EGERTON UNIVERSITY

FACULTY OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

SOFTWARE DESIGN DOCUMENT FOR

IMMUNOTRAK: VACCINATION STATISTICS TRACKING

PRESENTED BY: **ERICK OGARO**
REG. NO.: **S13/09690/15**

PROJECT SUPERVISOR: **MR. G. NDIRANGU**
PROJECT COORDINATOR: **DR. ING WILFRED G. GIKARU**

DATE: 17TH MAY 2019

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PURPOSE AND SCOPE

This software design document describes the architecture and system design of the ImmunoTrak Web application. The application provides a reliable, convenient and efficient means of keeping track of vaccination statistics all over the country This design will detail the implementation of the requirements as defined in the Software Requirements Specification for the ImmunoTrak application. This document is an overview of the system dependencies and basic implementation of the system. This document can be used by users of the system, system administrators, reviewers and anyone interested in the functioning of the system. This document however is not an API guide or anything of the sort. There is no code discussed in this specific document.

1.2 PROJECT EXECUTIVE SUMMARY

1.2.1 SYSTEM OVERVIEW

The recent growth of mobile and internet technology past has led to the rampant development and usage of the internet to create highly scalable web applications that can handle large amounts of traffic at the same time. However, this technology has not been exploited for efficient administration purposes and involvement of information keeping and analysis, especially in the health-care industry. The ImmunoTrak application comes in to fill this gap and provides a platform where the immunization statistics for different locations in the country can be updated, visualized and statistically analyzed to provide perspective on the state of the country when it comes to child health care. This document describes the design of this proposed application.

The application can be simplified by splitting it into three modules. The charts module, the schedules module and admin module.

The admin module is where the application's core functionality lies. This module enables users of the application to create children, parents, new vaccines and also new users for the ImmunoTrak superusers. The service module allows the citizens/public to request for emergency services the.namely ambulance and firefighting equipment services offered by the county government. The aim to improve service delivery to the people. This module will have to work in real-time to make it efficient and reliable.

The Events and activities module provides a channel where the county government can communicate to the people. They will be able to update the people on county activities they are undertaking and events they are planning. The aim is to keep the citizens informed on county activities and projects.

The Polls module allows the county government to conduct opinion polls to collect the views and the stand of the people on important issues about the society. This makes the citizens to fully participate in the decision-making activities that affect their lives.

The system has two categories of users; The citizens who have the capability to create posts, comment, vote if there is an opinion poll and to view county activities and events and the admin of the system. The admin will be a county government representative who can respond to public posts. They also publish

information to keep the public aware of county activities. The system will support several admins but usually one of them will have elevated privileges to monitor the system for accountability purposes.

