< Case: 1 >

Software Requirement Specification

Group: 13

by

Aayush Choubey	(111915002)
Adarsh Singh	(111915006)
Nachiket Punjabi	(111915075)
Prasad Dalwee	(111915092)
Siddhant Sharma	(111916052)

Under the guidance of:
Ms.Yesoda Bhargava

Edit History:

Version Number	Date	Brief Description
1.0	11/09/21	Draft
1.1	12/09/21	Addition of UML diagrams and description
2.0	10/10/21	Structural changes w.r.t wireframe
2.1	14/10/21	Changed class diagram
3.0	23/10/21	Restructured whole document
3.1	24/10/21	Added testing and designing part
3.2	25/10/21	Added new wireframe for NGO app
3.3	26/10/21	Added state diagram, new images for wireframe, rearrangement of sections

Note: Handling of images and text in google docs is rigid, which creates unnecessary whitespaces. Try to avoid accidental typing.

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

This document describes the features, interface, underlying system, and constraints of the proposed system for the analysis and collection of data regarding mosquito-borne diseases in rural areas. It also describes how to interact with and use the application. This document is applicable to both the developers and stakeholders of the system and is used to outline the system.

1.2 Scope

This application system will be used to collect the information from rural populations and also provide various features to analyze that information. More specifically, design and develop a simple application that acquires and saves information of individuals along with a website that will display and analyze all that data.

1.3 Definitions

1.3.1 NGO App :

The NGO App acts as an interface for the NGO staff to take information from the villagers about their health, sanitation, personal info, etc., and save it to a local database and then to the main database whenever possible.

1.3.2 Analysis Website :

Analysis website is meant to facilitate the analyst to query about the collected information to perform numerical and graphical analysis.

1.3.3 NGO Staff/Worker :

The NGO staff members are responsible for collecting data from rural areas. They need to be well versed with working of the application and are proficient in local languages so as to efficiently communicate with the rural masses.

1.3.4 Analyst :

Analysts are trained professionals who will use the website to analyze all the information.

1.4 Overview

Please note that this is a baseline document and may be updated as development progresses.

This document is structured as follows:
Section 2: Gives an overview of the system.
Section 3: Shows an overview of the code implementation, testing and proposed interface
Section 4: Gives an overview of functional and non-functional requirements.

2. Overall Description

To describe our approach towards design and development we will first discuss the basic outlook such as user interaction and core functionalities. And then we will gradually move towards a detailed discussion of functions, their data flow and their complexities.

2.1 User Interaction:

The system is divided into two separate parts:

1. **NGO app**: Used for data collection and operated by NGO workers.
2. **Analysis engine**: Used for analyzing data and operated by data analysts.

2.1.1 NGO Application:

The interaction by workers is required at 2 stages :

a. **Verification:** The worker has to verify himself in order to access the features of the application.
b. **Data Acquisition**: The worker acquires data from the villager by filling forms.

The use case diagram(fig. 2.1.1.a) illustrates the interaction between an NGO worker and the application

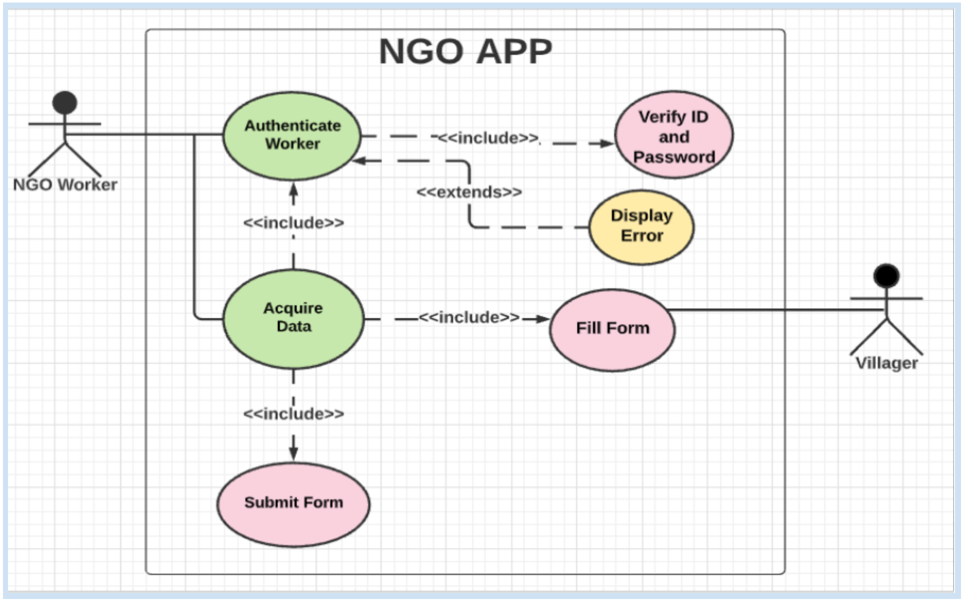


Fig 2.1.1.a. Use Case Diagram for NGO Application.

2.1.2 Analysis Engine:

The interaction by data analyst is required at 2 stages :

a. **Verification**: Data analyst has to verify himself in order to access the features of the analysis engine

b. **Data Analysis:** The analyst has to interact with the engine in order to set variables and get desired analysis. He can choose variables as well as the type of analysis (graphical or numerical).

