|  |  |  |
| --- | --- | --- |
|  | MANIPAL INSTITUTE OF TECHNOLOGY  (A constituent Institute of MANIPAL UNIVERSITY)  MANIPAL - 576 104, KARNATAKA, INDIA |  |

Industrial Training

on

Web Application Development using MongoDB

SUBMITTED

BY

Srivatsava Gummalla

Registration no.: 140905292

srivatsavagummalla@gmail.com

Phone: +91-9591311992

Under the Guidance of:

Pandarinath Siddineni

TABLE OF CONTENTS

[1 PROBLEM DEFINITION 6](#_Toc488333474)

[1.1 Introduction 6](#_Toc488333475)

[1.2 Requirements 6](#_Toc488333476)

[1.3 FIND A DOCTOR WEB PORTAL 6](#_Toc488333477)

[1.3.1 NPPES NPI Database 6](#_Toc488333478)

[1.4 Disease code mapping (ICD9 to ICD10) 6](#_Toc488333479)

[1.4.1 ICD Codes 6](#_Toc488333480)

[1.4.2 ICD9 Vs ICD10 6](#_Toc488333481)

[2 STUDY — FULL STACK WEB DEVELOPMENT PLATFORMS 8](#_Toc488333482)

[2.1 Stages of full stack: 8](#_Toc488333483)

[2.1.1 Database: 8](#_Toc488333484)

[2.1.2 server-Side code: 8](#_Toc488333485)

[2.1.3 Client-Side code: 9](#_Toc488333486)

[2.1.4 Middleware: 9](#_Toc488333487)

[2.2 MEAN Stack: 9](#_Toc488333488)

[3 Software Design and Development 10](#_Toc488333489)

[3.1 SOFTWARE DEVELOPMENT LIFE CYCLE 10](#_Toc488333490)

[3.2 Development Platform 11](#_Toc488333491)

[3.2.1 Programming Language— Python 3.6.0 11](#_Toc488333492)

[3.2.2 MVC Framework for Server and Client— Bottle 0.12.13 [3] 11](#_Toc488333493)

[3.2.3 Database— MongoDB 3.4.0 [4] 12](#_Toc488333494)

[3.2.4 Client Side— HTML, CSS, JavaScript, jQuery 12](#_Toc488333495)

[3.3 MVC pattern— Bottle 12](#_Toc488333496)

[3.4 NoSQL Data Bases: 13](#_Toc488333497)

[3.5 Mongo Database 13](#_Toc488333498)

[3.5.1 Physician Collection (BSON [6] format) 14](#_Toc488333499)

[4 DEVELOPMENT METHODOLOGY AND TOOLS 16](#_Toc488333500)

[5 RESULTS AND TESTING 18](#_Toc488333501)

[6 CONCLUSION 22](#_Toc488333502)

[7 REFERENCES 23](#_Toc488333503)

[8 APPENDIX-A 24](#_Toc488333504)

[9 APPENDIX-B 25](#_Toc488333505)

***ABSTRACT***

*This report describes the work conducted during a two-month internship at the company Informatics In Context. This internship is part of the industrial training before the 7th semester in Computer Science and Engineering.*

*The company develops healthcare software. The current project is a “Find A Doctor” web application. This software has been developed using the full stack programming approach developed using the agile model of software development. This application was designed to locate physicians at a given location with some required specialty.*

*A “Find A Doctor” application has been developed to find the nearest physician from a given location in U.S. This application can also find a physician based on their specialty and their*

*A MATLAB LCA tool has been developed which can perform LCA analysis and Monte Carlo simulations. The tool is compatible with the ecoinvent 2.2 database and can import several impact assessment methods. Performance of the tool is more than 100 times faster than SimaPro. Because of its efficiency, the tool can quickly calculate the coefficient of variation per impact category for all the products in an ecoinvent 2.2 database. It would take about 2-3 days calculation time, whereas it would take about one year to calculate the same results in SimaPro. The coefficient of variation could be useful to incorporate uncertainty in streamlined life cycle assessment tools. How this could be useful presented to the user needs to be further investigated.*

# PROBLEM DEFINITION

## Introduction

In the healthcare department, the ICD codes are provided for the diseases and NPI codes are given for each physician. Based on the disease code and the location of the patient the nearby physician with a relevant specialization needs to be suggested. The ICD codes have two versions which are the ICD-9 and the ICD-10. The presently used codes are the ICD-10 but the older ICD-9 codes are still used at some places. A lookup system for the relevant ICD10 code for an ICD9 code is needed.

## Requirements

The project contains two main requirements, first being “developing find a doctor portal for patients and hospitals” and the second is to develop a web portal for ICD9 to ICD10 mapping.

## FIND A DOCTOR WEB PORTAL

Fill the details here:

Why end user need such portal?

Geography: US This application is intended for US customers

### NPPES NPI Database

## Disease code mapping (ICD9 to ICD10)

Fill the details here:

What is ICD codes?

Why ICD-9 to ICD-10 mapping is required?