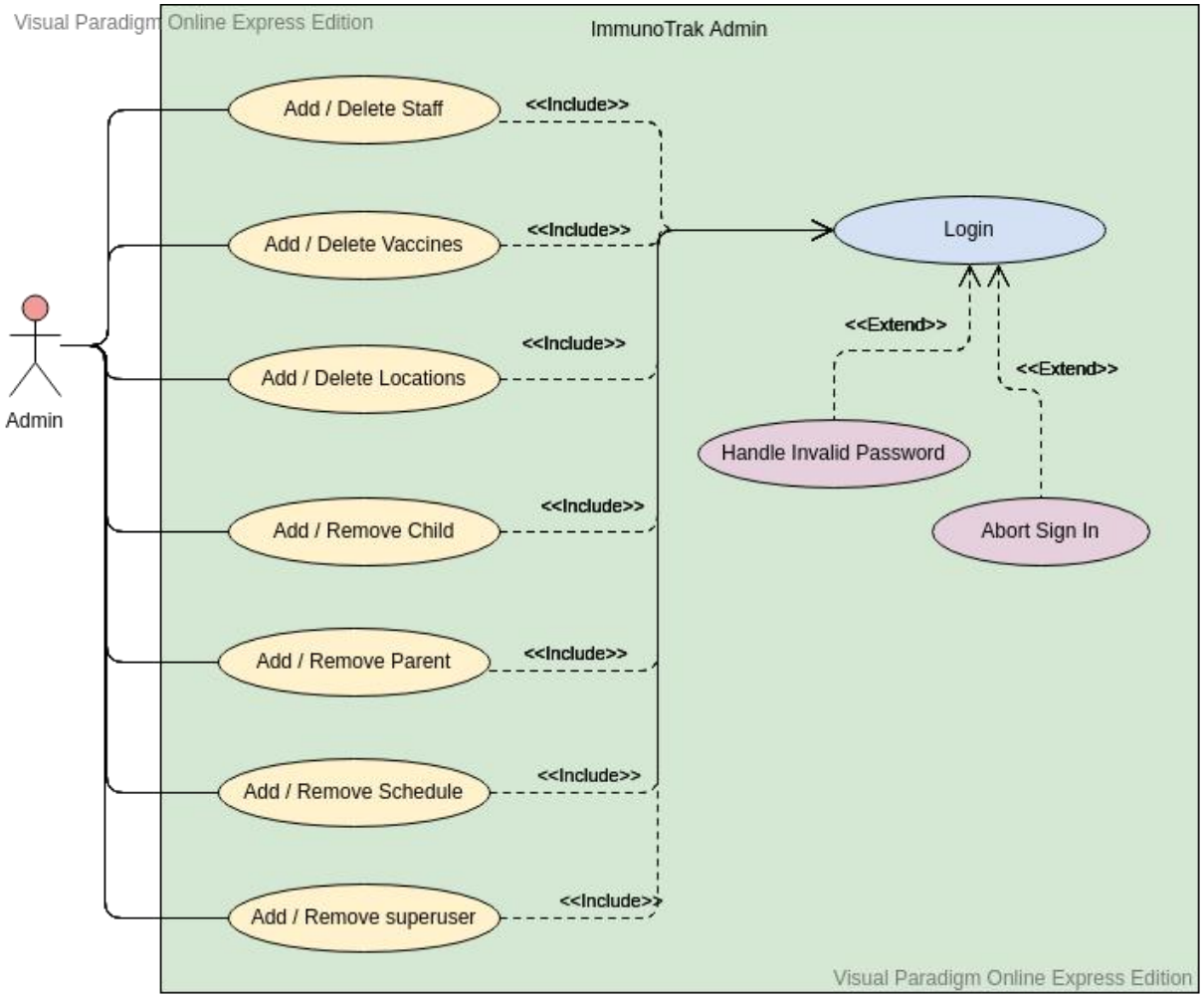

Fig. 1.0: ImmunoTrak Admin Use Case

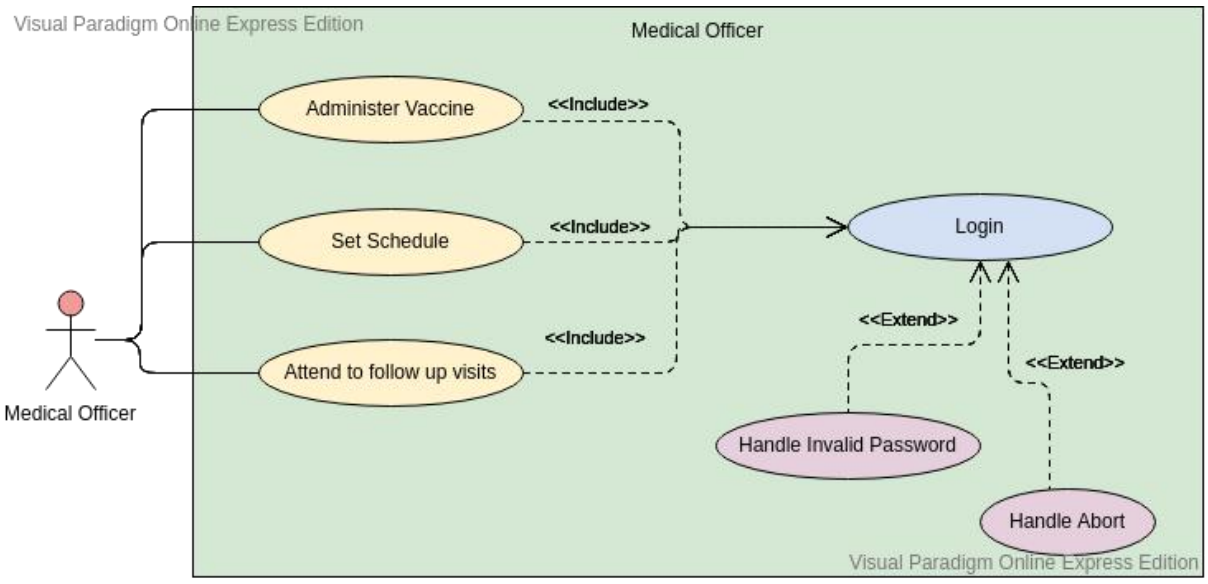

Fig. 1.1: *ImmunoTrak Medical Officer Use Case Diagram*

1.2.2 DESIGN CONSTRAINTS

The system is designed to operate in an environment as described in the Software Requirements Specification document for the ImmunoTrak application system. This will ensure that the complete functionality of the system are running as required.

The system will be built using Python programming language for the back-end. The Django framework will be used to scaffold the application and make sure it scales as expected and that the application is secure. JavaScript programming language will be used on front-end to ensure that the application is fully interactive. HTML and CSS will be used to display content and styling the web pages respectively.

The system is also designed to use MYSQL database which is free and open source for data storage

Being a web application, the entire system is online and requires internet connection in order to operate. All the users will be able to access the system through the specified URL where they can carry out their respective operations.

1.2.3 FUTURE CONTINGENCIES