The use case diagram(fig. 2.1.2.b) illustrates the interaction between a data analyst and the analysis engine.

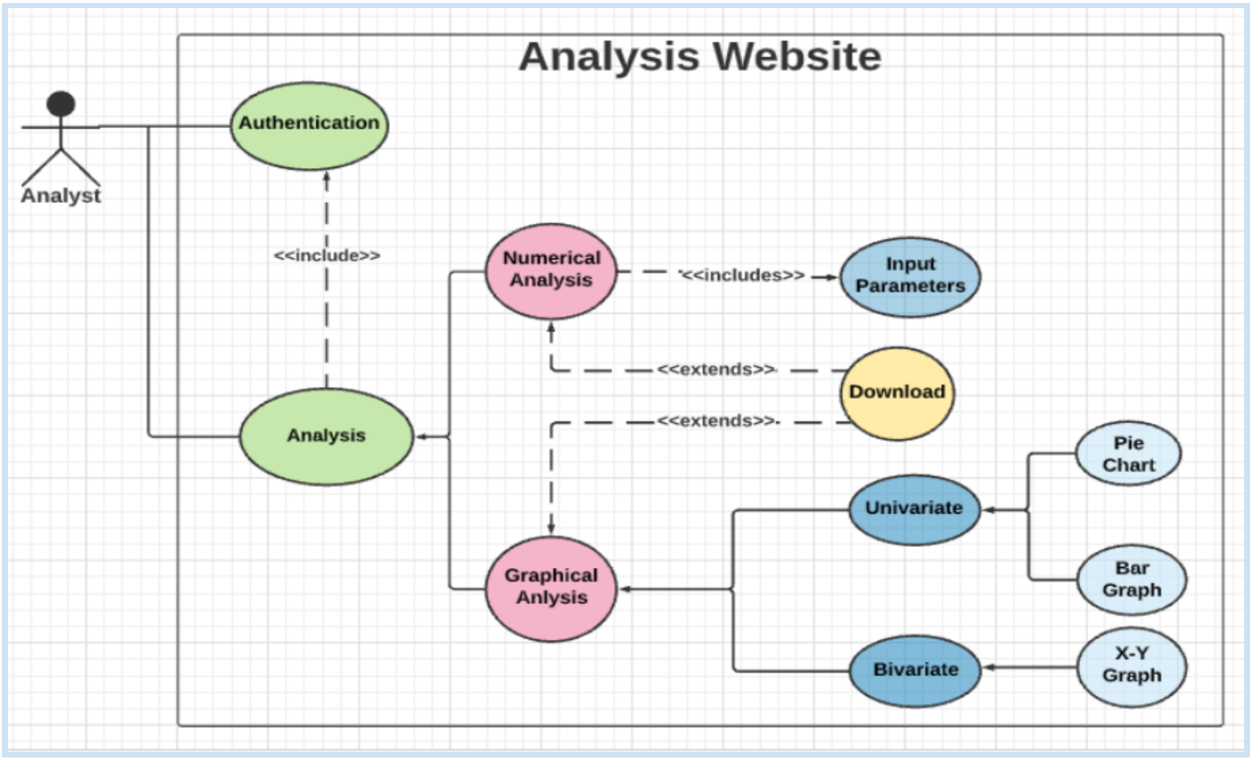


Fig 2.1.2.a Use Case Diagram for Analysis Engine.

2.2 Core Functionalities:

As stated before the NGO app and the analysis engine serve two separate purposes and thus have different functions and functional decompositions.

2.2.1 NGO App:

NGO app has three main functionalities:

1. **Login:** For worker verification.
2. **Data acquisition:** To collect the data from users.
3. **Data synchronization:** To store the collected data into a local database. To upload entries stored in the local database to the main database once the connection is established.