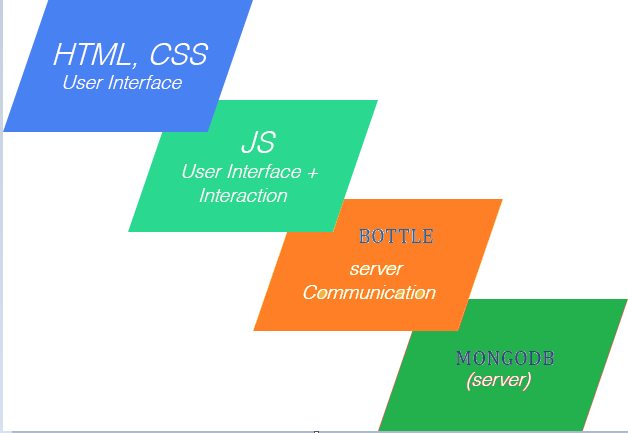
Geography: US This application is intended for US customers

### ICD Codes

### ICD9 Vs ICD10

# STUDY — FULL STACK WEB DEVELOPMENT PLATFORMS

Full stack means a collection of software used in developing a complete web application. It can be broadly classified into front-end and back-end development. The back-end development consists of a server, a database and an application to communicate with the database. The front-end development consists of web pages and client-side code which the user can directly interact with to use the application. There are various kinds of stacks available based on the programming languages that are chosen at the different stages in development. Some of the most common examples of full stack are LAMP, MEAN etc.



## Stages of full stack

### Database

Database is used to store all the data related to the service being provided by the application. The data stored in a database needs to be available at all times and it should be able to maintain the consistency of files. In the current project, a NoSQL database was chosen as it emphasizes on horizontal scalability. The database used is MongoDB. This database is a part of MEAN stack. It is easy to use as less restrictions are provided on data storage. The mongod server is used to provide access to the mongo database.

### server-Side code

The server-side code is the code used to build the backend software of the website. These codes, which are also called scripts, are designed to run on a server and interact with permanent storages like databases. It facilitates the transfer of data and also powers functions in dynamic web applications. It is also used to build the application programming interfaces(APIs). Some of the server-side scripts are PHP, Ruby, Python etc. In this project, the Python’s Bottle MVC framework has been used to develop the server-side scripts.

### Client-Side code:

This code is the code which is used to develop the webpages and is part of the front-end development. It is used to generate the web pages and provide user interaction and interface. The HTML, CSS languages are used to provide the interface and develop static webpages. The JavaScript is used to provide interaction for the user and make the web page dynamic.

### Middleware:

Middleware is computer software that connects software components or applications. It is present along with the server-side scripts in a web application and provides enhanced performance. The software consists of a set of services that allows multiple processes running on one or more machines to interact. The common middleware services are CORBA, DCE etc. In the current project, no middleware has been used.

## MEAN Stack:

One of the full stack technologies which is currently gaining popularity is MEAN stack. This stack uses MongoDB, Express.js, Angular.js and Node.js.



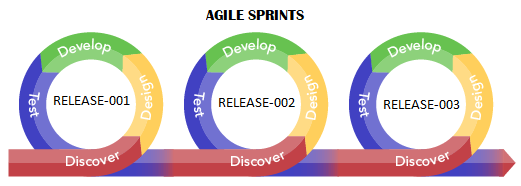
In MEAN stack the MongoDB is a NoSQL database and uses JSON style documents for data representation. The Express.js provides the server framework for web applications. Angular.js is a front-end java script framework to develop the client-side architecture. Node.js is a concurrent JavaScript environment for building scalable and fast web applications. The benefits of using MEAN stack is that the entire code can be written using Java Script. It supports the Model View Controller(MVC) framework. The other advantage is the use of JSON to transfer the data and the huge module library of Node.js.

# Software Design and Development

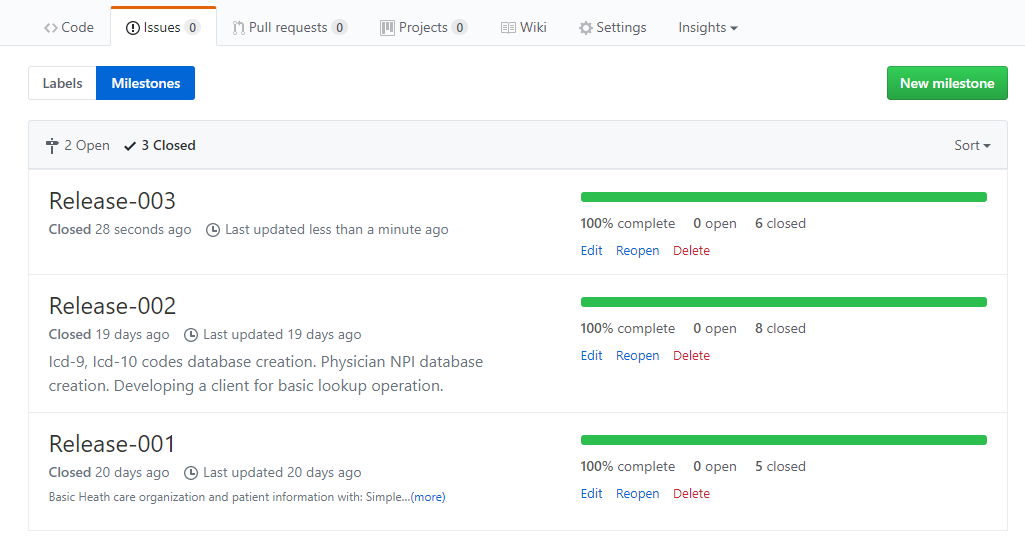
## SOFTWARE DEVELOPMENT LIFE CYCLE

Software Development Life Cycle (SDLC) defines the software development methodology. There are many SDLC models prevalent in the industry. Agile is on of such new SDLC model that is highly popular in startups.

