

EPIDEMIC ANALYZER

A research and development Computing Project on epidemic data management and analysis during humanitarian scenarios.

Biju Ale

College ID: [160026]

NCC ID: [0016399]

Computing Project
Level 5 Diploma in Computing

Softwarica College of IT & E-Commerce,
Kathmandu, Nepal

October 22, 2017

Acknowledgement

I am extremely grateful to all my supporters whose input, feedback, and backing went onto making my envisaged project a reality. This includes my college, Softwarica providing me the ecosystem with its amenities to carry out my academic undertakings, in particular this project.

Special gratitude goes to David Wallace Smith, Krishna Bahadur Thapa, and Andrew Pearlman at Folkekirkens Nødhjælp's South Asia Regional Office, Jhamsikhel, Lalitpur, Nepal (aka DanChurchAid Nepal) who gave me the opportunity to garner real-time experience of humanitarian response on the ground which opened up a whole gamut of possibilities for me, in leveraging technology to solve humanitarian problems. This project was largely inspired in DCA.

I am thankful to my module leader, Sudeep Lal Bajimaya for diligently supervising my project and providing me the guidelines for software analysis, modelling, implementation and documentation methods.

I am indebted to lecturer Achyut Timsina, perhaps the most incisive mind in Softwarica and altogether a magnanimous person, who laid in me the true foundations of object-oriented systems analysis and development. His invaluable insights into the subject matter illuminated the intricacies. Moreover, he provided a holistic learning experience His teachings shall be treasured and applied rightly forever.

I am grateful to lecturer Kiran Rana for providing me the references on pertinent technical resources, and for enthusiastically motivating me to go for enterprise level technologies. Timisna and Rana had a self-directed positive concern for my project and was always at the stance of helping me.

I should thank lecturer Manoj Shrestha for his March 2017 workshop on digital citizenship, where his trainings and materials on referencing and development of quality academic projects proved extremely helpful and applicable here.

Likewise, I should notably mention lecturers Nawaraj Bhandari, Shyam Sundar Khatiwada, Nishant Shrestha, Niman Maharjan, Pratik Bhusal, Madan Sharma Pokhrel (former), Chandan Deo and Roshan Maharjan. Without them, I would not have the foundations in information technologies to build this project.

Rekha Kapali and [] [] have always been courteous to me, responding favorably to my email requests, and helping with Turnitin account, verifications and submission of project materials.

Lastly, my heartfelt gratitude goes to the Lamb of God, my family and friends who were understanding and supportive in wonderful ways.

Biju Ale
22.10.2017
Kathmandu, Nepal

Abstract

This report documents all the activities executed in each of the phases of software development lifecycle of the project titled – 'Epidemic Analyzer'. All the requirements, use cases with scenario description, system architecture, the static and dynamic models and implementation details along with testing report and user manual is included in this report.

The project is a data management and analysis web based application that designed to be used in epidemic situation. It allows for persistence and management of epidemic data pertaining to patient cases and provides data visualization on geographic map. Based on the current infection status of a patient, the information is categorized and displayed accordingly. The system also integrates a simulation module based on the Kermack-McKendrick SIR Model ([Weisstein, 2017](#)). In a nutshell, humanitarian actors are benefited by this system as it provides an efficient system for epidemic data management and analysis.

Technical aspects consist of the system using Java 8 as the core back-end language with Spring Boot as primary framework while also including dependencies of other frameworks. Also, Javascript was used for front end programming and the view is rendered through Thymeleaf, the Java template engine for web and standalone environment. MySQL was used as database.

EPIDEMIC ANALYZER

Table of Contents

Acknowledgement	2
Abstract.....	3
CHAPTER 1 INTRODUCTION	9
Background of the system	10
Aims	10
Objectives	10
Overview of the design	11
CHAPTER 2 Analysis.....	12
Introduction to analysis	13
Need for Analysis	13
Object Oriented Analysis	13
Pitfalls of Classical or structured paradigms	13
Merits of Object Oriented Analysis (OOA) methodology	14
Activities involved – from use case to objects	14
Requirements.....	16
Information gathering for Epidemic Analyzer.....	17
Online research and web-journals	17
Personal experience & field research - humanitarian domain	17
Functional requirements.....	17
Non-functional requirements	20
Requirement Prioritization	21
Use Case	23
Uses cases for better understanding	23
Use Case for Epidemic Analyser	23
Title: “add case”	25
Title: “update case”	25
Title: “delete case”	25
Title: “view case”	26
Title: “search case”	26
Title: “sort case”	26
Title: “view map-projected case”	27
Title: “close session”	27

EPIDEMIC ANALYZER

Title: "simulate SIR model"	27
Title: "view user-manual"	28
Architecture	29
N-Tier System Architecture	29
Initial Domain level architecture.....	30
Application Architecture	33
CHAPTER 3 Design.....	34
Dynamic modelling	35
Sequence diagram.....	35
Activity diagram	39
Structural modelling	42
Architecture Level class diagram	42
CHAPTER 4 Implementation	44
Programming Language used	45
Kermack-McKendrick Model.....	45
System cutover from development architecture to the implementation	46
CHAPTER 5 Testing.....	47
Unit Test.....	48
Patient Case CRUD operations.....	48
Patient case data projection on map	49
SIR Modelling	49
Integration Testing.....	50
CHAPTER 6 Other Project Issues	56
Risk Management	57
Configuration Management.....	60
CHAPTER 7 Conclusion	61
CHAPTER 8 References & Bibliography.....	63
CHAPTER 9 Appendix	65
Code Snippets	66
Source code directory structure	66
ea.biju.Application.java.....	71
ea.biju.controller.IndexController java	72
ea.biju.controller.MapProjectionController.java.....	73

EPIDEMIC ANALYZER

ea.biju.controller.PatientCaseController.java	77
ea.biju.controller.SIRModellingController.java	81
ea.biju.enums.CurrentInfectionStatus.java	82
ea.biju.enums.Gender.java	83
ea.biju.model.Address.java	84
ea.biju.model.Patient.java	87
ea.biju.repo.AddressRepository.java	90
Ea.biju.repo.PatientRepository.java	91
ea.biju.security.SecurityConfig.java	92
static.css.styles.css	93
static.js.map_projection_logic.js	94
static.js.tabledata_logic.js	94
static.xml.EpidemicMapProjectionData.xml	95
static.templates.error.html	96
static.templates.fragments.html	97
static.templates.index.html	99
static.templates.patient_case.add_patient_case.html	101
static.templates.patient_case.display_patient_cases.html	103
static.templates.patient_case.epidemic_map_projection.html	105
static.templates.patient_case.sir_modelling.html	111
static.templates.patient_case.update_patient_case.html	116
pom.xml	118
nbbactions.xml	121
nb-configuration.xml	122
Test Scripts	123
Patient Case CRUD operations	123
Patient case data projection on map	128
SIR Modelling	130
User Manual	132
Logging in	132
Adding new patient case	133
Displaying all patient cases	135
Searching and sorting results – patient cases	136

EPIDEMIC ANALYZER

Deleting patient case	136
Updating patient case	138
Projecting patient cases on geographic map	139
Interacting with map.....	140
Information window	140
Legend.....	140
Zoom & navigate on map.....	140
Satellite view	141
SIR Modelling	141

EPIDEMIC ANALYZER

Figure 1: A UML representation of artefacts producing post-analysis.. (Duoit)	15
Figure 2 UML representation of Artefacts of requirement elicitation and analysis. (Dutoit, page 123)	16
Figure 3 Use Case Diagram - Epidemic Analyzer	24
Figure 4: N-Tier System Architecture	29
Figure 5: Initial domain level architecture	30
Figure 6: Class diagram legend and reference	31
Figure 7: Application architecture	33
Figure 8: Sequence Diagram - SIR Modelling	35
Figure 9: Sequence Diagram: Map Generation	38
Figure 10 Activity Diagram: Authentication	39
Figure 11: Activity Diagram Map Generation	41
Figure 12: Activity Diagram - SIR Modelling	41
Figure 13: Application architecture	42
Figure 14: Risk Likelihood values (Turner, 1999)	57
Figure 15: Risk consequence values (Turner, 1999)	57
Figure 16: Directory Structure for Project in the local machine.	60

CHAPTER 1 INTRODUCTION

Background of the system

On March 23, 2014, an epidemic of Ebola Virus Disease (EVD) in Guinea was reported to the World Health Organization (WHO). On August 8, the situation was declared to be a “public health emergency of international concern”. ([PMC NCBI, 2014](#)) . The following 2 years saw a total case fatality of 11310 in West Africa, as the EVD diffused in Sierra Leone and Liberia. ([cdc.gov, 2016](#))

In such milieu of a humanitarian crisis, the responders including public health personnels require epidemic data and its analysis. Such information is paramount for decision makers whose discernments and actions have direct impact on the control of the outbreak, consequently saving lives. Thus this system is developed an epidemic analysis system that aids in the acquisition of data and subsequently provides certain analytics on it.

Aims


1. To build a system that can capture epidemic data on ground zero and provide analytics for provincial and national level decision makers.
2. To contribute to the understanding of public health practitioners and humanitarian agencies about an epidemic situation through the introduction of information technologies.

Objectives


1. To provide a robust data entry form that captures a broad range of health data dynamics.
2. To provide essential analytics on epidemic data.
3. To provide a graphical representation of data for better comprehension & decision making.
4. To provide the simulation of disease spread and impact using SIR model.
5. To learn about the impact of information technology in international humanitarian interventions.
6. To apply the principles from academic learnings to the real world scenarios and garner experience in research and development.

EPIDEMIC ANALYZER

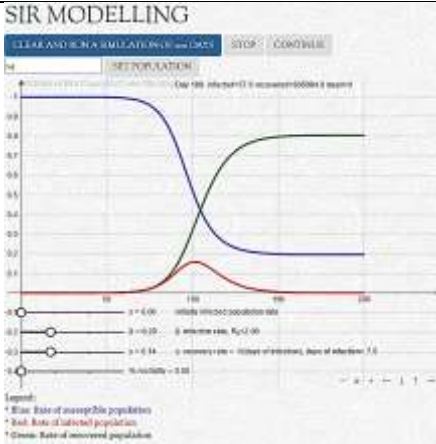
Overview of the design



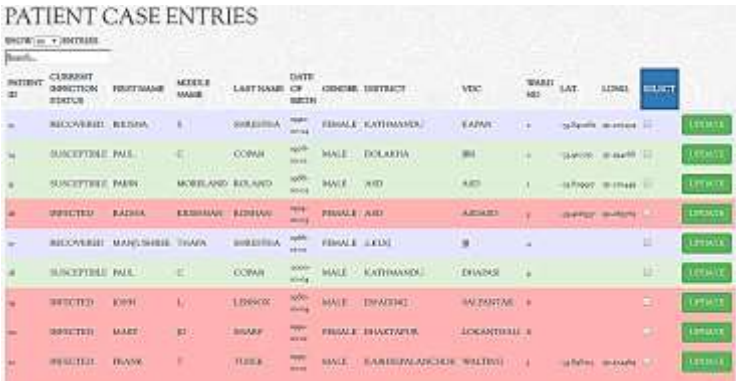
Epidemic Data is gathered from field and entered into the system



Geographic projection will help humanitarian actors make informed decisions.



Based on the patient's data susceptible, infection, and recovered population's impact can be analyzed.



The gathered data can be persisted and managed effectively. These data can be extended and saved for further analysis by other stakeholders.

Table 1: Rich picture of Epidemic Analyzer

The rich picture in table 1 above, can be a fairly apprehensive starting point to get the bigger picture, of what the system is about and what its design would entail.

The target group for the system is humanitarian responders, for example health officers or WASH officers from the UN who are deployed on field during an epidemic scenario. The design is guided by such user in mind. The system is intuitive to use, the user enters epidemic data from the field into the system and the system provides certain analytics on it. This is only a first prototype of the system therefore; database schemas are not extensive. Only a sample of patient case attributes were taken to design the system. However, it can be extended in future. The current system is fully functional and uses WHO's case investigation forms to standardize epidemic data attributes. It can be deployed in real time scenarios.

CHAPTER 2

ANALYSIS

Introduction to analysis

The etymological origin of the word, Analysis traces back to Ancient Greek, from ana – ‘up’ and lysis – ‘loosening’. The cross-cutting connotation therefore, whether used as an analysis of culture or be it a statistical analysis for that matter, is to examine a fairly complex entity, segregating it into its simpler dimensions.

Need for Analysis

Likewise, in software systems, it is the analysis of the requirements that will eventually translate into the proof of concept – modelling, that will be implemented. Thus, primary goal of analysis in software domain is to understand the requirements first and then to formalize it. ([Brugge et al, 2014](#)) This requirement can be looked at various levels of abstraction and categories, as guided by the methodology chosen for such as an analysis.