Should the proposed system be adopted and used as expected, the application can be extended to desktop platform users by building native specific applications for each platform for efficient, reliable and smooth experience. This is in spite the fact that the current application will be accessible to all users through the web.

The system can be extended to work not only at the county level but also at the national level for effective administration and service delivery at all levels of administration.

The functionality of the system can also be extended to include more services that can be accessed via the digital platform to improve overall service delivery by the government.

2.0 SYSTEM ARCHITECTURE

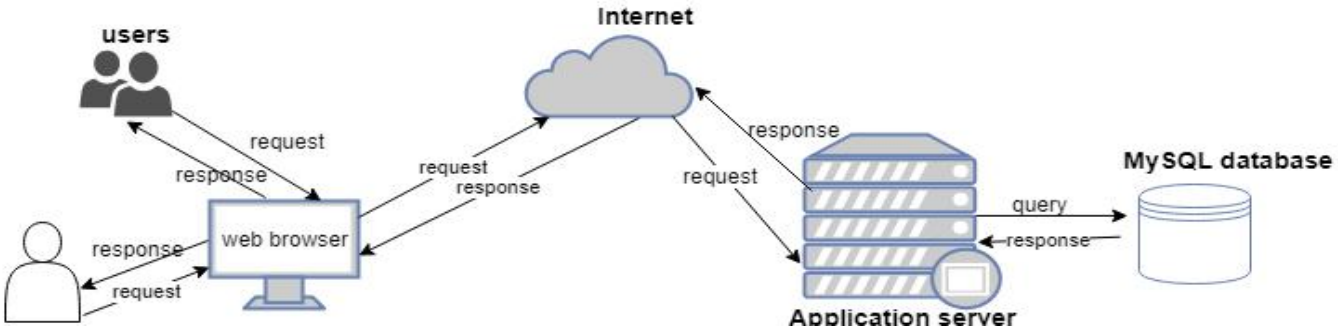

Fig. 2.0: *System architecture for ImmunoTrak application*

2.1 SYSTEM HARDWARE ARCHITECTURE

The system is housed in a server which comprises of a central processing unit, main memory and secondary memory as its main components. Each of these components has a specification that will suit the needs of the system. These specifications have been defined in the system requirements document for the ImmunoTrak Application. The server communicates with a MySQL database server that is used to store the information of the system.