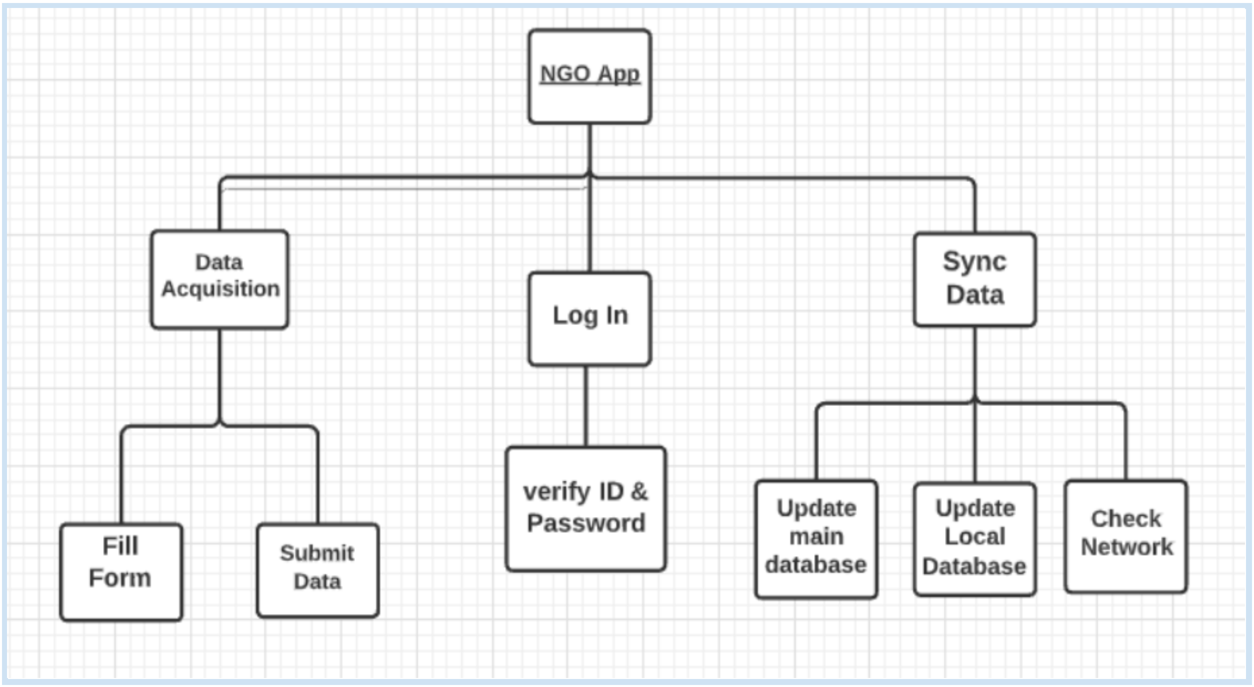


Fig 2.2.1.a Functional Decomposition Diagram for NGO Application.

2.2.2 Analysis Engine:

Analysis Engine has four main functionalities:

1. **Login:** For analyst verification.
2. **Data Fetch:** To fetch the data stored in the main database.
3. **Numerical Analysis:** Analysts can select the combination of required parameters to get statistical analysis and also can download the analysis.
4. **Graphical Analysis:** To get a visualization of the data and its distribution.

Analysts have to select the desired type of graphical analysis(univariate or bivariate) and can also select the type of plot in univariate analysis. Analysts can download the analysis.

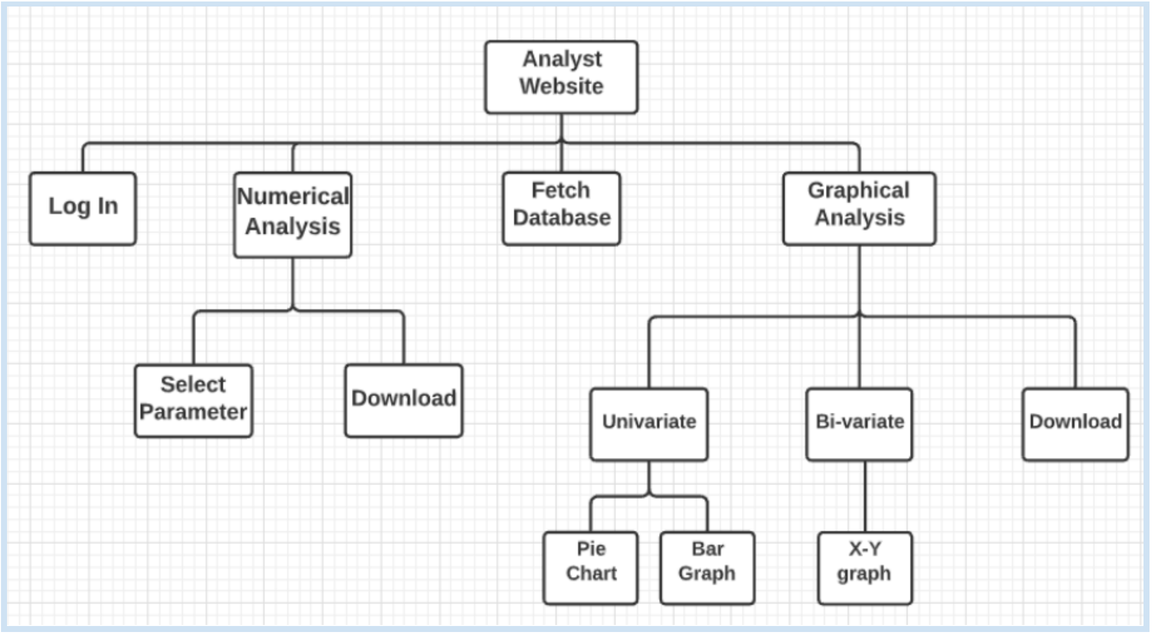


Fig 2.2.2.a Functional Decomposition Diagram of analysis engine.

2.3 Data Flow:

Generalized data flow in the entire system can be visualized by the following diagram (fig 2.3.a.)

The information from villagers is taken by NGO workers in the form of questionnaires, all the information of each villager is stored in one row in the local database. These rows are then sequentially passed into the main database once the internet connection is established. Once the data in the local database gets uploaded into the main database, analysts can perform their analysis on the updated data with the help of our analysis engine (deployed as a website).