In the agile model, the product is developed in small incremental builds. These builds are called sprints and at the end of each sprint a working product is developed. A major task is broken down into many minor tasks and these tasks are completed iteratively. This model focuses on adaptability and customer satisfaction by rapid delivery of working software product.



This internship project is developed using 5 sprints as shown in the diagram below. GitHub Issue tracker was used to capture each sprint and its tasks. At the time of this documentation four sprints were completed and the project was in final sprint. Each sprint is termed as Release-00X.



## Development Platform

Software development is done in Windows-10 PC with the following software and tools:

### Programming Language— Python 3.6.0

Python is an object-oriented programming language. It offers high readability and efficiency for web development. Python also has a lot of support from the community which was helpful for quickly solving the issues encountered when programming.

Python has many supporting packages, that helps in building large software programs in short span of time. I have used following python packages for the project:

#### Pip 9.0.1

This is a python utility for software installation and updates under python development environment.

#### PyMongo 3.4.0 [1]

PyMongo is a python distribution containing tools for working with MongoDB. This is the lowest level of interfaces to make connection and interact with MongoDB. For interface details refer PyMongo website[1].

#### Mongoengine 0.13.0 [2]

Mongoengine is a document object mapper for working with MongoDB. This software is used in establishing connection to the database and to write queries to the database.

For interface details please refer the mongoengine website[2].

### MVC Framework for Server and Client— Bottle 0.12.13 [3]

Bottle is a fast, simple and lightweight WSGI micro web-framework for Python. Bottle version 0.12.13 has been used for the server-side scripting in the project.

Bottle has a built in HTTP development server and has convenient access to form data, file uploads, cookies, headers and other HTTP-related metadata. It also provides easy to use templates for creating webpage design and URL mapping and routing.

### Database— MongoDB 3.4.0 [4]

MongoDB is a NoSQL and document oriented database. Data is stored in the form of JSON documents in the database. The MongoDB version 3.4.0 has been used in this project. It provides the database for storing documents and a server for accessing the database.

### Client Side— HTML, CSS, JavaScript, jQuery

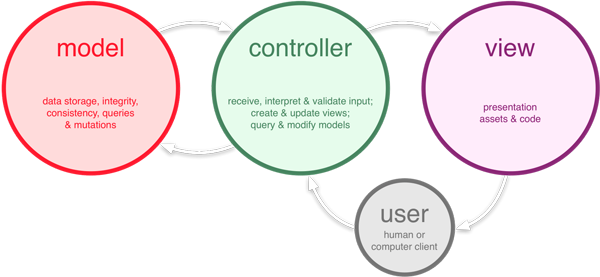
The client-side code to develop the front-end has been written using HTML and CSS as the base.

For static page were developed mostly using HTML and CSS, while JavaScript and jQuery was used for dynamic pages.

JavaScript is a high-level, dynamic, weakly typed, object-based, multi-paradigm, and interpreted client-side programming language. This is used alongside HTML and CSS in creating dynamic and interactive webpages. The jQuery using Ajax has also been implemented in this project.

## MVC pattern— Bottle

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. This pattern has been used for the development of the product.



The Model component corresponds to all the data-related logic that the user works with. In this project, all the collections present in the database have been created as models using MongoEngine and have been titled starting with the word “model”. These models are used when the data is being imported to the database.

The View component is used for all the user interface logic of the application. It is these views that are visible to the user. The template files of bottle have been created as views and have been titled starting with the word view. These views are written in HTML, CSS and JS. Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

The controller files in the project have been written using python’s bottle framework. These files have been named starting with the word “cnt”.

## NoSQL Data Bases:

NoSQL database provides a mechanism for the storage and retrieval of data that is modeled in means other than the tabular relations.

Some NoSQL Database Types are given below:

* **Document Databases** pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.
* **Graph stores** are used to store information about networks of data, such as social connections. Graph stores include Neo4J and Giraph.
* **Key-Value stores** are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or 'key'), together with its value. Examples of key-value stores are Riak and Berkeley DB. Some key-value stores, such as Redis, allow each value to have a type, such as 'integer', which adds functionality.
* **Wide-Column** stores such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows.

When compared to relational databases, NoSQL databases are more scalable and provide superior performance, and their data model addresses several issues that the relational model is not designed to address:

* Large volumes of rapidly changing structured, semi-structured, and unstructured data
* Agile sprints, quick schema iteration, and frequent code pushes
* Object-oriented programming that is easy to use and flexible
* Geographically distributed scale-out architecture instead of expensive, monolithic architecture

## Mongo Database

MongoDB is a document database with the scalability and flexibility that is required for the querying and indexing. MongoDB stores data in flexible, JSON like documents. JSON means Java Script Object Notation. It is a lightweight data-interchange format that is easy for humans to read and write. JSON has two structures which are a collection of name/value pairs and an ordered list of values. As MongoDB uses JSON objects, the data stored is easily readable and is represented in key/value pairs. MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use. In MongoDB, index can be made on any attribute and it increases the speed of queries using that attribute.