The server is connected to the internet where the system can be interacted with through a user interface that can be accessed from a web browser by entering the respective URL.

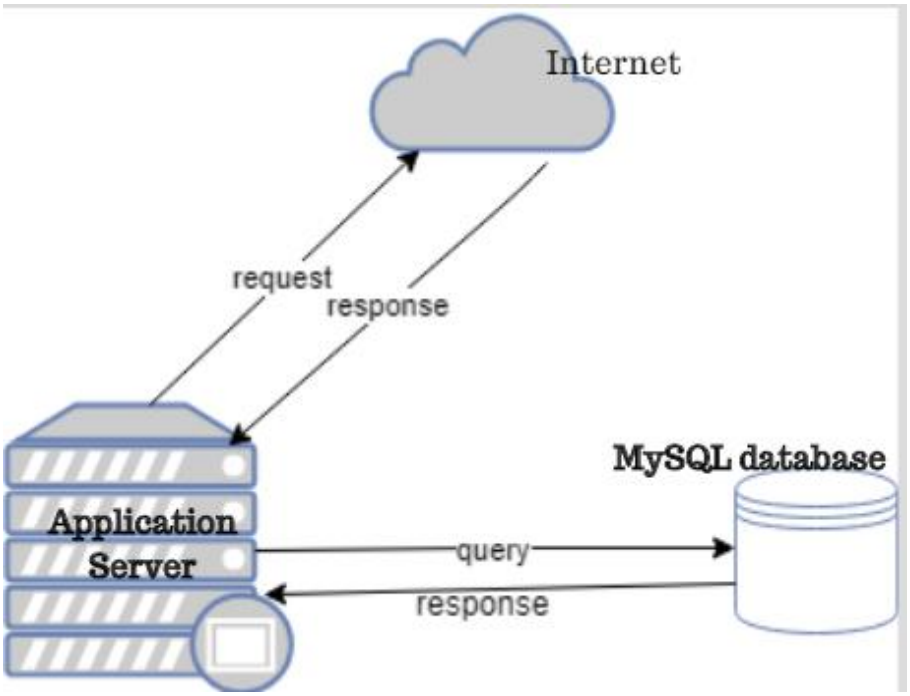

Fig. 2.1: *Hardware architecture for the ImmunoTrak application*

2.2 SYSTEM SOFTWARE ARCHITECTURE

The ImmunoTrak Application is divided into two major components: a client-side application and a server-side application which interacts with MySQL database.

The client-side application runs on the user's browser and is responsible for presenting user interface that the user interacts with. The client application also consists of a functional component that fetches data from the back-end server to client application. Client-side validation logic will also be implemented in this part of the application. The server-side component is responsible for processing user requests and generating appropriate responses. It also interacts with the database server for data persistence.

The following are the software components that will be required during development:

i. Visual Studio Code – This will be the text editor that will be used to develop the system.

ii. Web browser, for example Mozilla Firefox, Chrome or any other according to clients' preferences.

iii. Apache2 server – It is an application server from Apache Software Foundation that executes JavaScript scripts and renders web pages upon users' request.

iv. MySQL server – It is a DBMS (Database Management Software) that is used to manage the database files involved in the systems allowing data storage and access.

Below are activity diagrams that only shows the top-level communication between the system and external entities. It demonstrates how the ImmunoTrak application interacts with the system users.