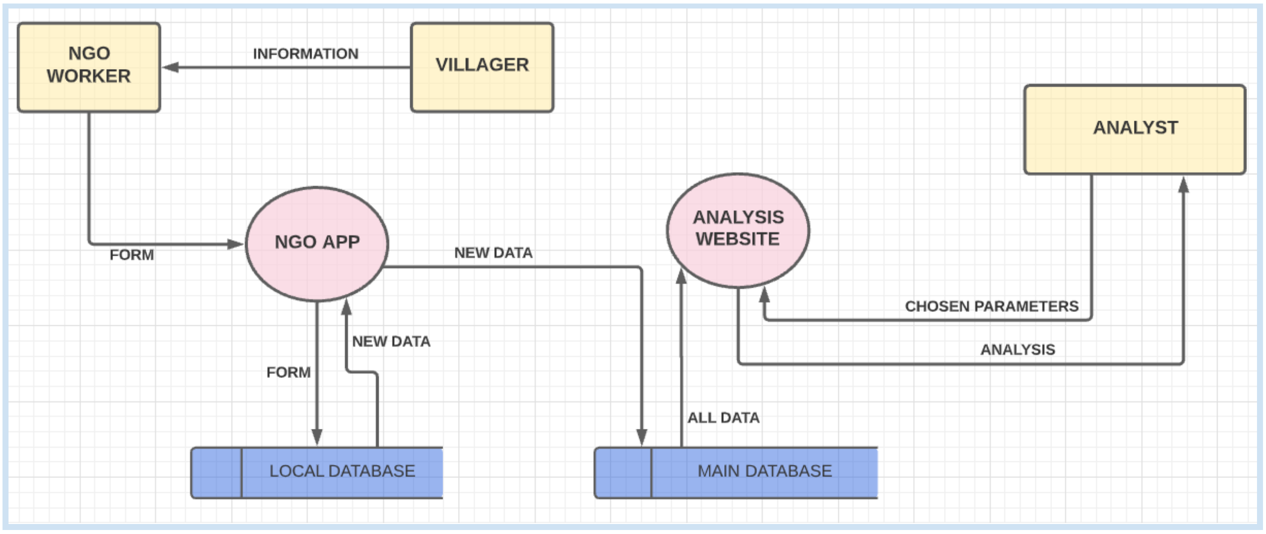


Fig 2.3.a level 0 DFD of the system, describes general overview of the data flow

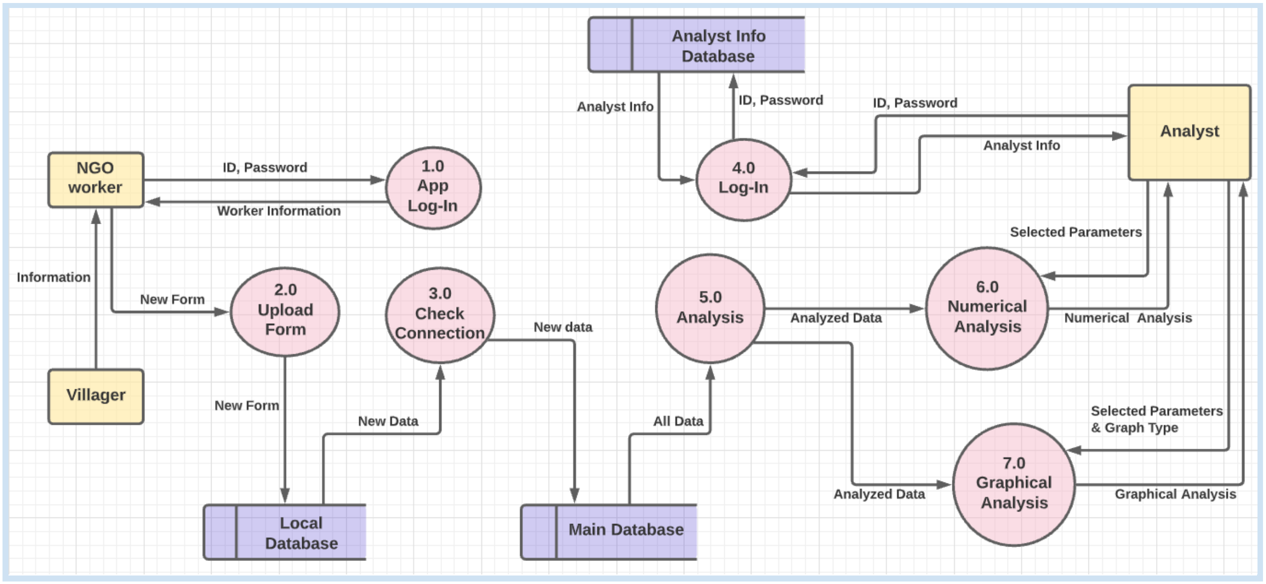


Fig 2.3.b level 1 DFD of the system, describes a detailed overview of the data flow.

2.3.1 Analysis process:

The data from the main database is fetched and preprocessed by our engine. This preprocessed data then gets divided field wise. These fields are the parameters that the analyst will choose for analysis. Based on the type of analysis required this field-wise data is processed to give statistical and graphical analysis. Fig 2.3.1.a illustrates all these subprocesses.