In a single MongoDB server, there can be multiple databases. A database is a physical container for collections, with its own set of files in the file system. A collection is a group of MongoDB documents. Documents within a collection can have different fields as it does not enforce any schema. All documents of a related purpose are generally stored in a single collection. In this project, the database was named as ClinicalData. In it collections for the physician data (NPI data), ICD-9 codes, ICD-10 codes and patient data (randomly generated names and information) were made. Data import files were written with command line parameters to import these collections to the database. The data models for the collections were written in MongoEngine and have been used to form the key/value pairs and insert each document to database. MongoDB is horizontally scalable and the capacity can be increased by connecting multiple hardware or software entities so that they work as a single logical unit. Because of this MongoDB is suitable for handling large amounts of data spread across multiple servers. The physician collection in the ClinicalData database consists of more than 4.8 million records. This collection can be handled with ease when using the MongoDB.

This database is used in storing the NPI physician data and the ICD codes. These data are first imported to the database and then is made available through the server. As the Mongo database stores data in JSON format dealing with these objects is easier when using JavaScript.

Few database models and collections taken from the project:

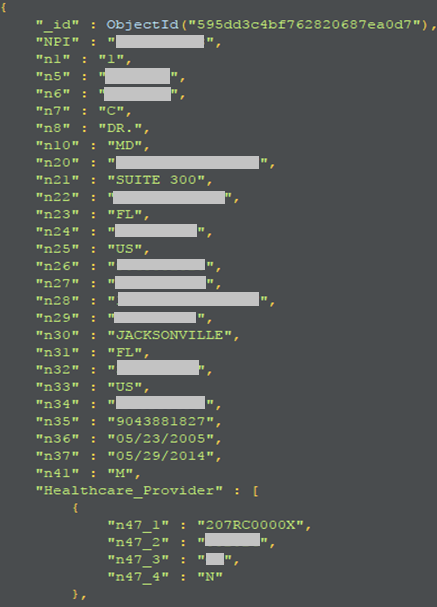
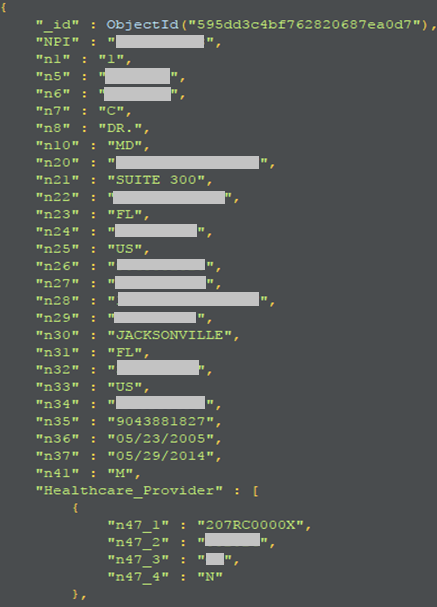
### Physician Collection (BSON [6] format)

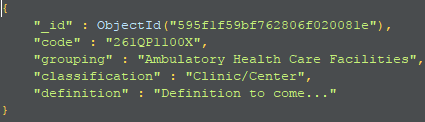
Physician collection is generated from the NPI [5] data available from Center for Medicare and Medicaid Services. The data model class for the physician is shown below:

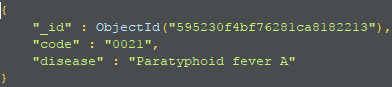


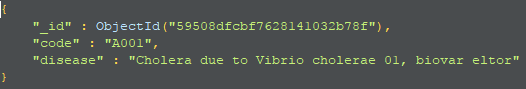


MongoDB saves its data into BSON format. The equivalent of the data model in the BSON format is:



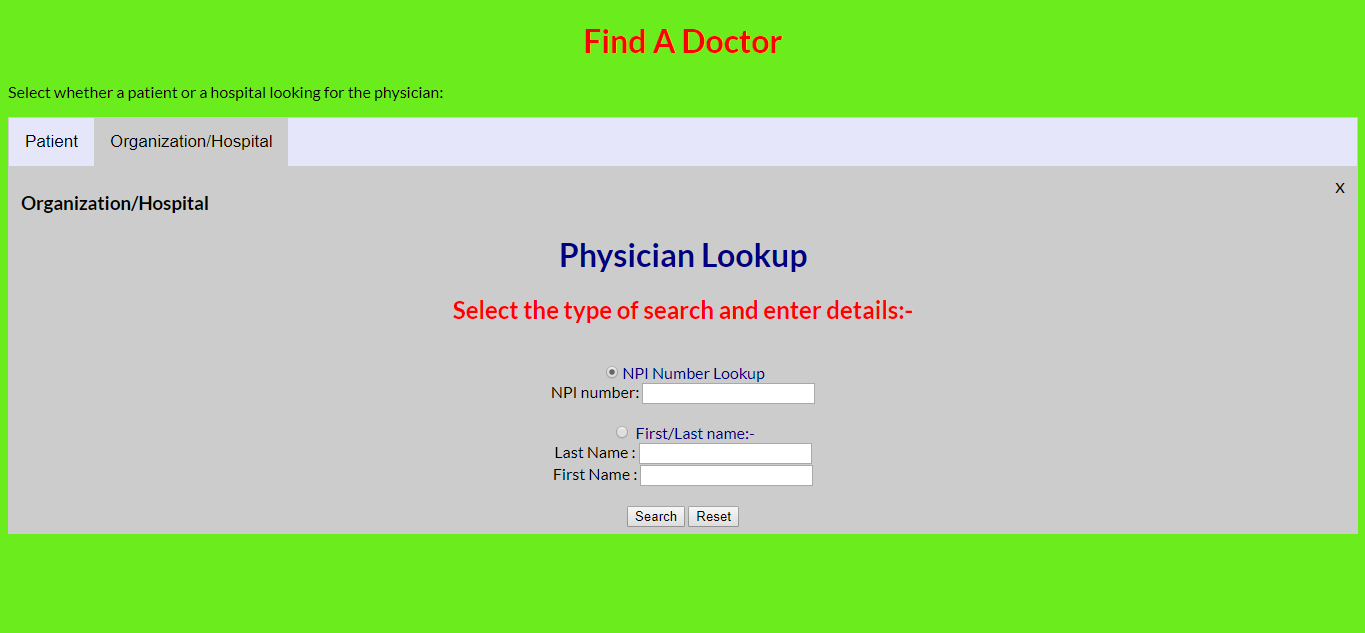




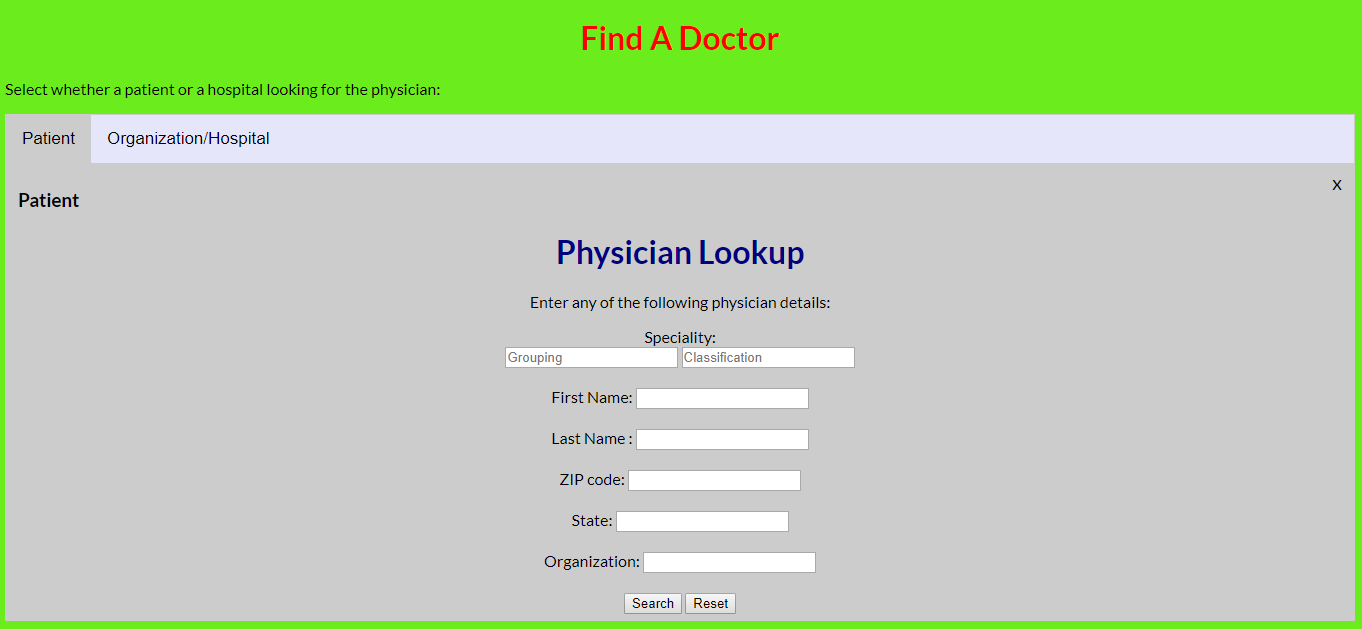


# RESULTS AND TESTING

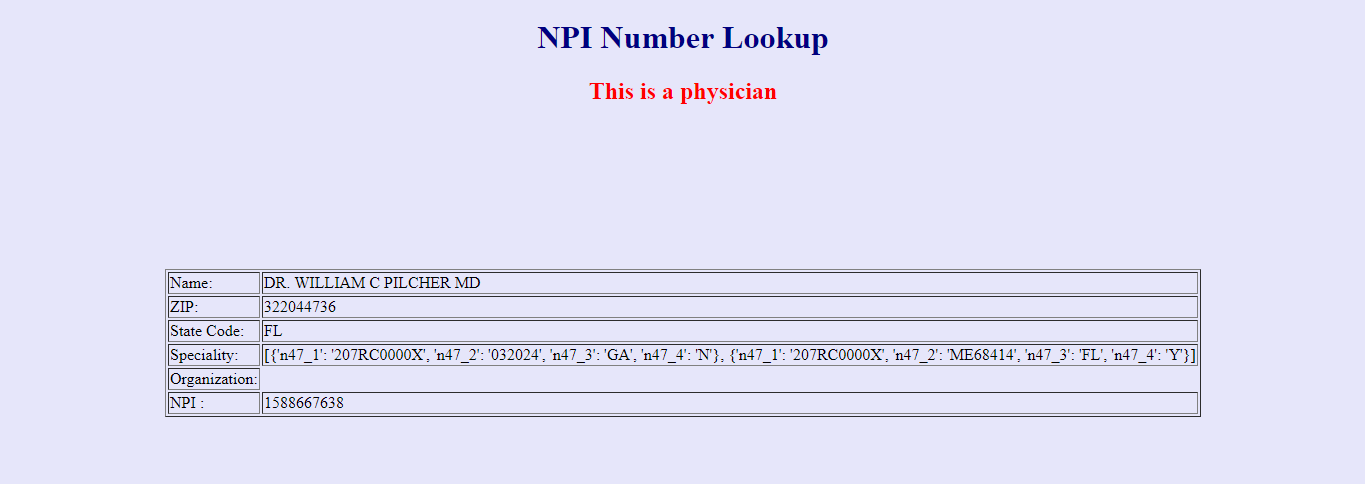
1.Physician Search for organizations based on the NPI number or the first/last name



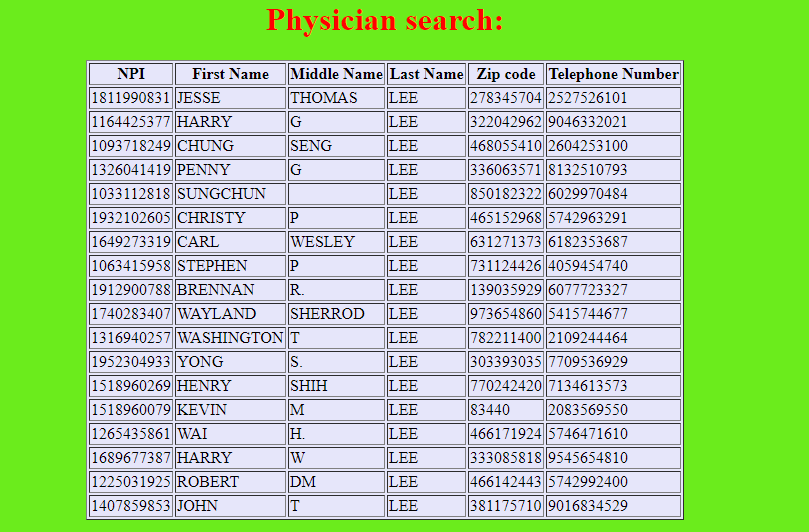
2.Physician Search for individuals based on specialty and location



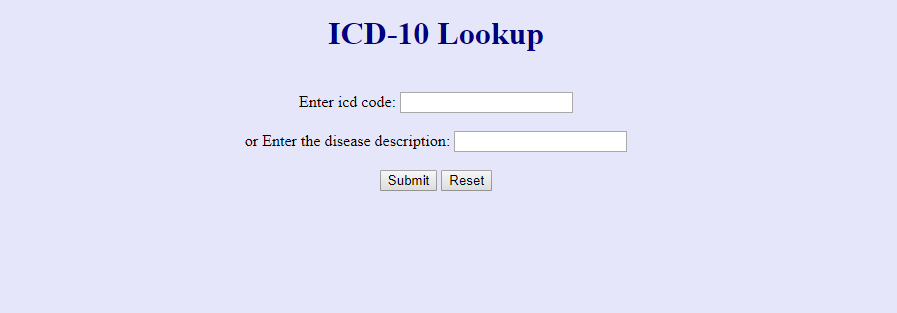
3. Display of physician details based on NPI number

