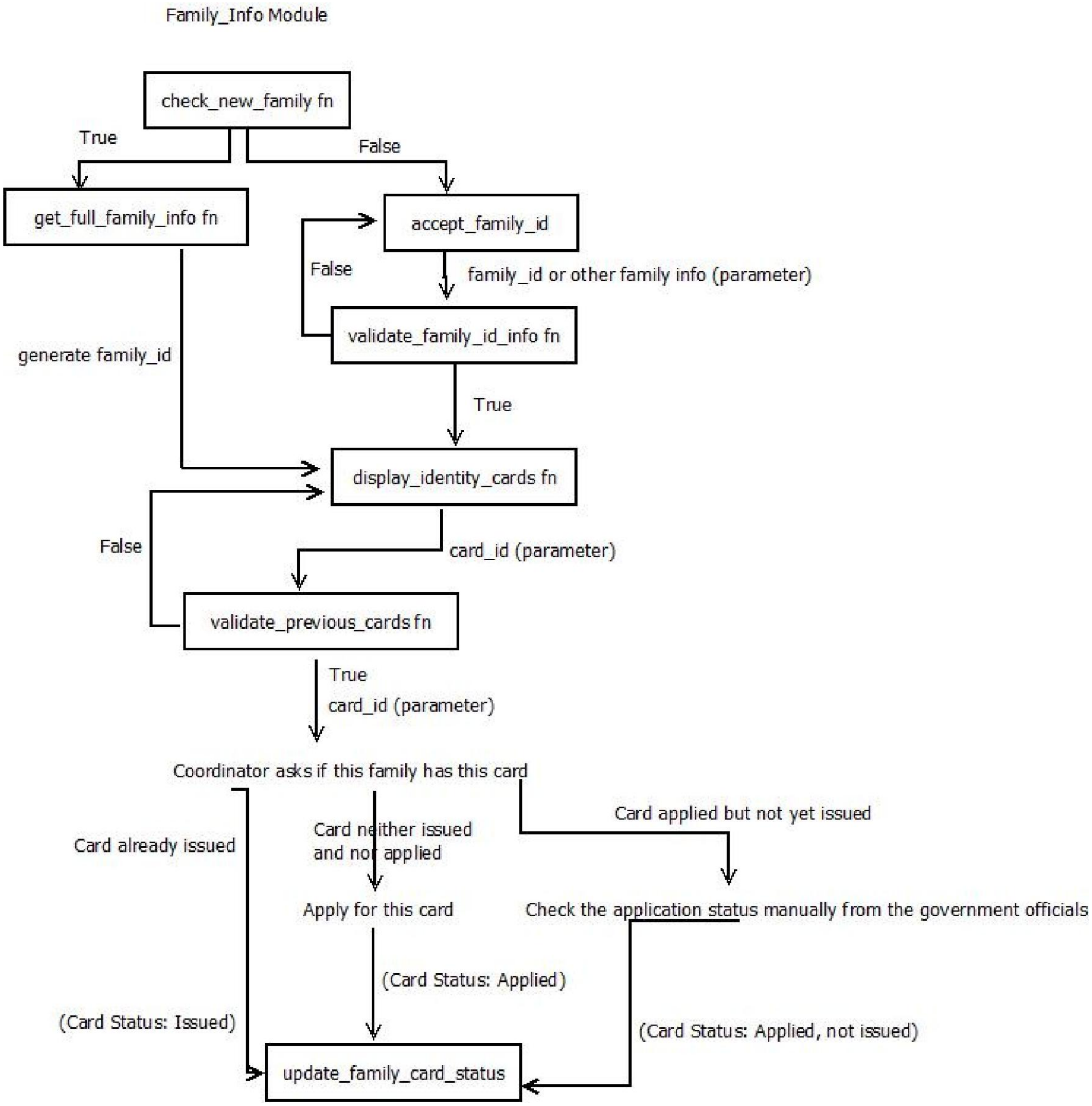

Family_Info Module
check_new_family fn
True
False
get_full_family_info fn
accept_family_id
False
family_id or other family info (parameter)
validate_family_id_info fn
generate family_id
True
display_identity_cards fn
False
card_id (parameter)
validate_previous_cards fn
True
card_id (parameter)
Coordinator asks if this family has this card
Card already issued
Card neither issued
and nor applied
Card applied but not yet issued
Apply for this card
Check the application status manually from the government officials
(Card Status: Applied)
(Card Status: Issued)
(Card Status: Applied, not issued)
update_family_card_status