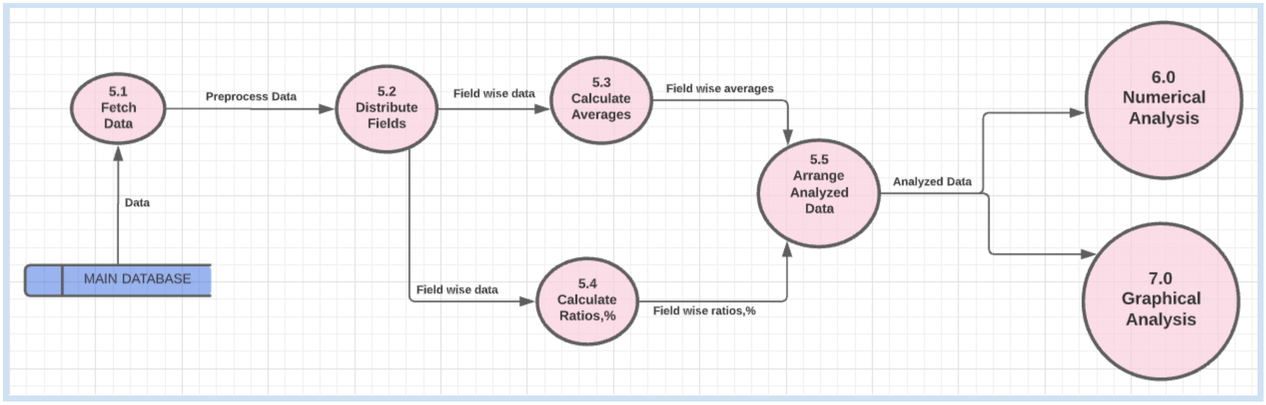


Fig 2.3.1.a Level 2 DFD: Analysis process

2.4 SDLC:

We have chosen the **spiral model** as our SDLC as it specifically emphasizes the risk management aspect of development. Our project till now has gone through many revisions in terms of implementation and requirements. We started with a simple react and dart(Flutter) based code to read and write local databases, as we moved towards implementing and integrating further complex functionalities we had to rethink our approach and change our platforms for development. Currently, we have completed the implementation, testing and debugging of all the desired core functionalities in the python language in Jupyter Notebooks. We are yet to create a frontend.

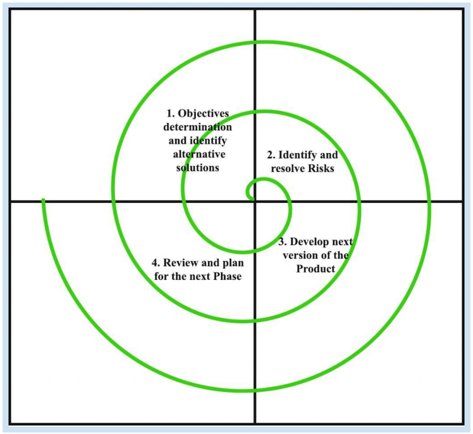

1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

Fig 2.4.a Spiral Model

3. Implementation, testing approach and proposed interface

3.1 General overview :

While the functions and their purposes are the same as discussed in the above section, they have been divided into modules/classes for encapsulation.

We have three main classes in the NGO app:

1. **NGO_App:**
   It is the skeleton of the NGO application and sequences all the subparts.
2. **Form:**
   Handles the questionnaire and data entry part of the application.
3. **NGO_Worker:**
   Handles the login part.

There are two main classes in Analysis Website:

1. **Analysis_Engine:**
   Serves as the skeleton of the website and like NGO_App class handles the sequencing of all the subparts.
2. **Data_Analyst:**
   Handles the login part of the website.

The diagram below(fig 3.1.a) shows the interdependence of these modules. The functions mentioned in each class are the same as discussed in the previous section. The attributes, on the other hand, serve the same purpose as their names suggest in their respective classes.