Object Oriented Analysis

Contrary to structured systems analysis which focuses on hard systems, object oriented analysis (OOA) is a semiformal analysis technique that maps to real world objects and thought inferences. It is adapted in my project as the analysis methodology.

OOA was chosen over other analysis paradigms due to following reasons.

Pitfalls of Classical or structured paradigms

- **Scalability issue**

Considering the potential deployment of my project in real- time scenarios in future, it may need to be integrated with more features and modules. However, classical techniques do not allow this. ([Schach, 2011](#))

- **Post-delivery maintenance issue**

Adopting classical techniques has resulted in 70-80 percent or even more time and effort on post-delivery maintenance alone

- **Limited user involvement**

If this project deploys in real-time, user’s constant involvement to validate the release is required. All aspects of project may need to be reviewed once again. Other methods focus on technical aspects with little to no user involvement.

- **Lack of flexibility**

All requirements may not be clear at the beginning, as our understanding grows over time, there may be realizations and need for requirement fine-tuning as we traverse to later stages of the development lifecycle. This is a costly process since; waterfall models do not allow this.

Merits of Object Oriented Analysis (OOA) methodology

- **Maintainability**

If at any phase of the project lifecycle, a change needs to be introduced in the system. Object oriented analysis produces such an architecture, that there is little to no **regression fault** after the change is implemented. This is due to application of principles like dependency inversion, abstraction, and loose coupling. ([Schach, 2011](#))

- **Ease of modelling**

Since, real world object mostly directly maps to the logical objects in OOA, it is easier to envisage the components of the system and how they interact. Its concepts like **encapsulation** favors intuitive and logical thinking and abstracts away from problem or machine domain into the more familiar solution or natural domain.

- **Promotes reusability**

Modular nature of OOA focusing on separation concerns and high cohesion between components, promotes reusability of individual components. This saves time, resource, and effort to replicate and reuse their similar affect in other areas of system.

Activities involved – from use case to objects

Analysis involves modelling of 3 aspects of the system. ([Brugge et al, 2014](#))

Functional modelling

Some authors regard functional and dynamic modelling under same category. However, here by functional modelling refers to the high level analysis of interaction between users and the system. The role abstractions and their use cases. Use case diagram is used here.

- **Dynamic modelling**

It is depicted via state machine or sequence diagram. For this project the latter is employed. It focuses on modelling the behavior of the system i.e. how various components in the system function, interact, and change during the run-time execution.

- **Static modelling**

It is depicted via class diagram. Static modelling aims to show the relationship between system components and how they are structured. It is the framework or platform sitting on which the system's dynamic activities are orchestrated.

Mainly the first version of class diagram is the product of preliminary requirements analysis, resulting from natural language analysis. It is focused on only the domain level modelling.

EPIDEMIC ANALYZER

The second version, which is more detailed and can be considered the actual representation of system architecture, depicts the design pattern element and crosscuts into all layers i.e. presentation, logic, and persistence.

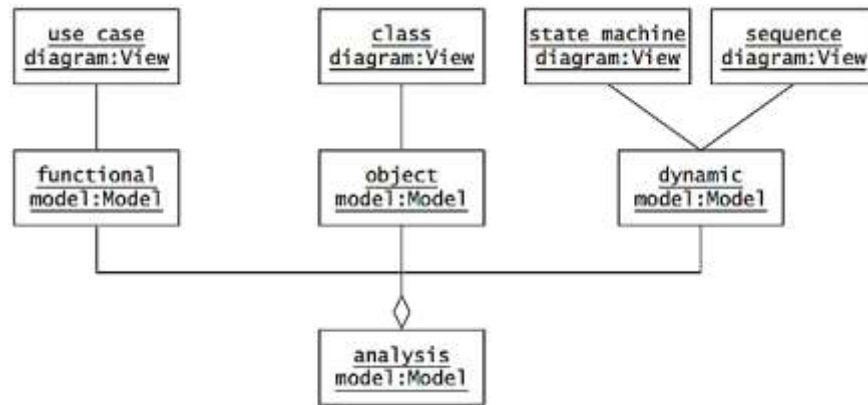


Figure 1: A UML representation of artefacts producing post-analysis.. (Duoit)

Requirements

A **requirement** is a feature or functionality that should be present in a system, or a constraint on the system required to be conformed which are elicited from targeted users and accepted by them. ([Brugge et al, 2014](#))

Requirements engineering is a preliminary set of activities carried out before the actual software modelling and development. It is conducted iteratively before every user story entry into a product backlog in agile methodology whereas in waterfall, it is a onetime affair done at the beginning.

As shown in figure 3 below, It consists of mainly 2 activities –

- requirement elicitation,
- and analysis

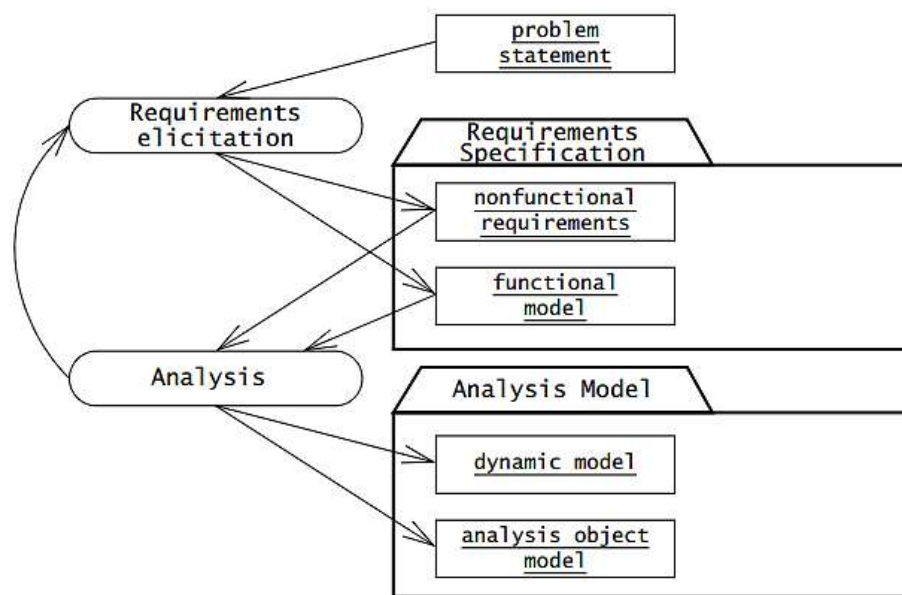


Figure 2 UML representation of Artefacts of requirement elicitation and analysis. ([Brugge et al, 2014](#))

that ultimately leads to a model that the system will be based on. Thus, It decides on the outlook of the system that will be developed and how functionally true it is compared to the client's envisaged model.

As shown in the diagram, requirements engineering begins with a problem statement that is a single statement

Out of the two activities, the requirement elicitation is a more challenging task and needs careful analysis, communication and discernments. It is so for mainly 2 reasons. The software developer does not necessarily have the domain knowledge and needs to constantly communicate with the client to verify every requirement. And the client may not have technical knowledge and thus the software developer must edify the client on the limitations and show alternatives

EPIDEMIC ANALYZER

Information gathering for Epidemic Analyzer

As explained above, requirements engineering is a paramount undertaking in a software project. For this particular project, I have used primary source of information based on my experience and largely the information is gained through secondary sources i.e. through research on web journals and books. This project is not mandated by any client rather the need for such project was felt by myself.

Online research and web-journals

List of all sources can be found in the references section of this report.

Personal experience & field research - humanitarian domain

An epidemic scenario such as the 2014 Ebola or a regional H1N1 can be a major humanitarian crisis existing on its own. In some cases, A disaster situation such as earthquake can trigger epidemic crisis due to degrading water, sanitation, and hygiene provisions.

During my visit & research to one of the leading humanitarian organizations (Folkekirkens Nødhjælp's South Asia Regional Office, Jhamsikhel, Lalitpur, Nepal) responding to the Nepal Earthquake I discovered that agencies mostly relied on 3W/4W analysis of OCHA's cluster reports ([Goo.gl, 2016](#); [Data.hmdata.org, 2016](#)), which all agencies were required to fill. All the data were locked into massive spreadsheets. & any kind of analysis was extremely difficult & time consuming. All this was being taken into account during my field research work, before proposing my system.

Functional requirements

ID	Title	Description	Rational	Dependencies
F1	User authentication	Users with their roles are registered in the backend in the system's embedded memory (JPA in-memory). When base URL is hit, credentials are asked & verified to allow or deny access.	To maintain confidentiality, integrity, and authenticity by allowing only verified users to access the system.	NA
F2	Case entry	Users should be able to enter patient cases. Bean validation must reflect on view with appropriate message on invalid inputs.	To add patient records of epidemic for persistence, manipulation, and analysis.	F1
F3	Case update	Users should be able to edit existing records. Validation must be done at	To edit existing patient records to maintain data integrity & semantic correctness.	F2

EPIDEMIC ANALYZER

		both front and back end.		
F4	Case deletion	Users should be able to delete existing records individually or in batch. A confirmation dialog box must be shown before executing the action.	To purge irrelevant, erroneous, or obsolete records.	F2
F5	Display all cases	Users should be able to view all existing records.	To view all existing records to perform sorting, and searching	F2
F6	Case search	Users should be able to search cases with input string and without page reloads on existing single display of records.	To locate specific epidemic cases.	F2
F7	Case sort	Users should be able to order the view of all records based on all attributes.	To order by patient attributes on existing records.	F2, F5
F8	Geographic map projection of cases	Users should be able to view a geographic map in real time (Google Map) and view all cases visualized on the map. The map should have interactivity and controls (google provided).	To get an overview of disease propagation over geographic layout. To get current infection status and individual case information on map.	F2, F5
F9	Custom markers on map	Users must be able to distinguish patient case types based on current infection status based on 3 types of marker icons used.	To identify patient's current infection status – infected, susceptible, or recovered.	F2, F8
F10	Map legend	Users must be able to view the legend for markers on the map.	To describe symbolic meanings used on map.	F8

EPIDEMIC ANALYZER

F11	Individual Marker interactivity to view information	Users must be able to zoom-in, zoom-out, navigate to other places on map, and click on the markers to view specific details of patient case.	To provide interactivity with map to show individual case information on click of a marker.	F8, F9
F12	Simulation of SIR Model	Users must be able to run a simulation of SIR model and get a 3 curve differential graph to view rate of recovery, infection, and susceptibility change over time on a population.	To provide prediction of epidemic impact over a population. To provide trend in disease propagation over time.	F2
F13	Input field for population	User must be able to provide custom population to pass on the SIR model.	To set a custom population for the SIR model to run on.	F12
F14	Slider controls for changing SIR model	Users must be able to adjust SIR model parameters – transmission rate, recovery rate, mortality, and initial infection population rate.	To change and adjust model parameters like recovery rate and transmission rate to make SIR model true to the epidemic being analyzed.	F12
F15	User manual	Users must be able to look for help within the system on every use cases.	To provide a guide to user and explain about all use case and how to effectively use the system. To provide a catalog for future reference.	F2-F17
F16	User session termination	User must be able to logout.	To allow admin to manually log out of current session for security.	F1
F17	Navigation for all use cases	Users must be able to perform all use cases with a fixed navigation bar.	To allow users to perform all tasks on system via an user-friendly front end.	F2 – F16

The functional requirements tabulated above is to some extent implementation dependent. Since, technologies such as APIs and libraries to be used in the system were researched before-hand and their usability were considered best fit to fulfill the for the system's requirements. These implementation details include libraries and APIs to plot graphs and project map visualizations. Specific tools used are detailed out in the analysis section (see :)

EPIDEMIC ANALYZER

Non-functional requirements

ID	Category	Description
NF1	Usability	The system should be fairly intuitive to use. Proper navigation bar must be displayed. The user should be provided with online user guide for reference and self-guided help.
NF2	Reliability	The system must have 99.9% uptime. Data integrity must be maintained and the analysis must be reliable.
NF3	Performance	There should not be latency and lags more than 2 seconds. This should occur rarely. Overall, the system must respond to user inputs at real-time speed.
NF4	Supportability (including portability & maintainability)	The system must be flexible so as to allow addition of geographic maps and models. System must be allowed for a different view integration or modification altogether easily. Loose coupling must be maintained using dependency injections and auto wiring.
NF5	Implementation	System must run on smoothly on any modern browser from the client side. A 2Ghz processor and 1gb RAM should suffice smooth functioning. No manual installation should be required.
NF6	Interface	Interface must be web-based with intuitive usability. Proper notifications and validations must be done.
NF7	Security	Users must be authenticated by the system as the site is entered. Only valid users with sufficient privileges must be allowed to use the system.