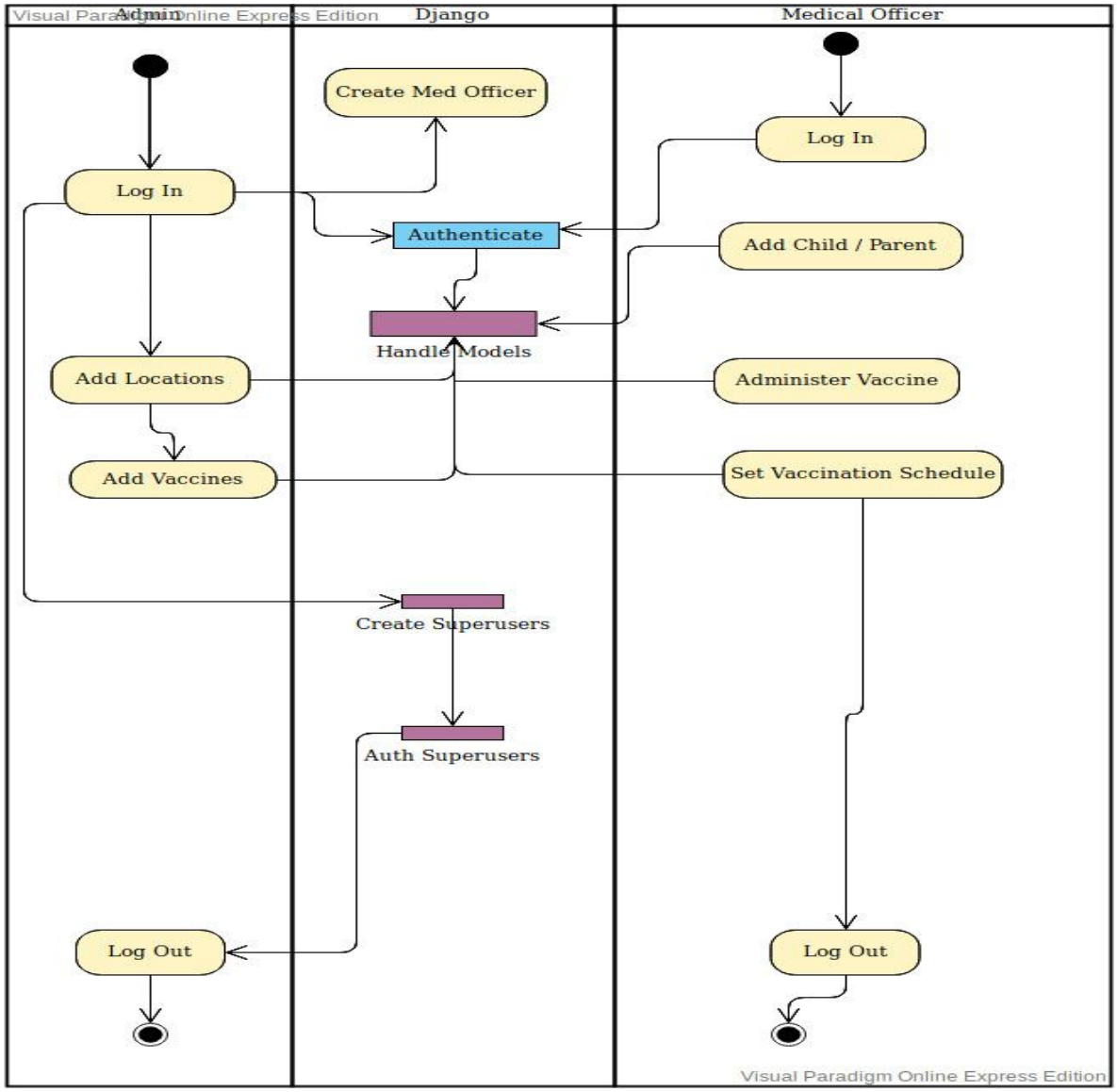

Fig. 2.1: *Admin and Medical Officer / Nurse activity diagram*

2.3 INTERNAL COMMUNICATIONS ARCHITECTURE

The communication process of the system will be achieved only in internet availability that will enable communication between the application and the database during retrieval of data from database and communication between various modules of the system. A web browser is used as the interface to make the request to the web server and also renders the response back to the user of the requested resource. The protocol used to ensure proper communication between the browser and server hosting the system is HTTP.

3.0 FILE AND DATABASE DESIGN

3.1 DATABASE MANAGEMENT SYSTEM FILES

The database, stored in the database server and managed through the application server, will be used to store data for the entire application. Below is a description of the database tables for the application.