Figure 3:- Dataflow Diagram for Family_info module

4.4.2 Server Side

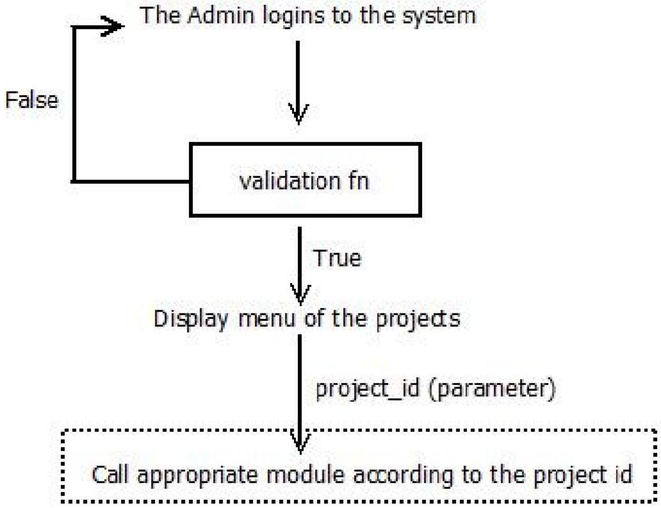

The Admin logins to the system
False
validation fn
True
Display menu of the projects
project_id (parameter)
Call appropriate module according to the project id