EPIDEMIC ANALYZER

Requirement Prioritization

ID	Requirement	MoSCoW	Rational
F1	User authentication	M	Crucial for security, only authenticated users of agency are allowed access.
F2	Case entry	M	Fundamental function within the system.
F3	Case update	M	Fundamental function within the system.
F4	Case deletion	M	Fundamental function within the system.
F5	Display all cases	M	Fundamental function within the system.
F6	Case search	W	For convenient data handling.
F7	Case sort	W	For convenient data handling.
F8	Geographic map projection of cases	M	Fundamental function within the system.
F9	Custom markers on map	M	Provides graphical information.
F10	Map legend	M	Provides graphical information.
F11	Individual Marker interactivity to view information	M	Provides graphical information.
F12	Simulation of SIR Model	M	Provides graphical information.

EPIDEMIC ANALYZER

F13	Input field for population	M	Simulation input data.
F14	Slider controls for changing SIR model	M	Simulation input data.
F15	User manual	M	Fundamental function within the system.
F16	User session termination	W	For logging out from current session.
F17	Navigation for all use cases	M	Fundamental function within the system.
NF1	Usability	M	Fundamental function within the system.
NF2	Reliability	M	Fundamental function within the system.
NF3	Performance	M	Fundamental function within the system.
NF4	Supportability (including portability & maintainability)	M	Fundamental function within the system.
NF5	Implementation	M	Fundamental function within the system.
NF6	Interface	M	Fundamental function within the system.
NF7	Security	M	Fundamental function within the system.
P1	Extended information for patient cases such as symptoms, and lab tests conducted.	W	Fundamental function within the system.

Use Case

Brugge and Dutoit, in their book “Object-oriented software engineering: Using UML, Patterns, and Java” ([Brugge et al, 2014](#)) presents use case diagrams as a tool for both high level requirement analysis and to some extent for the behavioural modelling.

Uses cases for better understanding

The aim of Use Case diagram is to depict the relationship or the method of interaction between actors and the system such that the externally exposed functionalities of the system are employed by the actors. Here, each of these functionalities are known as use case. And, the actors are role abstractions and not necessarily concrete objects or humans.

Use Case for Epidemic Analyser

The diagram below (Figure 2) shows the use case for “Epidemic Analyzer”. Subsequently, each use cases are described in their scenarios. Developing use cases, is fundamentally and for most part, a scenario expounding activity over merely presenting diagrams. ([Larman, 2012](#))

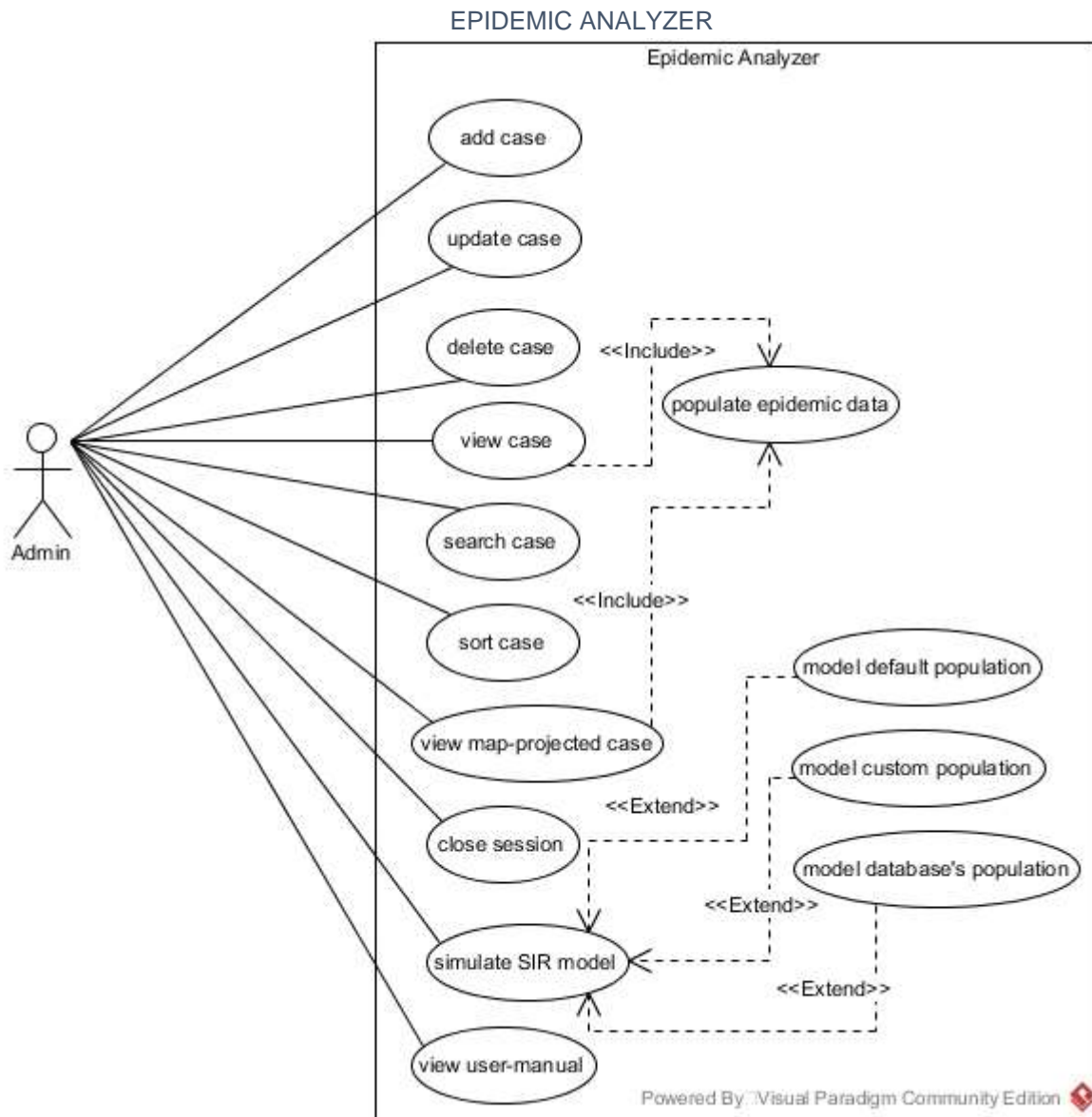


Figure 3 Use Case Diagram - Epidemic Analyzer

EPIDEMIC ANALYZER

Title: "add case"

ID	UC1
Justification	A crucial and fundamental use case which persists the entirety of data through the system, which will be used by all other use cases.
Primary actor(s)	Admin
Supporting actor(s)	NA
Primary Flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User navigates to the 'add patient case' form. 3. User inputs all valid data into the fields and submits the form. 4. System adheres to bean validation and verifies valid input. 5. System persists the data in database. 6. System redirects to new input form.
Alternative Flow	<ol style="list-style-type: none"> 4.1 User inputs invalid data into the field <ol style="list-style-type: none"> 4.1.1 System scan bean validation annotation. 4.1.2 System prints default or custom message for invalid input via model-view binding. 4.1.3 User expected to correct the field data. 4.1.4 Repeat from 4.1.1 until correct input.

Title: "update case"

ID	UC2
Justification	Important for correcting uncaught invalid data that as persisted in the database, to update cases to maintain data integrity.
Primary actor(s)	Admin
Supporting actor(s)	NA
Primary Flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User navigates to the 'view patient cases' option. 3. System displays all existing cases to user. 4. User clicks on the corresponding update button of the patient data. 5. System redirects to 'update patient case' form with current data in field. 6. User edits the current data in the field and submits the form. 7. System adheres to bean validation and verifies valid input. 8. System persists the updated data in database. 9. System redirects to page to display all existing cases.
Alternative Flow	<ol style="list-style-type: none"> 7.1 User inputs invalid data into the field <ol style="list-style-type: none"> 7.1.1 System scan bean validation annotation. 7.1.2 System prints default or custom message for invalid input via model-view binding. 7.1.3 User expected to correct the field data. 7.1.4 Repeat from 7.1.1 until correct input.

Title: "delete case"

ID	UC3
Justification	Semantically erroneous data or data not irrelevant to the system can be delete using this use case.
Primary actor(s)	Admin
Supporting actor(s)	NA

EPIDEMIC ANALYZER

Primary flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User navigates to the 'view patient cases' option. 3. System displays all existing cases to user. 4. User check-marks on the corresponding delete checkbox of the patient data and clicks on delete button. 5. System intercepts the selected checkboxes and prompts confirmation dialog box to user. 6. User confirms delete action. 7. System deletes the data from database.
Alternative flow	<ol style="list-style-type: none"> 4.1 User check-marks multiple checkboxes to delete patient cases in batch. <ol style="list-style-type: none"> 4.1.1 System intercepts batch selection. 4.1.2 Continue flow of execution. 6.1 User declines delete confirmation. <ol style="list-style-type: none"> 6.1.1 System redirects to displaying all cases to user.

Title: "view case"

ID	UC4
Justification	This use case provides function for other use cases such as by displaying all cases along with action buttons, user can view and manipulate patient cases from a single page conveniently.
Primary actor(s)	Admin
Supporting actor(s)	NA
Primary flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User navigates to the 'view patient cases' option. 3. System displays all existing cases to user
Alternative flow	NA

Title: "search case"

ID	UC5
Justification	User can drill down to specific records instantly and without page reloads, and with best possible matches for the given input. This is useful when trying to locate data from large pool.
Primary actor(s)	Admin
Supporting actor(s)	NA
Primary flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User navigates to the 'view patient cases' option. 3. System displays all existing cases to user 4. User types search string in the input to search specific case. 5. System without page reload, locates and singles out the best match.
Alternative flow	<ol style="list-style-type: none"> 5.1. User input string doesn't match any patient data in the system. <ol style="list-style-type: none"> 5.1.1 A 'no record matches with input' message is displayed.

Title: "sort case"

ID	UC6
-----------	-----

EPIDEMIC ANALYZER

Justification	Enables users to sort data by all fields. Useful to view patient cases ordered by fields like date of birth and current infections status.
Primary actor(s)	Admin
Supporting actor(s)	NA
Primary flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User navigates to the 'view patient cases' option. 3. System displays all existing cases to user 4. User clicks on the display table's columns to order the result.

Title: "view map-projected case"

ID	UC7
Justification	Important use case that aids humanitarian actors to visualize epidemic cases over geographic map and to see and overview of disease spread and current infection status of registered patients.
Primary actor(s)	Admin
Supporting actor(s)	NA
Primary flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User navigates to the 'view map-projected cases' option. 3. System fetches all patient cases from the database. 4. System uses project's API key to make an API call to Google Maps. 5. System renders a google map on the page using the javascripts. 6. System customizes the map and injects fetched data into the map to plot 3 markers based on the current infections status. 7. User views and interacts with the map.
Alternative flow	<ol style="list-style-type: none"> 7.1 User clicks on any of the 3 types of marker on the map. <ol style="list-style-type: none"> 7.1.1 A info-window is displayed about the marker including the patient's name, id age and address.

Title: "close session"

ID	UC8
Justification	It allows the current admin to logout of the system. The system uses embedded (JPA in-memory)credentials to close current session of the admin.
Primary actor(s)	Admin
Supporting actor(s)	NA
Primary flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User interacts with the system using other use cases. 3. User clicks on the logout button. 4. System closes the current session. 5. System redirects to index page and asks for credentials.
Secondary flow	NA

Title: "simulate SIR model"

ID	UC9
-----------	-----

EPIDEMIC ANALYZER

Justification	This is a crucial and powerful use case for the humanitarian actor as it provides simulation of epidemic propagation over susceptible, infected, and recovered population.
Primary actor(s)	Admin
Supporting actor(s)	Na
Primary flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User interacts with the system using other use cases. 3. User navigates to the 'SIR Modeling' option. 4. System fetches total population i.e. no. of records in the system and places the value on the input field of total population. 5. System renders the SIR model using the 'JSXGraph', a javascript based mathematical graph visualization library. 6. User clicks on the set total population button. 7. User clicks on the 'clear and run simulation' button. 8. System runs the SIR simulation.
Alternative flow	<ol style="list-style-type: none"> 6.1 User sets different total population <ol style="list-style-type: none"> 6.1.1 If blank input, system sets 1E-6 as default total population. 6.1.2 If certain value, system sets the input value as total population.