The figure below shows the description of the tables in the ImmunoTrak Project:

```
Database changed
mysql> show tables;
+----------------------------+
| Tables_in_immunotrak       |
+----------------------------+
| auth_group                 |
| auth_group_permissions     |
| auth_permission            |
| auth_user                  |
| auth_user_groups           |
| auth_user_user_permissions |
| django_admin_log           |
| django_content_type        |
| django_migrations          |
| django_session             |
| trak_child                 |
| trak_location              |
| trak_parent                |
| trak_vaccine               |
+----------------------------+
14 rows in set (0.00 sec)

mysql>
```

Children, Parents and Vaccines Table descriptions.

```
mysql> desc trak_child;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| id          | int(11)     | NO   | PRI | NULL    | auto_increment |
| first_name  | varchar(50) | NO   |     | NULL    |                |
| middle_name | varchar(50) | NO   |     | NULL    |                |
| last_name   | varchar(50) | NO   |     | NULL    |                |
| dob         | date        | NO   |     | NULL    |                |
| child_id    | int(11)     | NO   |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
6 rows in set (0.01 sec)

mysql> desc trak_parent;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| id          | int(11)       | NO   | PRI | NULL    | auto_increment |
| first_name  | varchar(50)   | NO   |     | NULL    |                |
| middle_name | varchar(50)   | NO   |     | NULL    |                |
| last_name   | varchar(50)   | NO   |     | NULL    |                |
| parent_id   | decimal(50,0) | NO   |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> desc trak_vaccine;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| id          | int(11)     | NO   | PRI | NULL    | auto_increment |
| name        | varchar(50) | NO   |     | NULL    |                |
| description | longtext    | NO   |     | NULL    |                |
| quantity    | int(11)     | NO   |     | NULL    |                |
| available   | tinyint(1)  | NO   |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Location and Users table descriptions.

```
og@roh: ~
| available   | tinyint(1)  | NO   |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> mysql> desc trak_location;
ERROR 1146 (42S02): Table 'immunotrak.trak_v' doesn't exist
mysql> desc trak_location;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| id           | int(11)     | NO   | PRI | NULL    | auto_increment |
| county       | varchar(50) | NO   |     | NULL    |                |
| vaccinations | int(11)     | NO   |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> desc auth_user;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int(11)      | NO   | PRI | NULL    | auto_increment |
| password     | varchar(128) | NO   |     | NULL    |                |
| last_login   | datetime(6)  | YES  |     | NULL    |                |
| is_superuser | tinyint(1)   | NO   |     | NULL    |                |
| username     | varchar(150) | NO   | UNI | NULL    |                |
| first_name   | varchar(30)  | NO   |     | NULL    |                |
| last_name    | varchar(150) | NO   |     | NULL    |                |
| email        | varchar(254) | NO   |     | NULL    |                |
| is_staff     | tinyint(1)   | NO   |     | NULL    |                |
| is_active    | tinyint(1)   | NO   |     | NULL    |                |
| date_joined  | datetime(6)  | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
11 rows in set (0.03 sec)