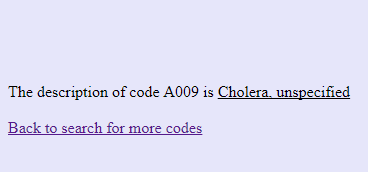


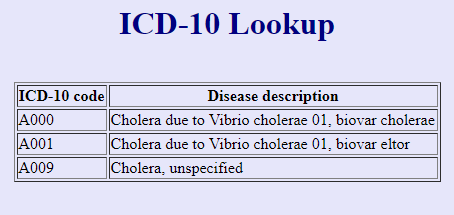
4.Physician search given first name or last name



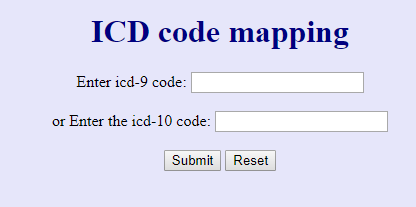
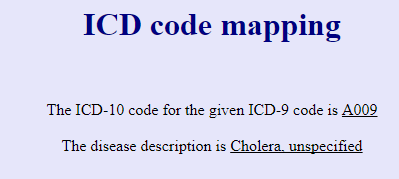
5.ICD code lookup:







6.ICD9 to ICD10 mapping:



# CONCLUSION

Healthcare is an important service in the modern human society. To bring the services of the physicians to the patient is the purpose of this project.