Title: "view user-manual"

ID	UC10
Justification	Important use case which provides guidelines and aids users on how to employ the available various use cases of the system.
Primary actor(s)	Admin
Supporting actor(s)	NA
Primary flow	<ol style="list-style-type: none"> 1. User logs into the system with valid credentials. 2. User interacts with the system using other use cases. 3. User navigates to the 'user manual' option. 4. System displays a navigation enabled help page. 5. User reads the user manual displayed on the system.

EPIDEMIC ANALYZER

Architecture

N-Tier System Architecture

Epidemic Analyzer uses Spring MVC with abstraction of Spring Boot framework.

Models will be created as Java beans, and annotated with hibernate object-relational mapping annotations. Hence, most models will map to database entities.

Controllers will be used with simple POST and GET requests to execute all client requests and response. Request Mapping will be done using base url and corresponding service's section and handler invocations.

Since, REST or SOAP services are not used, the simple POST and GET requests will be consumed and for the view, Thymeleaf will be used as templating engine which is superior than JSP usage and will render high compatibility web pages. It will be injected with dynamic content with mode/form or object/view binding implementations.

The entities will be persisted and manipulated using Spring Data specifically via contracts of CRUDRepository provided by the JPA repository. Its underlying implementations will be provided by Hibernate.

Spring Security is used to manage authentication which uses embedded H2 DB, so it is not accessible elsewhere. The database will be generated at run time so schemas must not necessarily be manually designed. DB can be hosted at remote server separate from Tomcat which will host the entire application.

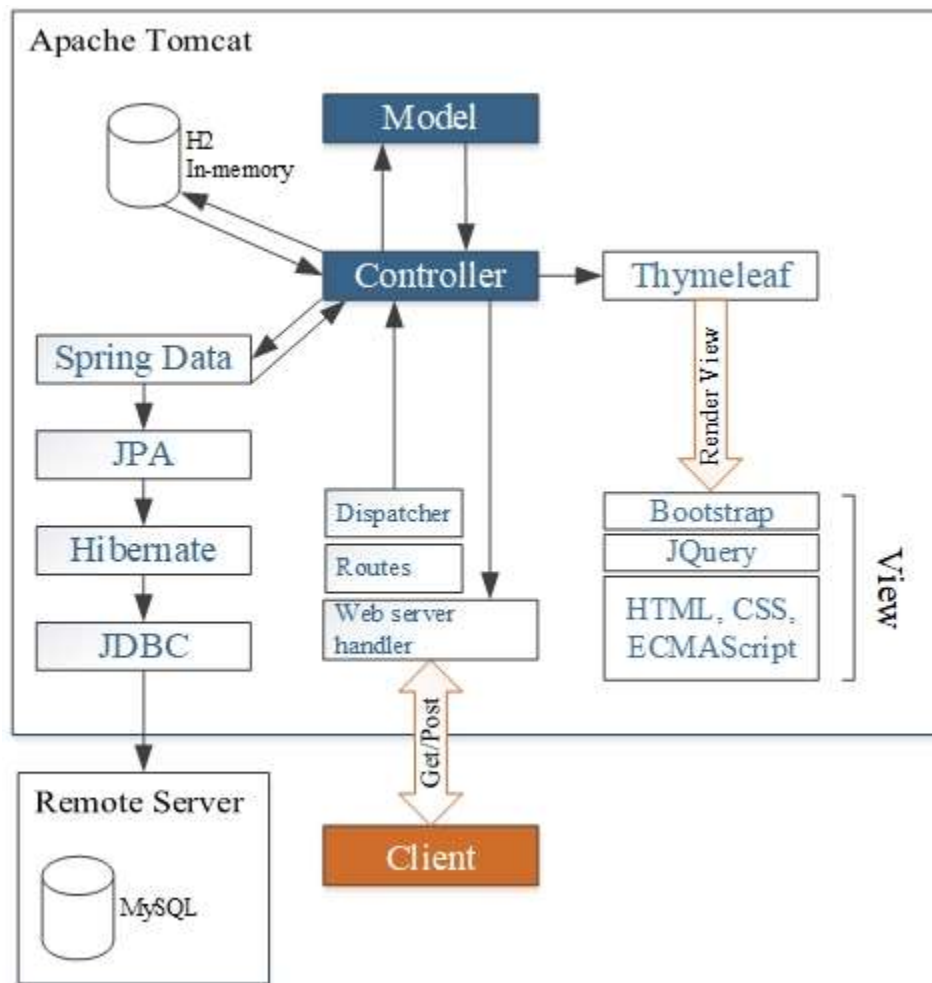


Figure 4: N-Tier System Architecture

EPIDEMIC ANALYZER

Initial Domain level architecture

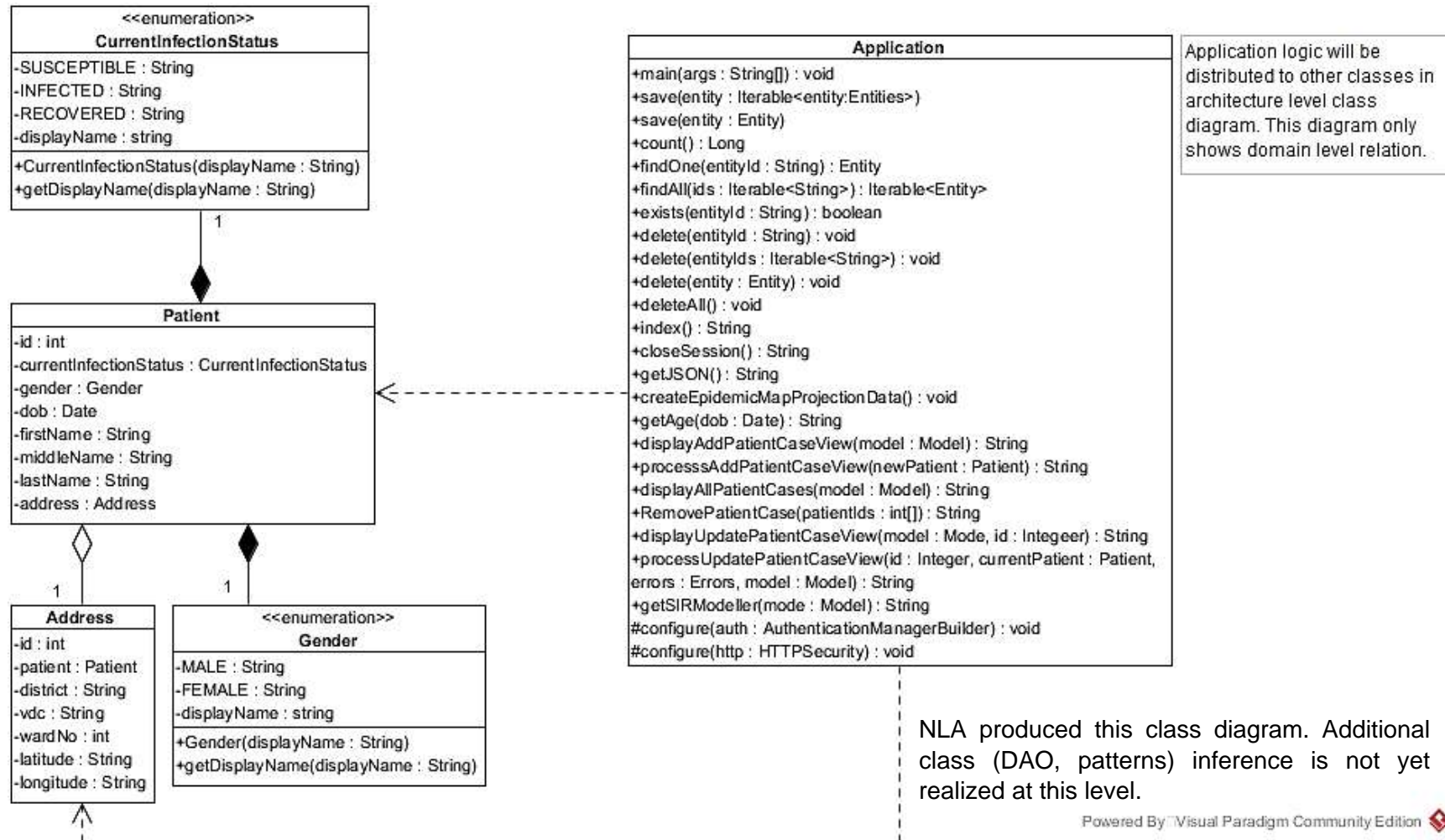


Figure 5: Initial domain level architecture

EPIDEMIC ANALYZER

Reference for interpreting presented class diagrams

- Diagram is based on **UML version 2.X**
- Modelling tool used was **Visual Paradigm Community Edition 14.2 (Build 20171005)**








Notation	Remark
 Generalization	Depicts inheritance from super/parent class to sub/child class.
 Realization	Depicts interface implementation.
	Depicts unidirectional association between classes.
 Association	Depicts bidirectional association between classes.
 Aggregation	Depicts 1 class aggregates another without binding its existence in own life.
 Composition	Depicts 1 class composes another binding its existence in its own life.
 Dependency	Depicts 1 class depends on another in such way that the change propagates to dependent.