Figure 4:- Dataflow Diagram when a administrator logins

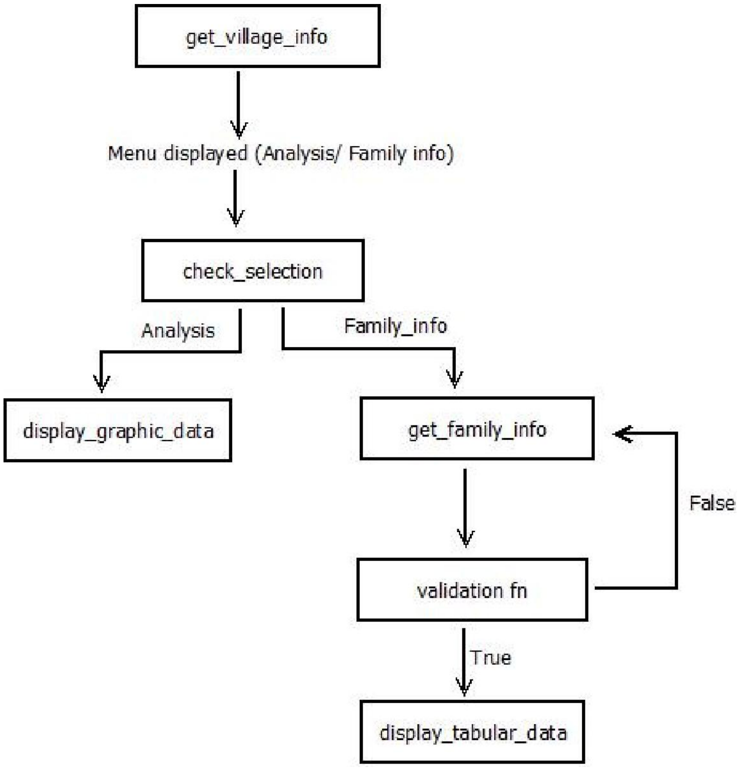


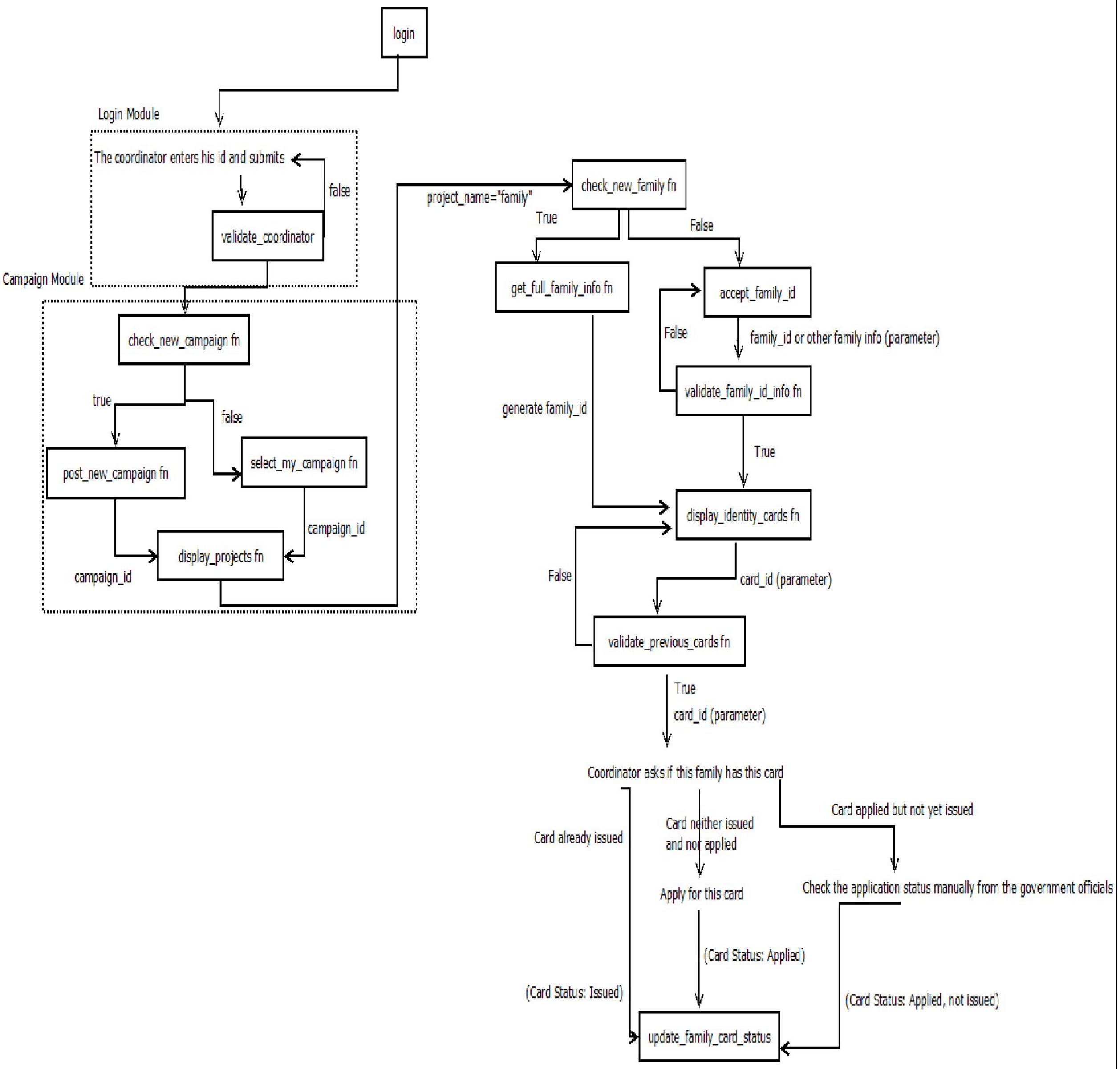
Figure 5:- Dataflow diagram for family_info module