This project has given exposure to client side as well as server-side development for web application. This project can be extended further with following development activity:

I. Development of Mobile application that interacts with the server for “FIND A DOCTOR”

II. Deploy the server in some cloud service.

# REFERENCES

1. Mongoengine, <http://mongoengine.org/>
2. PyMongo - MongoDB API, <https://api.mongodb.com/python/current/>
3. Bottle: Python Web Framework — Bottle 0.13-dev documentation, <https://bottlepy.org/>
4. MongoDB, <https://www.mongodb.com/>
5. Data Dissemination - Centers for Medicare & Medicaid Services, <https://www.cms.gov/Regulations-and-Guidance/Administrative-Simplification/NationalProvIdentStand/DataDissemination.html>
6. BSON data format, <https://en.wikipedia.org/wiki/BSON>
7. JSON data format, <http://www.json.org/>
8. Agile Development, <https://en.wikipedia.org/wiki/Agile_software_development>

# APPENDIX-A

**Environment Setup Instructions**

**Python**

1. Download the executable for python 3 and run the executable.
2. On the first screen, enable the “Add Python 3.6 to PATH” option and then click “Install Now.”
3. Click “Close” to finish the installation.

**MongoDB**

1. Download the msi package for windows from this link <https://www.mongodb.com/download-center#community>
2. Follow the instructions and install the database.
3. Open up a command prompt, then navigate to `bin` in the mongo folder [Navigating to mongo's bin folder]
4. Type `mongo.exe` (which is the command used to start mongo Db Power shell). Then see the below response [Executing mongo.exe][5] > Couldn’t connect to server.
5. We have to start the mongo db by using the command `mongod`. Execute this from the `bin` folder of mongo.
6. We have to create the `data\db` folders in the `C` drive of our BOX in which we are installing mongo. Create the folder structure in C drive
7. Next, we can go and start the db using the command `start mongo.exe`
8. Open a new cmd window.
9. Navigate to Bin
10. In the command prompt, execute the command mongo or mongo.exe
11. Refer to this link <https://docs.mongodb.com/getting-started/shell/installation/> to set up the mongod as a windows service.
12. Setting up the service:-
    1. Open an Administrator command prompt.
    2. Create directories for your database and log files:

mkdir c:\data\db

mkdir c:\data\log

* 1. Create a configuration file.

systemLog:

destination: file

path: c:\data\log\mongod.log

storage:

dbPath: c:\data\db

* 1. Install the MongoDB service.

Install the MongoDB service by starting mongod.exe with the --install option and the -config option to specify the previously created configuration file.

"C:\Program Files\MongoDB\Server\3.4\bin\mongod.exe" --config "C:\Program Files\MongoDB\Server\3.4\mongod.cfg" --install

**Mongoengine**

1. Use pip install mongoengine after installing python with pip.
2. Make sure to run the install command when the command prompt is opened as an administrator.
3. Pymongo is installed along with mongoengine, if not use pip install pymongo.

**Bottle**

* 1. Use pip install bottle.
  2. When running bottle make sure the port is empty.

# APPENDIX-B

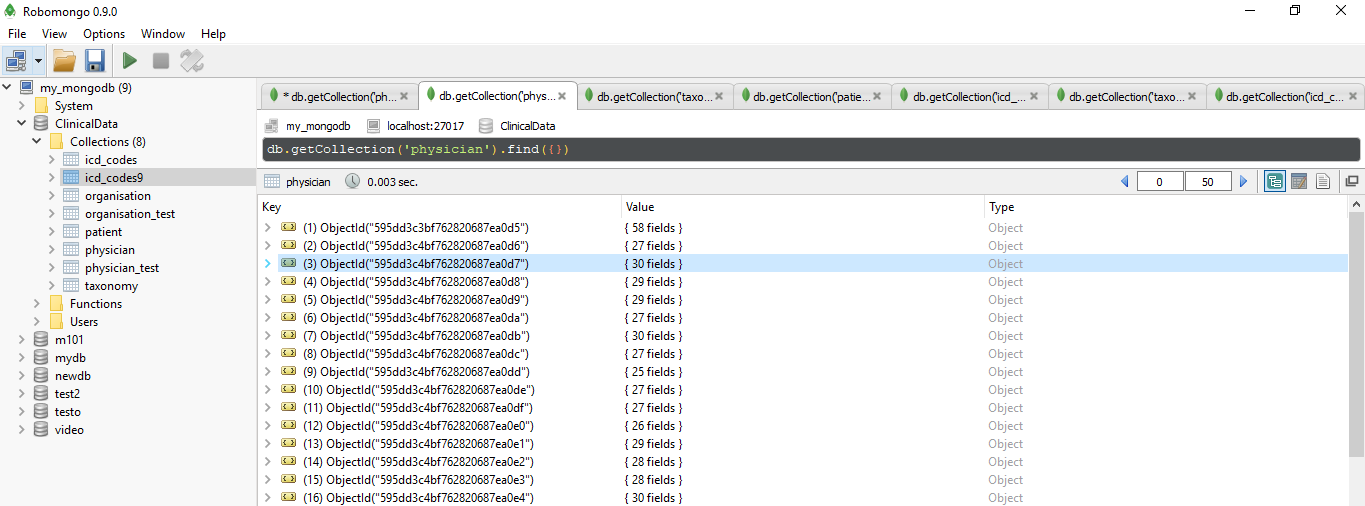
**DEVELOPMENT TOOLS**

1. GitHub: Version Control and Configuration Management

GitHub is a web based version control repository and Internet hosting service. It provides access control, source code management and is used to host open source software products. It offers public and private repositories. All the versions of the code at each commit are preserved from the time the code was added to the repository. In case of errors or bug tracking, the previous versions of the code can be accessed and retrieved. The URL for the current GIT project is <https://github.com/Srivatsava96/Internship>.