Figure 6: Class diagram legend and reference

EPIDEMIC ANALYZER

EPIDEMIC ANALYZER

Application Architecture

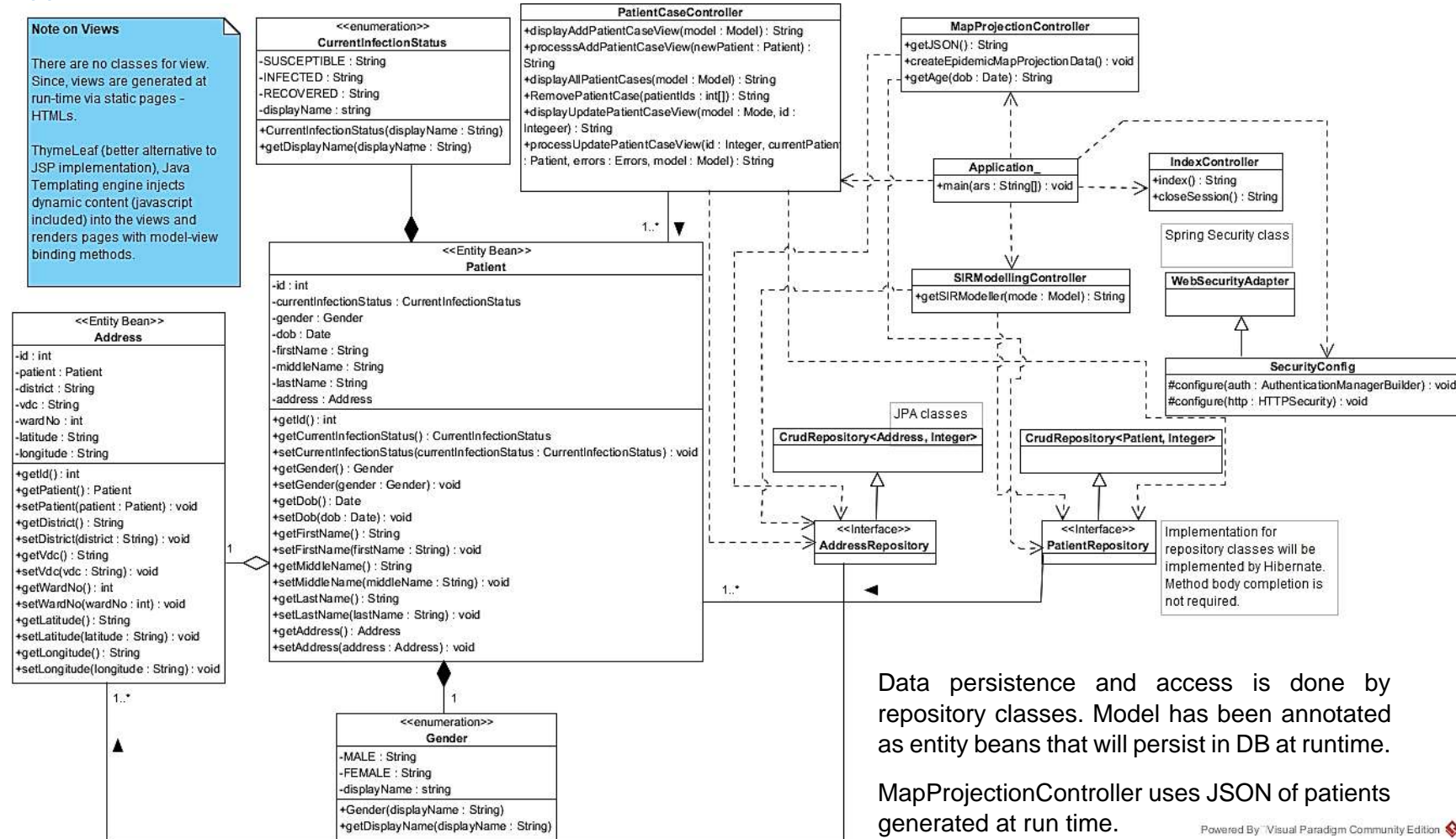


Figure 7: Application architecture

CHAPTER 3

DESIGN

EPIDEMIC ANALYZER

Dynamic modelling

Sequence diagram

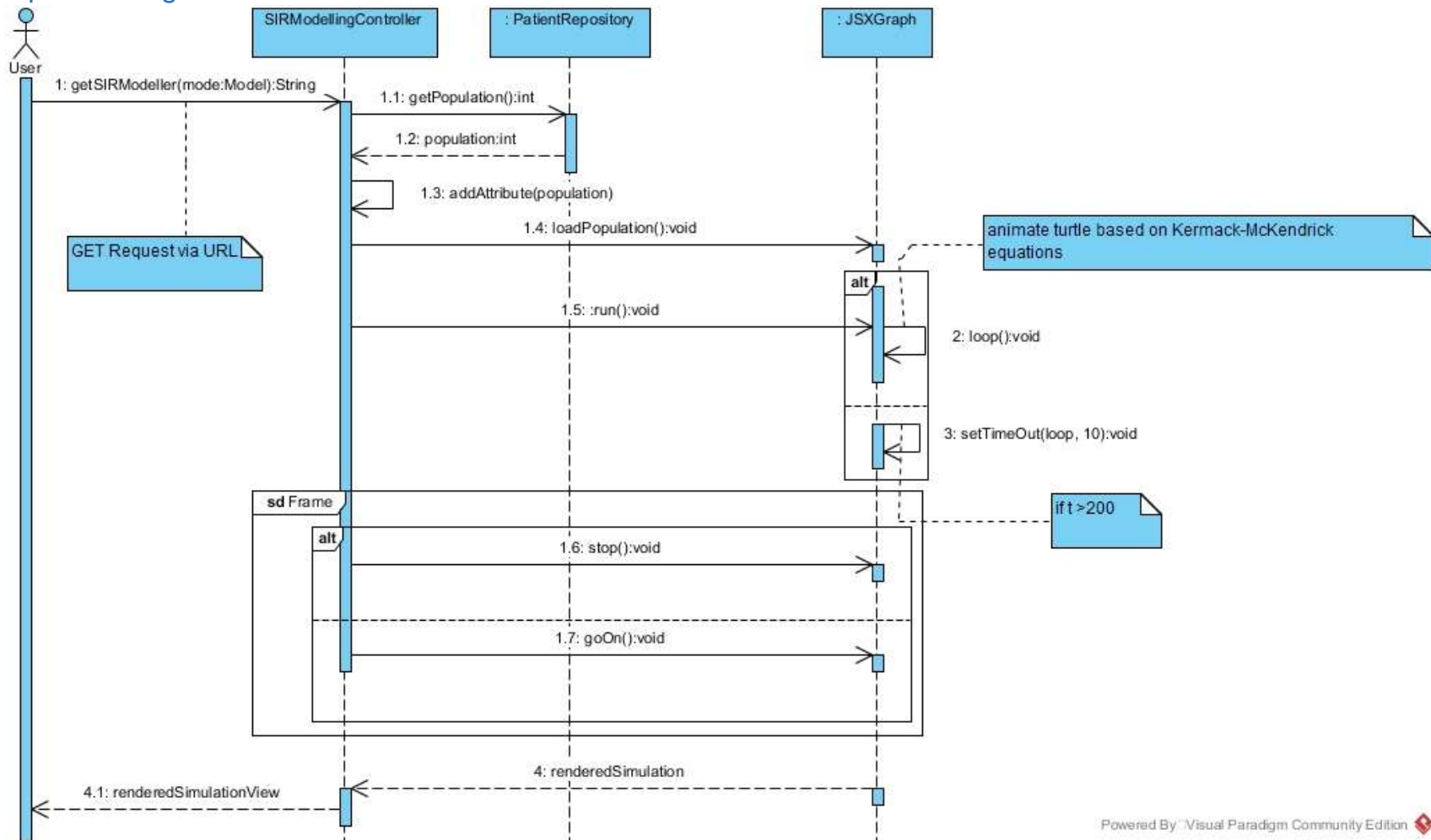
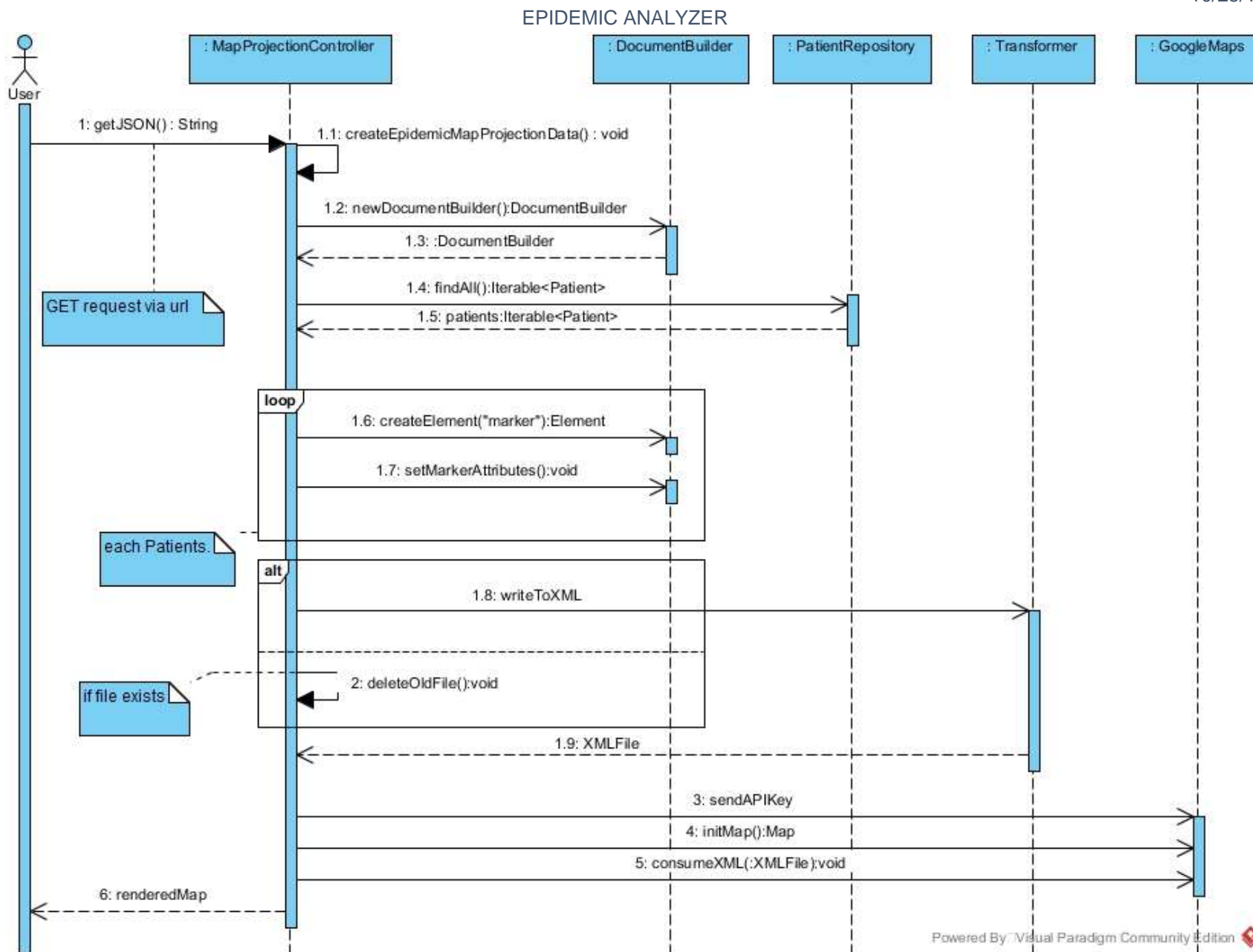


Figure 8: Sequence Diagram - SIR Modelling

EPIDEMIC ANALYZER

Sequence diagram is a type of interaction diagram used to model dynamic behavior of a system. Sequence diagram depicts communication and message passing between object in sequence of time. Typically, it captures the behavior of one scenario.
My.safaribooksonline.com, 2017)



EPIDEMIC ANALYZER

Figure 9: Sequence Diagram: Map Generation

EPIDEMIC ANALYZER

Activity diagram

Activity diagrams are UML standardized, graphical representation of algorithm or in general the workflow of activities and actions with the attributes of decisions, concurrency support, and iteration (go to like implementation). ([Brugge et al, 2014](#))

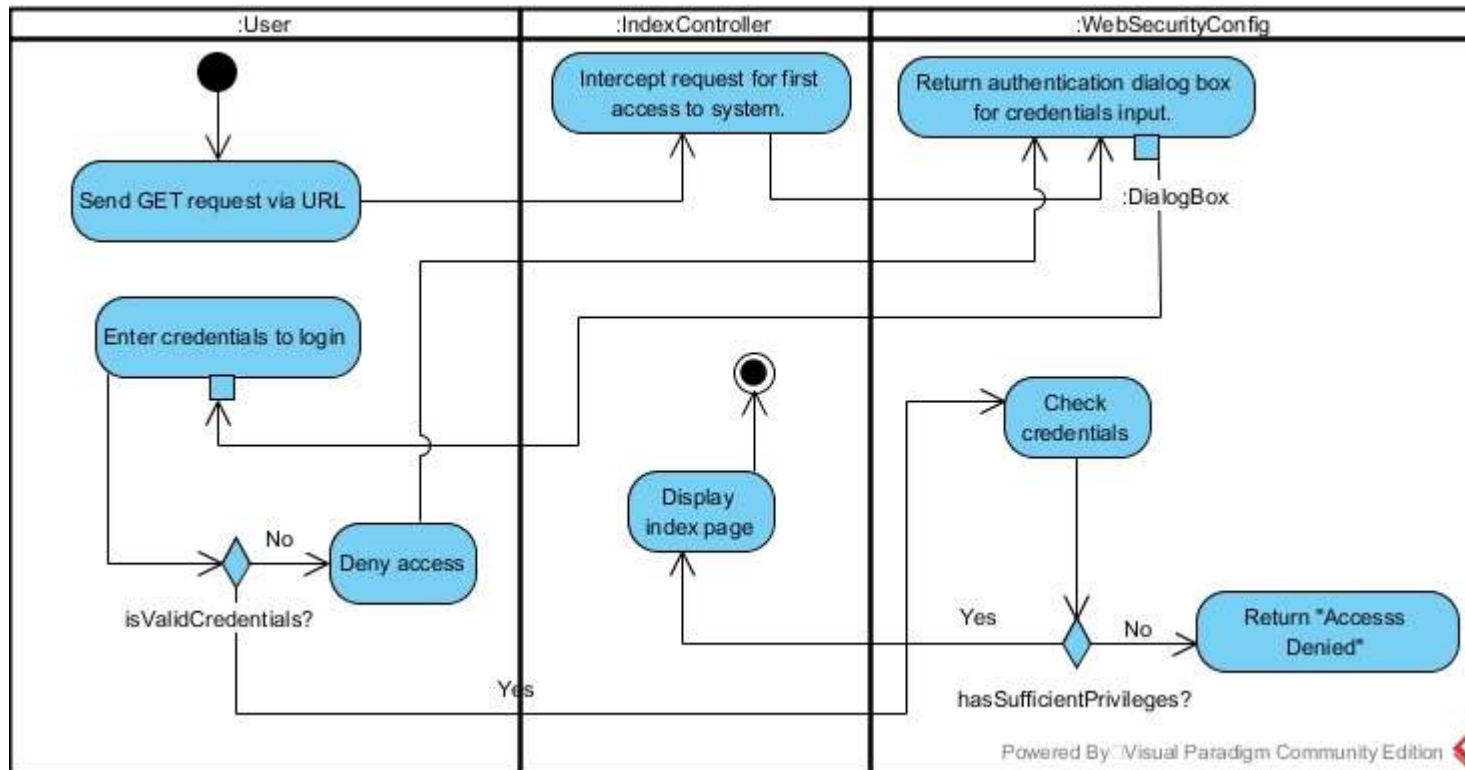
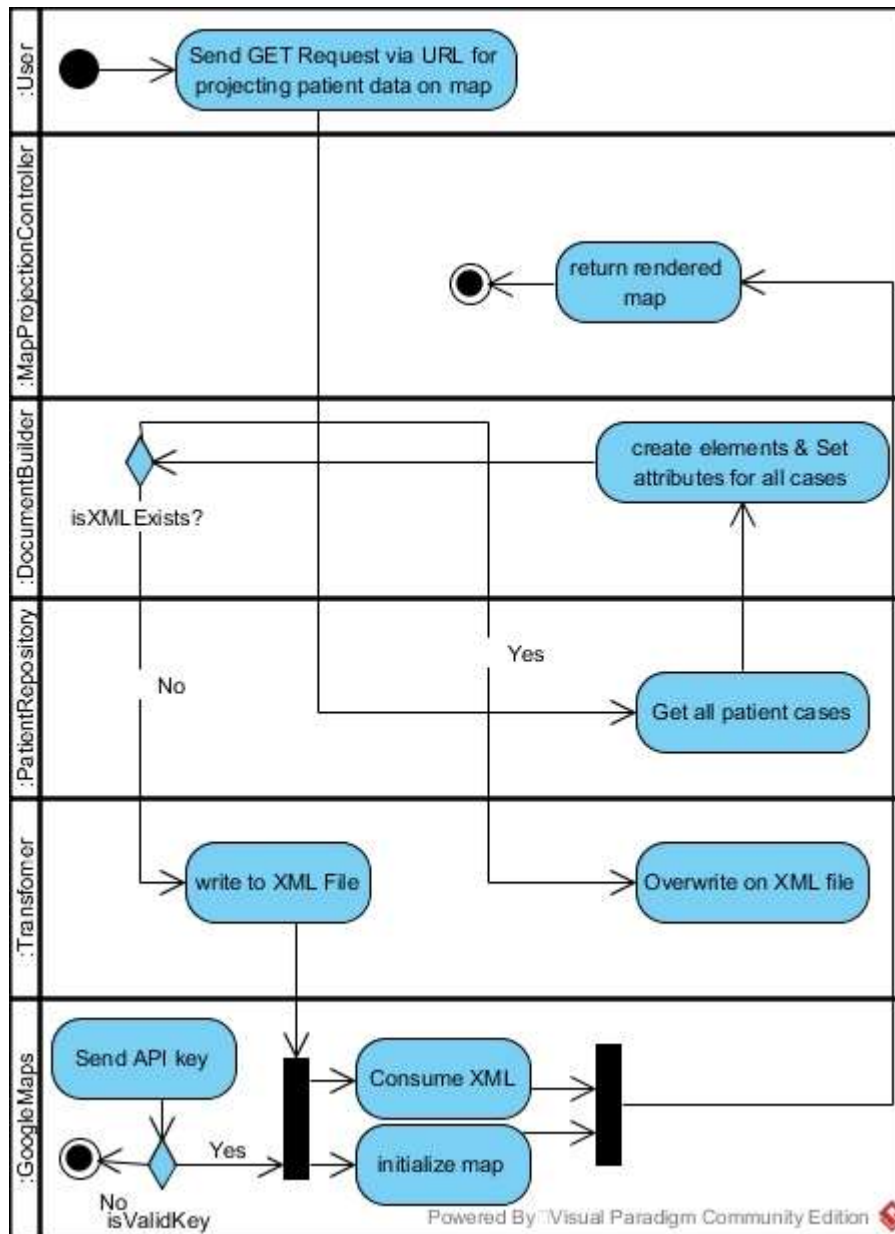