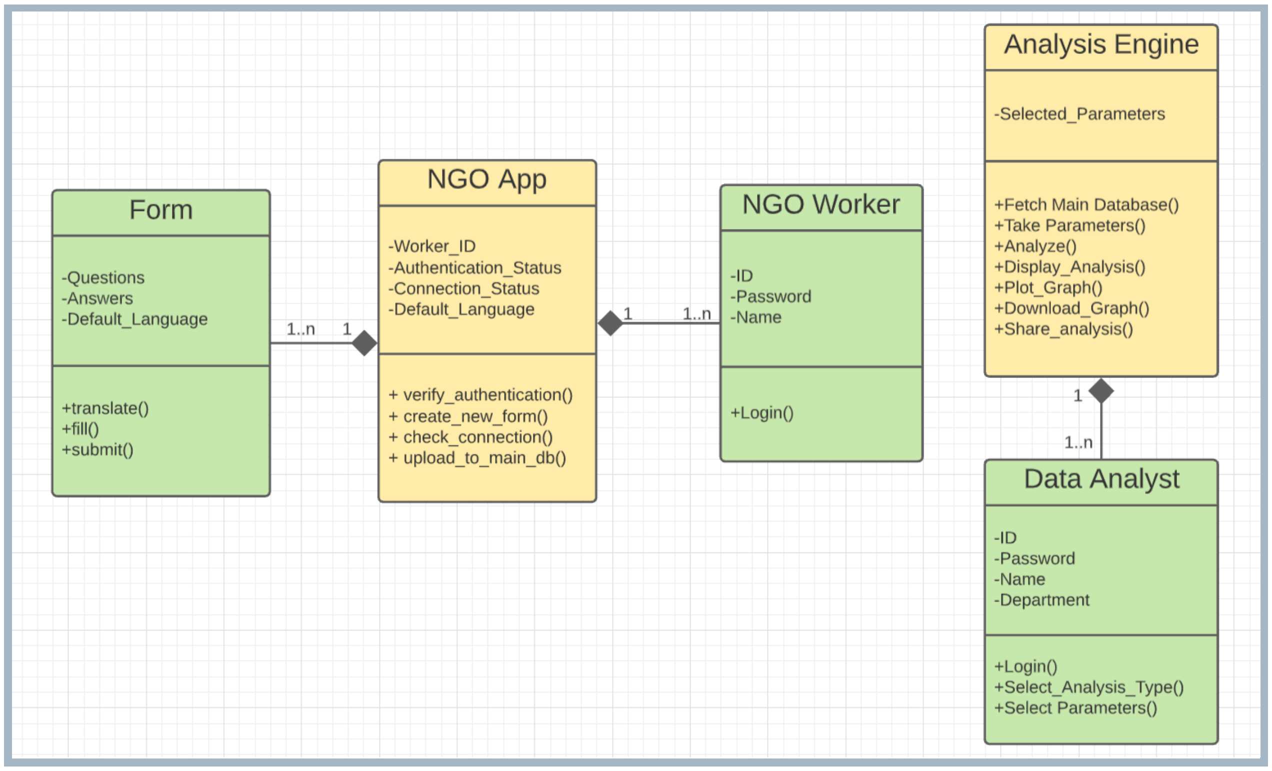


Fig 3.1.a Class diagram of both the systems.

3.2 Designing Approach:

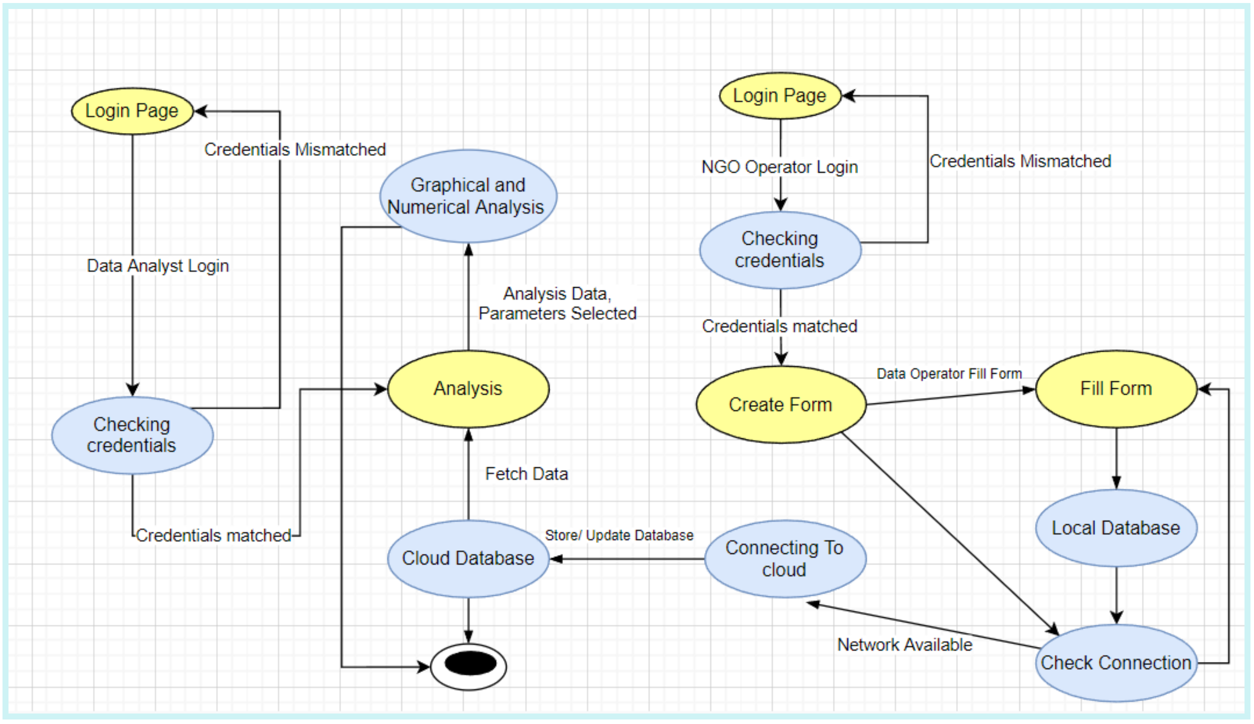
We have incorporated a bottom-up design approach. We initially created functionalities, then we divided them into classes and then we integrated functions in the classes. Finally, we integrated all these classes to form one complete system.

3.3 Testing Approach:

The approach used for testing was also bottom-up. As we were building the model in a bottom-up manner the bottom-up approach for testing seemed adequate. We tested each functionality individually then we checked its integration with higher modules. We have used Jupyter Notebook which helped us in both designing and testing as we could separately develop and check modules in each kernel, as well as integrate them.