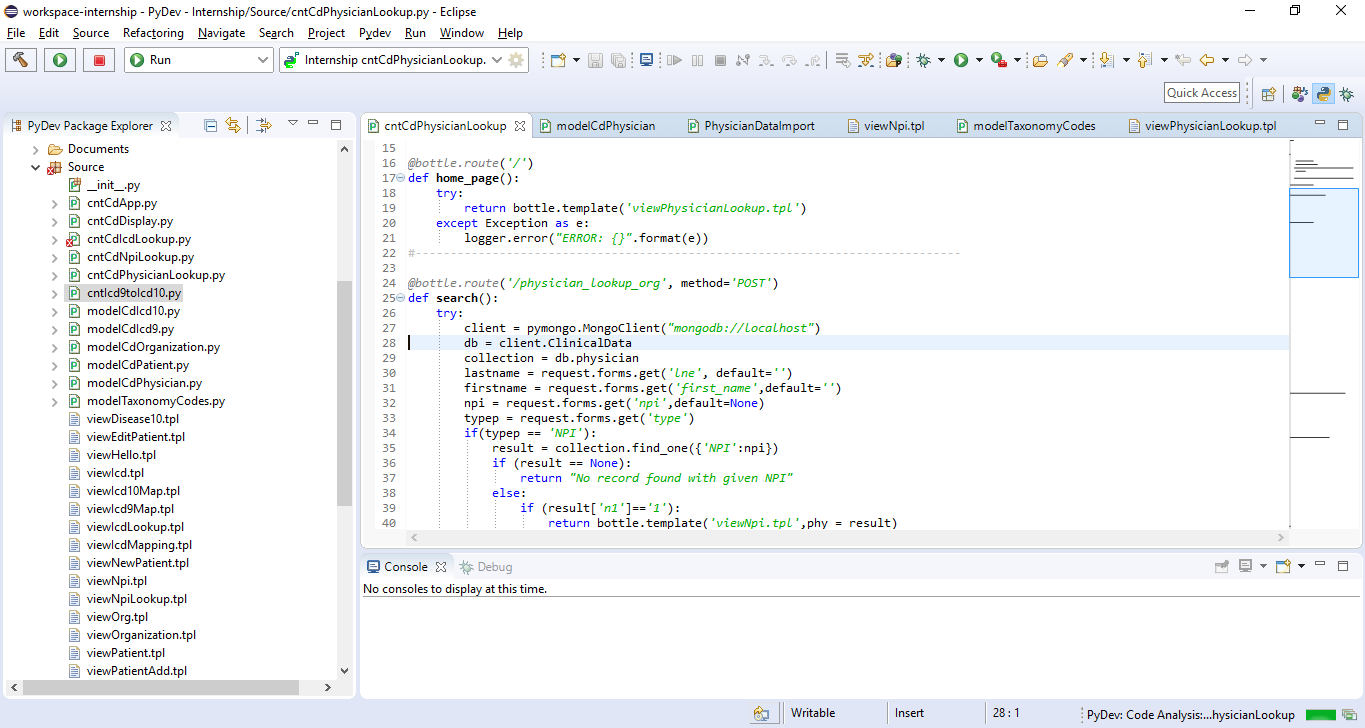
1. RoboMongo, Version-0.9.0: DB administration and Visualization

RoboMongo is a MongoDB management tool. We can use it to connect to the mongo server and view the database. We can run queries on the collections present in the database. The documents present in the collection can be visualized in the JSON representation. We can add, modify and drop collections present in the database. This tool is used to check whether the data has been correctly imported to the database.



1. Eclipse IDE Version: Neon.3 Release (4.6.3):

Eclipse is an integrated development environment(IDE) used in programming. An IDE is a software application that provides comprehensive facilities to computer programmers for software development. This project, has used the Python programming language which was written in the Eclipse IDE. Eclipse IDE offers features such as code completion, resource management, debugging tools, compile and build the object files. These features are highly useful in developing the software. Navigation through the file system is made easier. All the files can be viewed and managed on the same screen.



1. Task Assignment and Issue Tracking: Git Issue:

The entire project has been broken down into small tasks and each task has been created as an issue in GitHub issues. These issues have been given the label task and the label verify is added when the task is done. The commits to the source code can be associated with the issues in Git. These commits are then verified and the issue is closed. Comments for each issue can be written by any of the collaborator for the project which can be used to know the status of the issue between the collaborators. Each issue can be associated with a milestone. These milestones can be given the deadlines and they are closed when all the issues pertaining to the milestone have been closed. These milestones can be used to keep track of the sprints in the agile model of development.