Figure 10 Activity Diagram: Authentication

EPIDEMIC ANALYZER



EPIDEMIC ANALYZER

Figure 11: Activity Diagram Map Generation

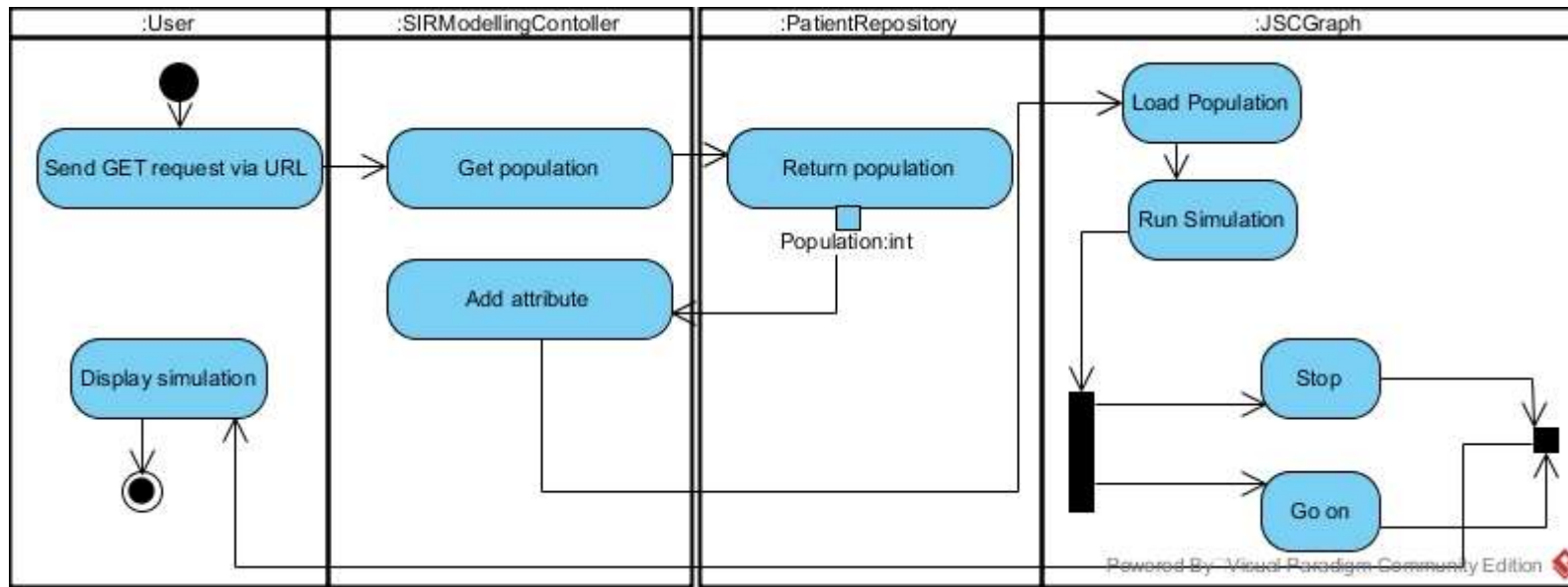


Figure 12: Activity Diagram - SIR Modelling

EPIDEMIC ANALYZER

Structural modelling

Architecture Level class diagram

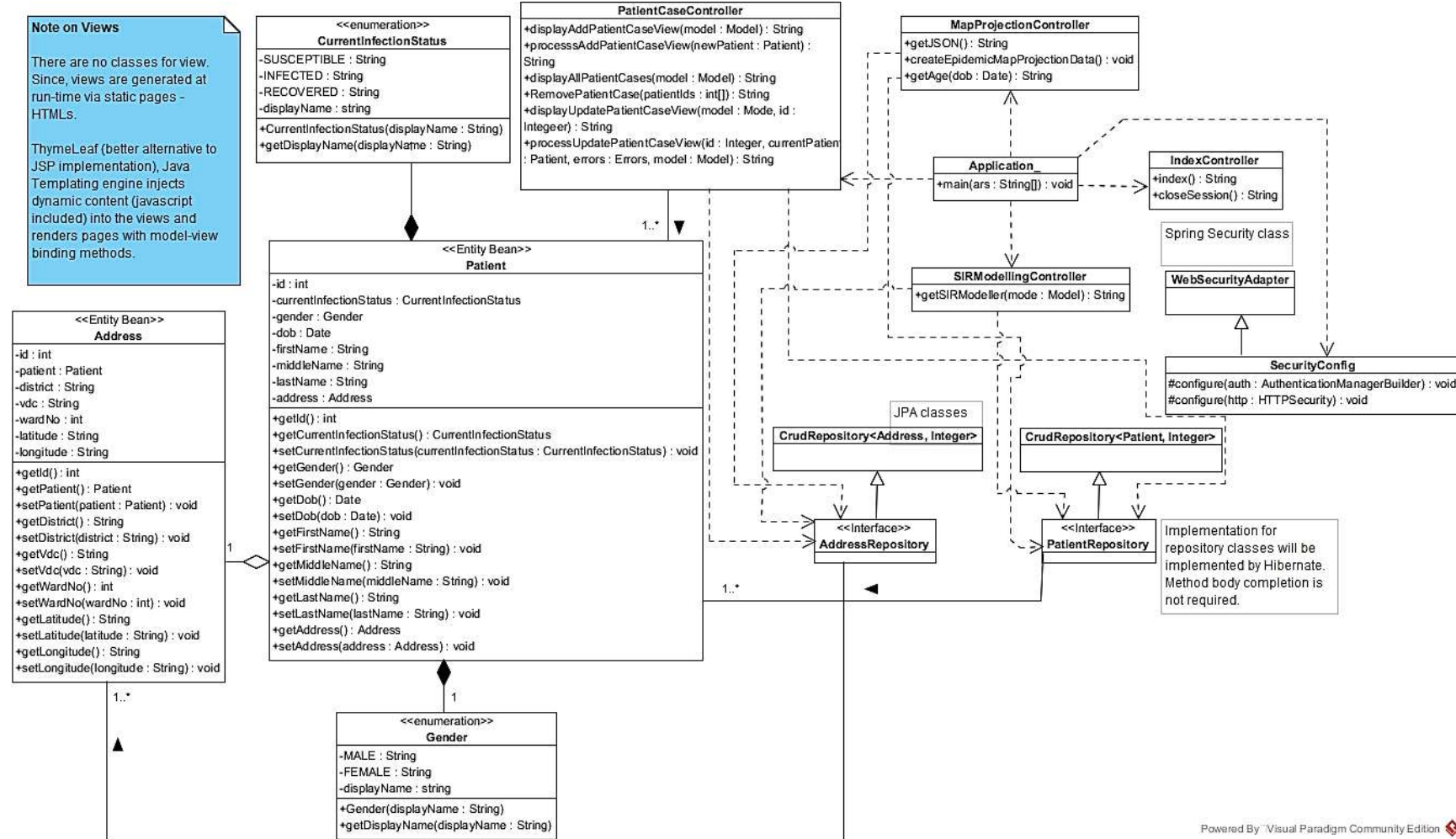


Figure 13: Application architecture

EPIDEMIC ANALYZER

The architecture diagram is an extension of domain level class diagram where implementations are more detailed and supportive classes are introduced. Additionally, design patterns can be visible in this level of class diagram. All the layers of presentation, logic and data access are included. ([Brugge et al, 2014](#))

In the diagram shown above data persistence and access is done by repository classes. Model has been annotated as entity beans that will persist in DB at runtime. MapProjectionController uses JSON of patients generated at run time.

CHAPTER 4

IMPLEMENTATION

Programming Language used

- **Java 9**

It is Used for back-end programming. This the main language that the system is based on. The framework Spring Boot is used in conjunction with other frameworks like Spring Security, Spring MVC, Thymeleaf, Spring Test, and others. The core programming used in Spring Boot is Java.

It handles all servlet requests and maps them to specific handlers in the model controllers. The communication between model and view via model-form binding and results display including all other actions are performed on the structure created in Java.

- **ECMAScript 2017 (JavaScript)**

It is used for front-end programming particularly to generate the SIR model as discussed below. And it is also used to provide sorting, filtration, and searching features on table data that displays all the existing patient cases. Likewise, it is used for front-end interactions such as in navigation dropdowns and alert boxes.

Kermack-McKendrick Model

KMM is a non-spatial temporal SIR (Susceptible, Infected, Recovered) model that represents impact of contagious infection in a closed population over time. Its assumptions are as follows. ([Weisstein, 2017](#); [Anderson and May, 1979](#))

- Epidemic population domain is closed i.e. no birth or death occurs during epidemic.
- Population is homoge

The model compartmentalizes population into 3 groups – susceptible, infected, and recovered. Their relations and rate are shown with 3 non-linear ordinary differential equations. ([Jones and Sleeman, 2003, p.112](#))

$$\frac{dS}{dt} = -\beta SI \quad (1)$$

$$\frac{dI}{dt} = \beta SI - \gamma I \quad (2)$$

$$\frac{dR}{dt} = \gamma I, \quad (3)$$

Where t is time, S is the susceptible population, I is the currently infected, R is the recovered, β is the susceptible to infected rate, γ is the infected to recovered rate.