3.4 State Diagram:

This state diagram summarizes the flow of our system. Check connection function will be running in the background and will be activated from time to time when the user logs in and when an entry is submitted in the database. The graphical and numerical analysis can now be performed simultaneously so no need for separate functions.



3.5 Proposed Interface:

We are yet to develop a proper interface for both of our systems. These are the wireframes that are used to simulate the basic outlook of the interface and the position of objects in it.

3.5.1 NGO APP:

3.5.1.1 Login page:

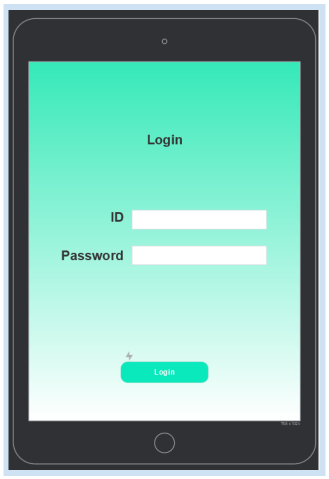
This is the first page where the worker will land when he opens the application.

3.5.1.2 Create Form page:

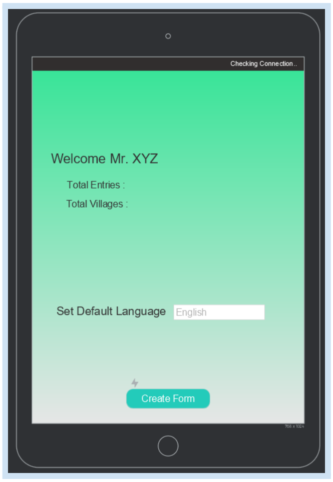
The worker will have the option to set a default language for all the forms. He can reset it again as he will be taken back to this screen after filling each form.

3.5.1.3 Form page:

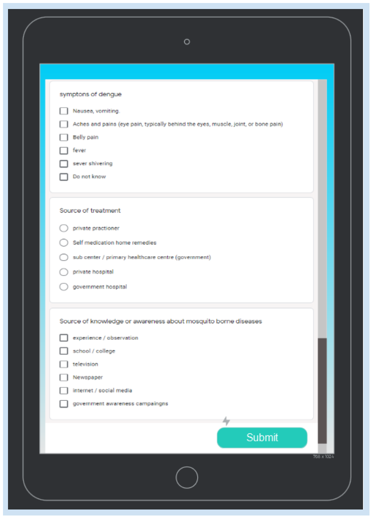
After the worker taps the 'Create Form' button on the previous screen, he will be taken to this screen where he can fill the form.
Following are the images of the pages mentioned above.



3.5.1.1.a Login Page



3.5.1.2.a Create Form Page


symptons of dengue

☐ Nausea, vomiting.
☐ Aches and pains (eye pain, typically behind the eyes, muscle, joint, or bone pain)
☐ Belly pain
☐ fever
☐ sever shivering
☐ Do not know

Source of treatment

○ private practioner
○ Self medication home remedies
○ sub center / primary healthcare centre (government)
○ private hospital
○ government hospital

Source of knowledge or awareness about mosquito borne diseases

☐ experience / observation
☐ school / college
☐ television
☐ Newspaper
☐ internet / social media
☐ government awareness campaingns

3.5.1.3.a Form Page

3.5.2 Analysis Engine:

3.5.2.1 Login Page:

The Data Analyst will log in here.

3.5.2.2 Analysis Page:

All the analysis and parameters will be displayed on this page. (Note this is a screenshot of our implemented analysis page and the proposed interface is similar to this page)

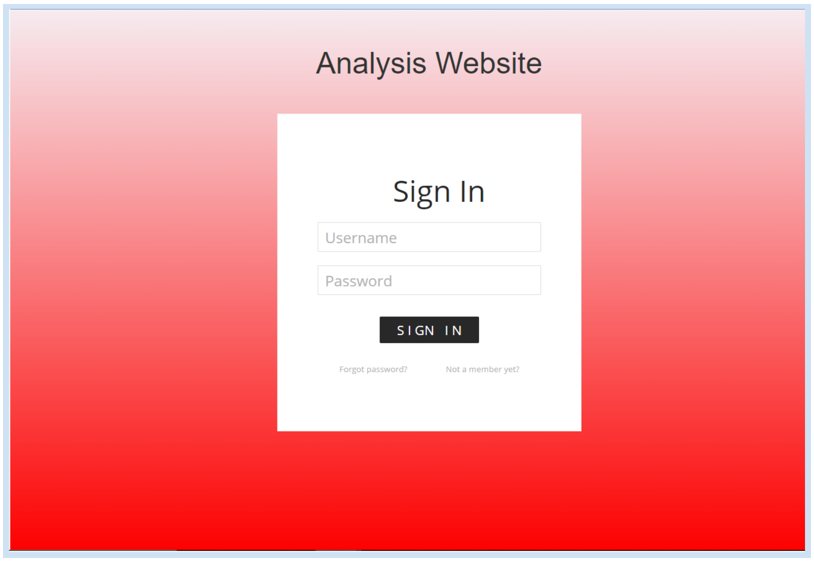
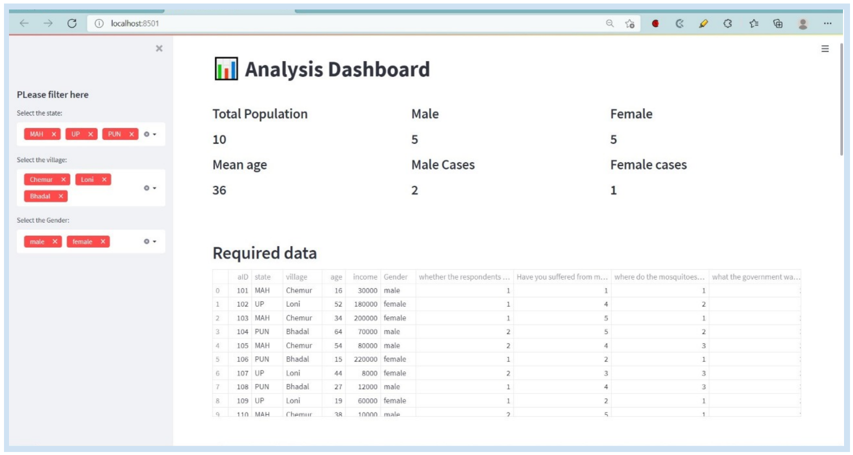