mysql>
```

3.2 NON-DATABASE MANAGEMENT SYSTEM FILES

The ImmunoTrak Application will also make use of the file system for storage. This will be used for storage of image files uploaded by users. System logs are also stored in text file format. All these files are stored in the secure web server in a given specified directory.

4.0 HUMAN-COMPUTER INTERFACE

The system can be generalized to have two sections from a user interface perspective. This is based on the two kinds of users: the public/citizens and the admins. Each users has a specific user interfaces that they are restricted to and are redirected to their specific pages upon successful registration. The system comes pre-bundled with one admin account and this admin has the highest privileges in the system. He can create other admin accounts and assign the specific roles. These accounts have lower privileges and can be monitored by the main admin. All user's information is stored in the database and the passwords are encrypted.

4.1 INPUTS

The ImmunoTrak Application inputs all data through data entry by mostly text fields, drop down lists and check boxes. All the necessary data is recorded and securely stored in the database from which can be retrieved and used throughout the system. Before data is recorded to the database, they undergo some form of verification in order to ensure that the data stored in the database is consistent, accurate and reliable. Each data entry field has its own acceptable value such as numerical value, text values etc. For instance, in the sign up page for the users, all fields are mandatory and the user cannot proceed without filling up all the fields. Email fields in this form must be valid email values of the format XXXX@XXX.XX The email must also be unique and not currently existing in the system. Failure to comply with these input standards, the system will prompt the user to correct the errors before the system can complete the registration process. Upon successful registration, the system notifies the user that the process was successful. Sensitive information such as the passwords are encrypted before they are stored in the database to prevent them from leaking out in the case that some unauthorized individual gains access to the database. All users gain access to the system through the login form by entering the correct credentials which the system verifies that the user has the valid information to access the system. If not, an error message will be displayed indicating that incorrect credential have been entered.

4.2 OUTPUTS

The system's main interface is a dashboard for all users. Each of the different user types have a different dashboard displaying the most crucial information that the user may be looking for and have access to. Each of the user types can only view the information that they are limited to.

Apart from these immediate data, the system will display information depending on the user's need.

5.0 DETAILED DESIGN

5.1 HARDWARE DETAILED DESIGN

The required hardware components in order to run the system include a web server with the requirements as specified in the system requirements document for ImmunoTrak Application. The web server should also have the recommended database management system installed. The database management system can also run on a database server running on a different machine. Other hardware components include the network switch, routers and a client PC or digital devices that have web browsers installed in them.

5.2 INTERNAL COMMUNICATIONS DETAILED DESIGN

The internal flow of information between the system modules is necessary in order for the system to fully perform its functionality. This is facilitated by the communication between class objects under different modules through the methods calls. Also, to prevent tight coupling between the modules, each module operates entirely independently and any change of the state of the system is update through the database from which other modules can get the updated from. The database therefore acts as a single source of truth.

The client-side application communicates with the application server through HTTP protocol. The application server in turn communicates with the database server through the database driver. Data is captured through the various user interfaces and persisted to the database. The user interfaces also display data from the database.

6.0 EXTERNAL INTERFACES

6.1 INTERFACE ARCHITECTURE

External interfacing helps facilitate communication between the system and other external systems. Since the communication is via internet, TCP/IP protocol will be used. To ensure proper external interfacing, an active link of communication between the client, server and

database PCs components and any other external systems that the application will be exchanging information with is very vital. The network architecture is not much of a consideration.

6.2 INTERFACE DETAILED DESIGN

Data does not need to be reformatted before transmission or reception over the interface between two different communicating modules since communication is through TCP/IP. The protocol regulates data exchange over the interface and also dynamically handles errors arising from the data exchange. A brief error description will also be provided to the user in HTML format.

7.0 SYSTEM INTEGRITY CONTROLS

To ensure a strong system base security, the proposed system will implement data encryption before any piece of critical and confidential information or data is stored in the database. Successful protection of data will ultimately ensure integrity of the system. Thus integrity checks need to be employed in order to ensure correctness of stored data and accountability for every system user. The following security controls will be implemented to ensure integrity of the system:

a) **Authorization control**. This ensures that users can only access data that they have the required permissions.

b) **Integrity control**. This ensures that data stored in the database cannot be violated or altered by any unauthorized parties in any way whatsoever.

c) **Authenticity control**. This aspect of security will verify user credentials at the time of login into the system. This ensures that only authorized users are allowed to access the system.

d) **Confidentiality control**. This is a security technique that prohibits unauthorized access of personal information.

e) **Client and server-side validation**. This will ensure that no wrong inputs are accepted into the system.

The system being web application is open to a variety of attacks. However, security measures have been placed to prevent attacks such as Cross Site Scripting attacks and Cross Site Request Forgery attacks (CSRF) e.g. All data submitted to the server must include CSRF tokens which is verified by the server to prevent this kind of attacks.