Model-Project Integration

- I have employed KMM SIR model as described above, into my project by using these equations in constructing the SIR model and thus simulating epidemic effect-propagation over a period on a population.
- To paint the graph, I used 'JSXGraph', a dynamic mathematics visualization library based purely on javascript. ()

System cutover from development architecture to the implementation

[See Code snippets]

CHAPTER 5

TESTING

EPIDEMIC ANALYZER

Unit Test

Patient Case CRUD operations

Test Suite#	UT1
Test Class	PatientCaseController
Test method	displayAddPatientCaseView() :String
Test Data	GET ("/patient_case/add_patient_case")
Expected outcome	Patient case registration interface is rendered in browser
Actual outcome	Patient case registration interface is rendered in browser
Test scripts with result	(view in appendix)

Test Suite#	UT2
Test Class	PatientCaseController
Test method	processAddPatientCaseView() :String
Test Data	POST ("/patient_case/add_patient_case"); newPatient();
Expected outcome	New patient is persisted to DB
Actual outcome	New patient is persisted to DB
Test scripts with result	(view in appendix)

Test Suite#	UT3
Test Class	PatientCaseController
Test method	displayAllPatientCasees() :String
Test Data	GET ("/patient_case/display_patient_cases")
Expected outcome	All patient cases are returned with CRUD options on each patient record.
Actual outcome	All patient cases are returned with CRUD options on each patient record.
Test scripts with result	(view in appendix)

Test Suite#	UT4
Test Class	PatientCaseController
Test method	removePatientCase(): String
Test Data	POST("patient_case/update_patient_case/{id}")
Expected outcome	Selected patient case is removed.
Actual outcome	Selected patient case is removed.
Test scripts	(view in appendix)

Test Suite#	UT5
Test Class	PatientCaseController
Test method	updatePatientCaseView() :String
Test Data	POST("patient_case/display_patient_cases")
Expected outcome	Selected patient case is updated.
Actual outcome	Selected patient case is updated.

EPIDEMIC ANALYZER

Test scripts	(view in appendix)
---------------------	------------------------------------

Patient case data projection on map

Test Suite#	UT6
Test Class	MapProjectionController
Test method	getJSON() : String
Test Data	GET("patient_case/project_to_map")
Expected outcome	Patient case record is parsed to JSON format & written to file
Actual outcome	Patient case record is parsed to JSON format & written to file
Test scripts with result	(view in appendix)

Test Suite#	UT7
Test Class	MapProjectionController
Test method	getAge(dob: Date): String
Test Data	GET("patient_case/project_to_map"); Date dob;
Expected outcome	Current age of patient is returned from date object
Actual outcome	Current age of patient is returned from date object
Test scripts with result	(view in appendix)

Test Suite#	UT8
Test Class	MapProjectionController
Test method	createEpidemicMapProjectionData():void
Test Data	GET("patient_case/project_to_map")
Expected outcome	Map is rendered on view based on patient cases
Actual outcome	Map is rendered on view based on patient cases
Test scripts with result	(view in appendix)

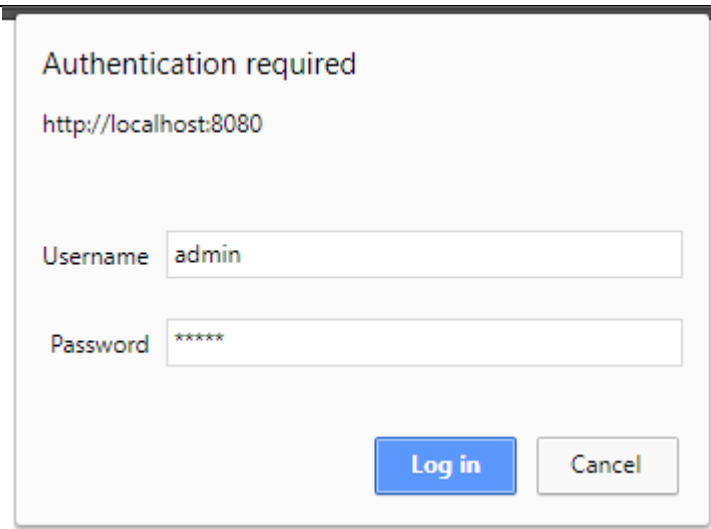

SIR Modelling

Test Suite#	UT9
Test Class	MapProjectionController
Test method	getJSON() : String
Test Data	GET("patient_case/project_to_map")
Expected outcome	Patient case record is parsed to JSON format & written to file
Actual outcome	Patient case record is parsed to JSON format & written to file
Test scripts with result	(view in appendix)

Test scripts with result	(view in appendix)
--------------------------	--------------------

Integration Testing

Each individual unit in the application i.e. the methods of main controllers were tested above. To test the communication between the units and how application functions cohesively a walkthrough of the whole system was conducted and the test result was documented below.

Functionality	Walkthrough screenshot	Test Result
Login		Login was successful.
Patient case addition		Case addition was successful.

EPIDEMIC ANALYZER

Displaying all patient cases	<div><div>EPIDEMIC ANALYZER</div><div>PATIENT CASE ▾</div><div>SIR MODELLING</div><div>USER MANUAL</div><div>LOGOUT</div></div> <div><div>WELCOME!</div><div><div>ADD NEW PATIENT CASE</div><div>DISPLAY ALL PATIENT CASES</div><div>PROJECT CASES ON MAP</div></div><div><p>In the milieu of a humanitarian crisis (e.g., epidemic), responders including public health personnel require epidemic data and its analysis. Such information is paramount for decision makers whose discernments and actions have direct impact on the outcome of the crisis.</p></div><div>EPIDEMIC ANALYZER</div></div> <div>PATIENT CASE ENTRIES</div> <div><div>SHOW 10 ▾ ENTRIES</div><div>Search...</div></div> <table><tr><th>PATIENT ID</th><th>CURRENT INFECTION STATUS</th><th>FIRST NAME</th><th>MIDDLE NAME</th><th>LAST NAME</th><th>DATE OF BIRTH</th><th>GENDER</th><th>DISTRICT</th><th>VDC</th><th>WAR NO</th></tr><tr><td>13</td><td>RECOVERED</td><td>BDKISHA</td><td>S</td><td>SHRESTHA</td><td>1990-10-24</td><td>FEMALE</td><td>KATHMANDU</td><td>KAPAN</td><td>2</td></tr><tr><td>14</td><td>SUSCEPTIBLE</td><td>PAUL</td><td>C</td><td>COPAN</td><td>1979-10-12</td><td>MALE</td><td>DOLAKHA</td><td>JIRI</td><td>2</td></tr><tr><td>15</td><td>SUSCEPTIBLE</td><td>PABIN</td><td>MORELAND</td><td>ROLAND</td><td>1988-10-03</td><td>MALE</td><td>ASD</td><td>ASD</td><td>1</td></tr><tr><td>16</td><td>INFECTED</td><td>RADHA</td><td>KRISHNAN</td><td>ROSHAN</td><td>1974-10-03</td><td>FEMALE</td><td>ASD</td><td>ASDASD</td><td>3</td></tr><tr><td>17</td><td>RECOVERED</td><td>MANJUSHREE</td><td>THAPA</td><td>SHRESTHA</td><td>1986-11-01</td><td>FEMALE</td><td>[KIKI]</td><td>[]</td><td>4</td></tr><tr><td>18</td><td>SUSCEPTIBLE</td><td>PAUL</td><td>C</td><td>COPAN</td><td>2000-10-04</td><td>MALE</td><td>KATHMANDU</td><td>DHAPASI</td><td>9</td></tr><tr><td>19</td><td>INFECTED</td><td>JOHN</td><td>L</td><td>LENNOX</td><td>1980-10-04</td><td>MALE</td><td>DHADING</td><td>SALYANTAR</td><td>6</td></tr><tr><td>20</td><td>INFECTED</td><td>MARY</td><td>JO</td><td>SHARJ</td><td>1990-10-13</td><td>FEMALE</td><td>BHAKTAPUR</td><td>LOKANTHALI</td><td>8</td></tr><tr><td>21</td><td>INFECTED</td><td>FRANK</td><td>T</td><td>TUREK</td><td>1993-10-11</td><td>MALE</td><td>KABHREPALANCHOK</td><td>WALTING</td><td>3</td></tr><tr><td>22</td><td>INFECTED</td><td>JAMES</td><td>PORTER</td><td>MORELAND</td><td>1979-10-15</td><td>MALE</td><td>LALITPUR</td><td>SANKHU</td><td>1</td></tr></table> <div><div>SHOWING 1 TO 10 OF 14 ENTRIES</div><div>PREVIOUSNEXT</div></div>	PATIENT ID	CURRENT INFECTION STATUS	FIRST NAME	MIDDLE NAME	LAST NAME	DATE OF BIRTH	GENDER	DISTRICT	VDC	WAR NO	13	RECOVERED	BDKISHA	S	SHRESTHA	1990-10-24	FEMALE	KATHMANDU	KAPAN	2	14	SUSCEPTIBLE	PAUL	C	COPAN	1979-10-12	MALE	DOLAKHA	JIRI	2	15	SUSCEPTIBLE	PABIN	MORELAND	ROLAND	1988-10-03	MALE	ASD	ASD	1	16	INFECTED	RADHA	KRISHNAN	ROSHAN	1974-10-03	FEMALE	ASD	ASDASD	3	17	RECOVERED	MANJUSHREE	THAPA	SHRESTHA	1986-11-01	FEMALE	[KIKI]	[]	4	18	SUSCEPTIBLE	PAUL	C	COPAN	2000-10-04	MALE	KATHMANDU	DHAPASI	9	19	INFECTED	JOHN	L	LENNOX	1980-10-04	MALE	DHADING	SALYANTAR	6	20	INFECTED	MARY	JO	SHARJ	1990-10-13	FEMALE	BHAKTAPUR	LOKANTHALI	8	21	INFECTED	FRANK	T	TUREK	1993-10-11	MALE	KABHREPALANCHOK	WALTING	3	22	INFECTED	JAMES	PORTER	MORELAND	1979-10-15	MALE	LALITPUR	SANKHU	1	Model and simulation was rendered correctly.
PATIENT ID	CURRENT INFECTION STATUS	FIRST NAME	MIDDLE NAME	LAST NAME	DATE OF BIRTH	GENDER	DISTRICT	VDC	WAR NO																																																																																																							
13	RECOVERED	BDKISHA	S	SHRESTHA	1990-10-24	FEMALE	KATHMANDU	KAPAN	2																																																																																																							
14	SUSCEPTIBLE	PAUL	C	COPAN	1979-10-12	MALE	DOLAKHA	JIRI	2																																																																																																							
15	SUSCEPTIBLE	PABIN	MORELAND	ROLAND	1988-10-03	MALE	ASD	ASD	1																																																																																																							
16	INFECTED	RADHA	KRISHNAN	ROSHAN	1974-10-03	FEMALE	ASD	ASDASD	3																																																																																																							
17	RECOVERED	MANJUSHREE	THAPA	SHRESTHA	1986-11-01	FEMALE	[KIKI]	[]	4																																																																																																							
18	SUSCEPTIBLE	PAUL	C	COPAN	2000-10-04	MALE	KATHMANDU	DHAPASI	9																																																																																																							
19	INFECTED	JOHN	L	LENNOX	1980-10-04	MALE	DHADING	SALYANTAR	6																																																																																																							
20	INFECTED	MARY	JO	SHARJ	1990-10-13	FEMALE	BHAKTAPUR	LOKANTHALI	8																																																																																																							
21	INFECTED	FRANK	T	TUREK	1993-10-11	MALE	KABHREPALANCHOK	WALTING	3																																																																																																							
22	INFECTED	JAMES	PORTER	MORELAND	1979-10-15	MALE	LALITPUR	SANKHU	1																																																																																																							
Searching and sorting results	<div><div>EPIDEMIC ANALYZER</div><div>PATIENT CASE ▾</div><div>SIR MODELLING</div><div>USER MANUAL</div><div>LOGOUT</div></div> <div>PATIENT CASE ENTRIES</div> <div><div>SHOW 10 ▾ ENTRIES</div><div>Paul</div></div> <table><tr><th>PATIENT ID</th><th>CURRENT INFECTION STATUS</th><th>FIRST NAME</th><th>MIDDLE NAME</th><th>LAST NAME</th><th>DATE OF BIRTH</th><th>GENDER</th><th>DISTRICT</th><th>VDC</th><th>WAR NO</th></tr><tr><td>14</td><td>SUSCEPTIBLE</td><td>PAUL</td><td>C</td><td>COPAN</td><td>1979-10-12</td><td>MALE</td><td>DOLAKHA</td><td>JIRI</td><td>2</td></tr><tr><td>18</td><td>SUSCEPTIBLE</td><td>PAUL</td><td>C</td><td>COPAN</td><td>2000-10-04</td><td>MALE</td><td>KATHMANDU</td><td>DHAPASI</td><td>9</td></tr></table> <div><div>SHOWING 1 TO 2 OF 2 ENTRIES (FILTERED FROM 14 TOTAL ENTRIES)</div><div>PREVIOUSNEXT</div></div> <div>DELETE SELECTED</div>	PATIENT ID	CURRENT INFECTION STATUS	FIRST NAME	MIDDLE NAME	LAST NAME	DATE OF BIRTH	GENDER	DISTRICT	VDC	WAR NO	14	SUSCEPTIBLE	PAUL	C	COPAN	1979-10-12	MALE	DOLAKHA	JIRI	2	18	SUSCEPTIBLE	PAUL	C	COPAN	2000-10-04	MALE	KATHMANDU	DHAPASI	9	Searching and sorting functioned correctly.																																																																																
PATIENT ID	CURRENT INFECTION STATUS	FIRST NAME	MIDDLE NAME	LAST NAME	DATE OF BIRTH	GENDER	DISTRICT	VDC	WAR NO																																																																																																							
14	SUSCEPTIBLE	PAUL	C	COPAN	1979-10-12	MALE	DOLAKHA	JIRI	2																																																																																																							
18	SUSCEPTIBLE	PAUL	C	COPAN	2000-10-04	MALE	KATHMANDU	DHAPASI	9																																																																																																							