Fig 3.5.2.1.a Login Page



Analysis Dashboard

Total Population	Male	Female
10	5	5
Mean age	Male Cases	Female cases
36	2	1

Required data

	aID	state	village	age	income	Gender	whether the respondents ...	Have you suffered from m...	where do the mosquitoes...	what the government wa...
0	101	MAH	Chemur	16	30000	male	1	1	1	
1	102	UP	Loni	52	180000	female	1	4	2	
2	103	MAH	Chemur	34	200000	female	1	5	1	
3	104	PUN	Bhadal	64	70000	male	2	5	2	
4	105	MAH	Chemur	54	80000	male	2	4	3	
5	106	PUN	Bhadal	15	220000	female	1	2	1	
6	107	UP	Loni	44	8000	female	2	3	3	
7	108	PUN	Bhadal	27	12000	male	1	4	3	
8	109	UP	Loni	19	60000	female	1	2	1	
9	110	MAH	Chemur	38	10000	male	2	5	1	

Fig 3.5.2.2.a Analysis Page

4. Specific Requirements

4.1 Functional Requirements

4.1.1: NGO App:

Requirement	Priority
Input data through forms	High
Save to local database	High
Uploading to the main database	High
Authentication of staff using the app	Medium
Check connectivity	Medium

4.1.2: Analysis Website:

Requirement	Priority
Fetch database	High
Authentication	Medium
Perform numerical analysis	Medium
Perform graphical analysis	Medium

Functional Requirements Description:

The NGO staff inputs the responses accumulated by the villagers into the NGO application through forms. Authentication of the NGO staff is mandatory before accessing the application. The data filled via application is stored in the database. If Internet connectivity is available it will upload to the main database or else will remain stored offline. For accessing the website, again authentication is required. Analysts can get analysis from the website with a selection of adequate options. He/She can use numerical analysis to compare data and get basic statistical insights. Also, he/she can use graphical analysis to get visual insights into correlation and frequency distribution.

4.2 Non Functional requirements

Non Functional requirements(NGO app):

Features	**Measures**
Performance	1. Approximately 2000 entries per day. 2. 20 entries per second.
Size	1. Memory required less than 100 MB.
Ease of Use	1. Minimal training is required. 2. Multi-language support.
Reliability	1. The system will not lose any offline entry. 2. Data is not lost during transmission.
Security	1. Login required to use the app.

Non- functional requirements (Analysis website) :

Performance	1. Any given query will be fetched in under 10s. 2. Visualization/analysis within 10s.
Size	1. At max 300 MB RAM is required by the website.
Ease of use	1. Minimal training required(< 2 hours).
Reliability	1. Accurate query results.
Maintainability	1. The system uses MongoDB atlas for maintaining the database.
Security	1. Login required by the analyst. 2. Passwords are encrypted.

4.2.1 Performance Requirements:

The application can take 2000 entries per day, with 20 entries per second

at maximum. For the analysis website query will be fetched within 10 seconds.
This tells about the load the NGO application and website can take simultaneously.

4.2.2 Availability

The NGO application and the website both will be available at all times unless some exception of downtime of the server.

4.2.3 Reliability

NGO workers can access the application with the least to negligible crash or failure of the application. In case of failure, re-initialization of the program is recommended. The system will not lose any offline entry and will be guaranteed end-to-end transmission of data. Maximum accuracy in querying the database for the analysis website.

4.2.4 Supportability

The system uses Mongodb atlas for maintaining the database, and the Atlas server takes care of the site.

4.2.5 Portability

The NGO application will be compatible with the android operating system on tablet devices. The web application used for analysis is fully portable and any system using any browser should be able to use the feature of the application.

4.2.6 Security

Passwords will be encrypted in the database in order to ensure the user’s privacy. Login/Signup will be required for accessing the NGO application for NGO staff, and the website as well. The system will be protected against vulnerabilities.

4.2.7 Usability

Minimal training will be required and multi-language support will be provided on the NGO application.

4.2.8 Software System Attributes

1. NGO Application(python based)
2. MongoDB database
3. MongoDB atlas server
4. The database will be consistent and persistent at all times.