4. Architecture

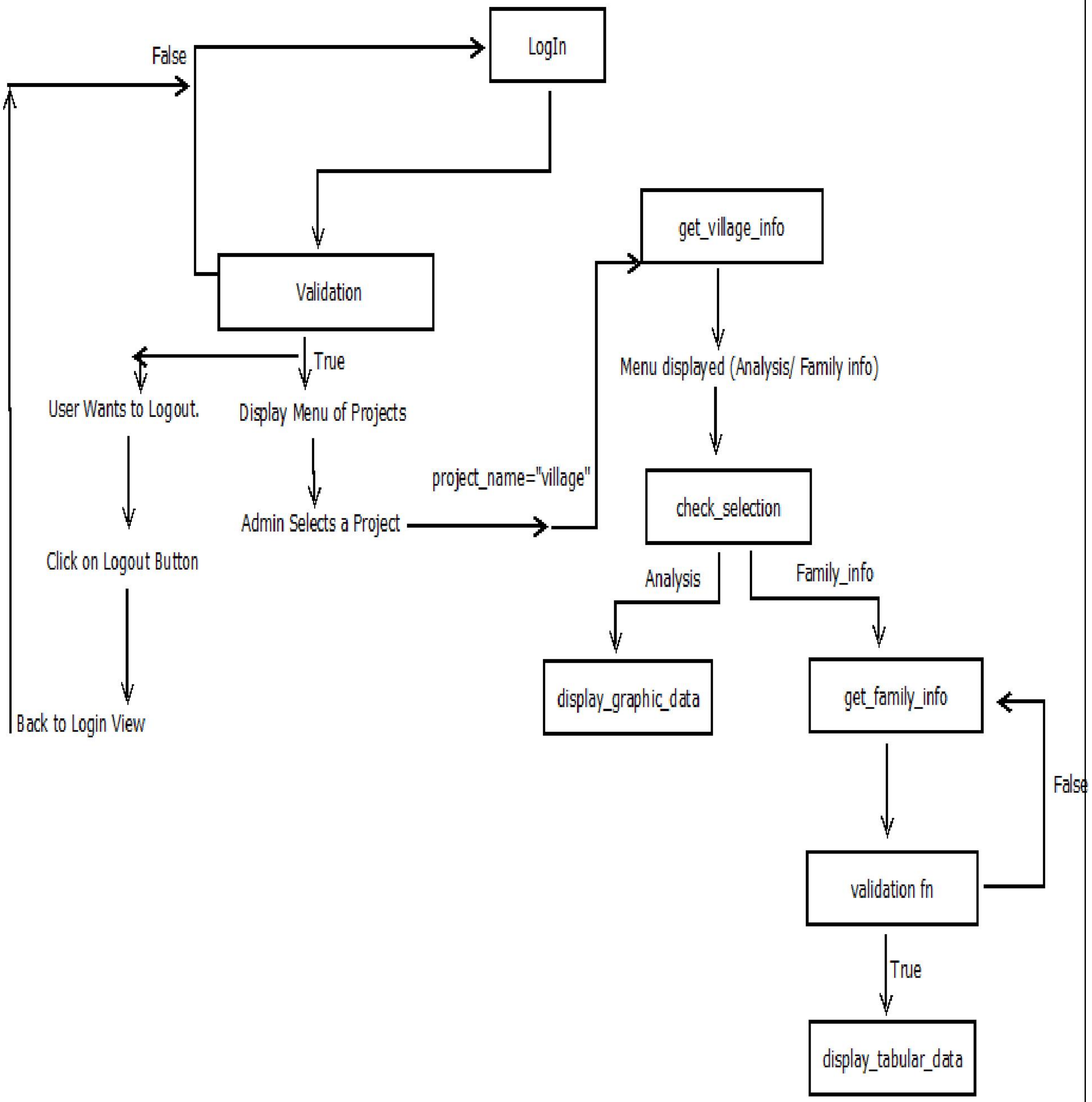
Architecture is the hierarchical structure of program components or modules, the manner in which these modules interact to build our system which caters to the requirements as specified in SRS document.

4.1 Architecture Design

4.1.1 Architecture Design for the Coordinator



4.1.2 Architecture Design for the NGO Admin



4.2 High level Description of Modules

4.2.1 Client Side Modules

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

Campaign Module: This module is used to manage and maintain the information of the campaigns by the coordinator.

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

4.2.2 Server Side Modules

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

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.

5. 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.