EPIDEMIC ANALYZER

Deleting
patient
case

The screenshot displays the 'Epidemic Analyzer' application. At the top, there is a table with columns: 'WARD NO', 'LAT.', 'LONG.', and 'SELECT'. The first row is highlighted in light blue and contains the values '2', '-33.840282', '151.207474', and a checked checkbox. The second row is highlighted in light green and contains the values '2', '-33.910751', '151.194168', and an unchecked checkbox. To the right of the table, there are two green 'UPDATE' buttons. Below the table, a red bar indicates 'SHOWING 1 TO 10 OF 14 ENTRIES'. Below that, there are links for 'PREVIOUS' and 'NEXT', and a red button labeled 'DELETE SELECTED'. In the foreground, a white dialog box with a close button (X) is open. It contains the text 'localhost:8080 says:' and 'Confirm deletion?'. At the bottom of the dialog box are two buttons: 'OK' (blue) and 'Cancel' (white).

WARD NO	LAT.	LONG.	SELECT
2	-33.840282	151.207474	<input checked="" type="checkbox"/>
2	-33.910751	151.194168	<input type="checkbox"/>

SHOWING 1 TO 10 OF 14 ENTRIES
PREVIOUS NEXT



DELETE SELECTED

localhost:8080 says:
Confirm deletion?

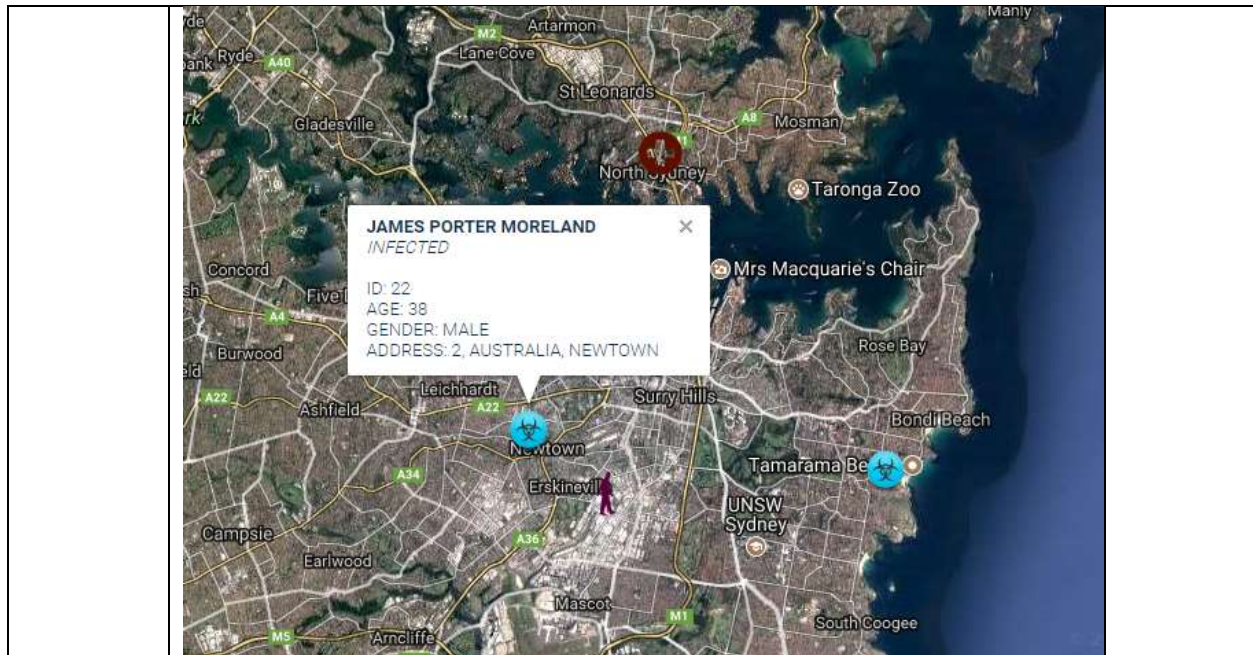
OK Cancel

Patient
case
was
deleted
successf
ully.

EPIDEMIC ANALYZER

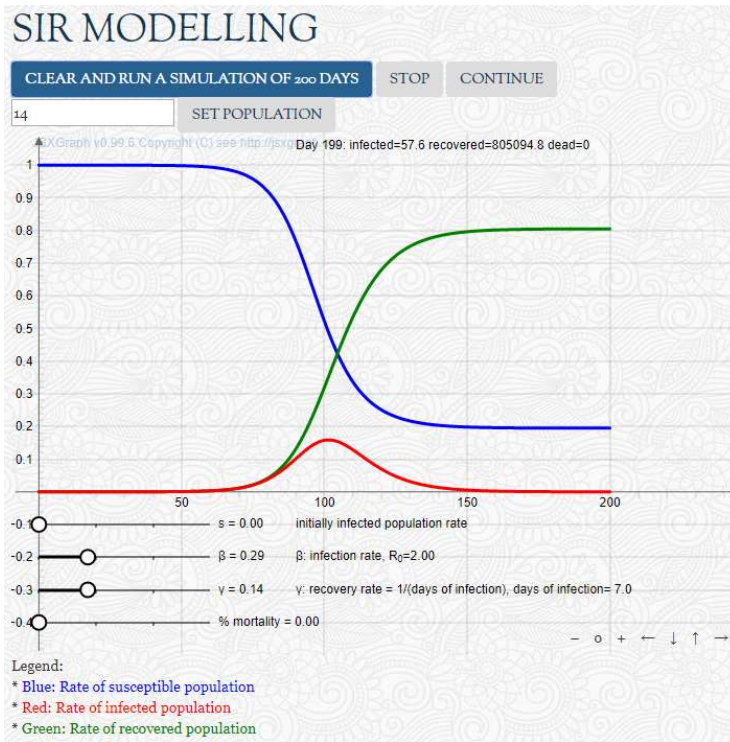
<p>Updating patient case</p>	 <p>The screenshot shows the 'UPDATE PATIENT CASE' interface. At the top, there's a table with columns: WARD NO, LAT., LONG., and SELECT. It lists two patients, both with 'WARD NO' 2. The first patient has coordinates -33.840282, 151.207474 and a checked 'SELECT' box. The second has coordinates -33.910751, 151.194168 and an unchecked 'SELECT' box. Green 'UPDATE' buttons are next to each row. Below this is the 'UPDATE PATIENT CASE' form with input fields for Name (William), Lane, Craig, Rasuwa, Langtang, a dropdown menu with '2', GPS Latitude, GPS Longitude, a date field (10/11/1971), a gender dropdown (Male), and an infection status dropdown (Infected). A green 'UPDATE CASE' button is at the bottom.</p>	<p>Update functioned as expected.</p>
<p>Project cases on map</p>	 <p>The screenshot shows the 'PROJECT CASES ON MAP' page. It features a dark header with navigation links: EPIDEMIC ANALYZER, PATIENT CASE, SIR MODELLING, USER MANUAL, and LOGOUT. The main content area has a 'WELCOME!' message, a list of links (ADD NEW PATIENT CASE, DISPLAY ALL PATIENT CASES, PROJECT CASES ON MAP), and a brief description of the application's purpose. A green 'PROJECT CASES ON MAP' button is highlighted.</p>	<p>Map was rendered correctly with along with patient data.</p>

EPIDEMIC ANALYZER



EPIDEMIC ANALYZER

SIR
modelling



Model
and
simulatio
n
redered
correctly,
as
expected
.

CHAPTER 6

OTHER PROJECT ISSUES

Risk Management

The universe looms with ubiquitous uncertainties and the unwanted event can interrupt the flow and execution of any other events at any instance. The solution is to be best prepared and anticipate potential risks early on while having some kind of plan to mitigate or navigate through such situations. This is the philosophy of risk management.

Just as project management involves the development of a project plan and control of the project using that plan as the project progresses, risk management involves the identification of risk. Risk management involves identification of risks at the project's outset and the control of those risks as the project unfolds. ([Dawson, 2015](#))

Four steps to risk management are:

1. Identify risks
2. Assess impact of risk
3. Alleviate critical risk
4. Control risk

Likelihood	Value
Low	1
Medium	2
High	3

Figure 14: Risk Likelihood values ([Turner, 1999](#))

Consequence	Value
Very Low	1
Low	2
Medium	3
High	4
Very High	5

Figure 15: Risk consequence values ([Turner, 1999](#))

Taking reference from the above tables, to estimate the impact of each identified risks we use,
 $\text{Impact} = \text{Likelihood} \times \text{Consequence}$

EPIDEMIC ANALYZER

Based on Turner's ([Turner, 1999](#)) quantitative measure for assessing risk, & based on the guidelines for conducting such assessment provided by Dawson ([Dawson, 2015](#)), I've presented below, the risk management matrix for my project.

Risk Type	Risk	Likelihood	Consequence	Impact	Action Type	Action
Non technical	Scope Creep – Uncontrolled & incremental growth of project's scope at any point of project causing cost or time overrun. (Knight, Thomas, and Angus, 2013)	1	5	5	Avoidance	Follow requirements specifications from analysis phase. Remind oneself of impending deadlines using reminder applications and wall calendars.
	Gold Plating – It refers to intentionally adding extra features or functions to the products which were not included in the scope statement. (Usmani, n.d.)	1	4	4	Avoidance	Follow requirements specifications from analysis phase. Remind oneself of impending deadlines reminder applications and wall calendars.
	Natural Disaster	1	5	5	Contingency	Regularly backup entire project directory in Google drive.
	Deadline overruns	2	5	10	Contingency	Remind oneself of impending deadlines reminder applications and wall calendars. Commit to earlier deadline over actual stipulated deadlines.
Technical	Hardware failure	2	4	8	Contingency & avoidance	Perform system diagnosis and hardware condition checks on the local machine at the service center. Apply upgrades and fixes if necessary. Use power surge protection. Regularly backup project to google drive. Keep a backup machine in the vicinity on standby.
	Hard disk failure	1	5	5	Contingency &	Clone the local machine's hard drive once before initiating the project.

EPIDEMIC ANALYZER

					avoidance	Regularly backup project on google drive.
	Technical difficulties (ignorance of algorithm, sticky bugs, problems with modeling) during design and implementation.	3	4	12	Deflection	Get expert advice from supervisor and faculty members. Research on the web and contact industry experts if required.
	Steep algorithmic complexity to achieve certain functionality.	3	4	12	Deflection	Get expert advice from supervisor and faculty members. Research on the web for APIs, and libraries to perform such tasks.

Configuration Management

All project artifacts reside in two locations – the local machine, where all the files are created and the GitHub repository (<https://github.com/BijuAle/>) under the main folder titled – CP Project. Both locations are in sync using the Git Bash.

The directory is based on major phases of SDLC. Each phase has a folder and all documents, code base, and modeling diagrams reside in respective folders. The 'Project Management' folder contains all the research materials and planning documents such as research literature, proposal, and schedules. There is a folder 'Backups' where the entire project is backed up and saved in sub-folder named by date and time of the time of backup. This will help me rollback changes made at particular instances of significant commit points. Finally, the whole project structure is backed up in Google Drive regularly.

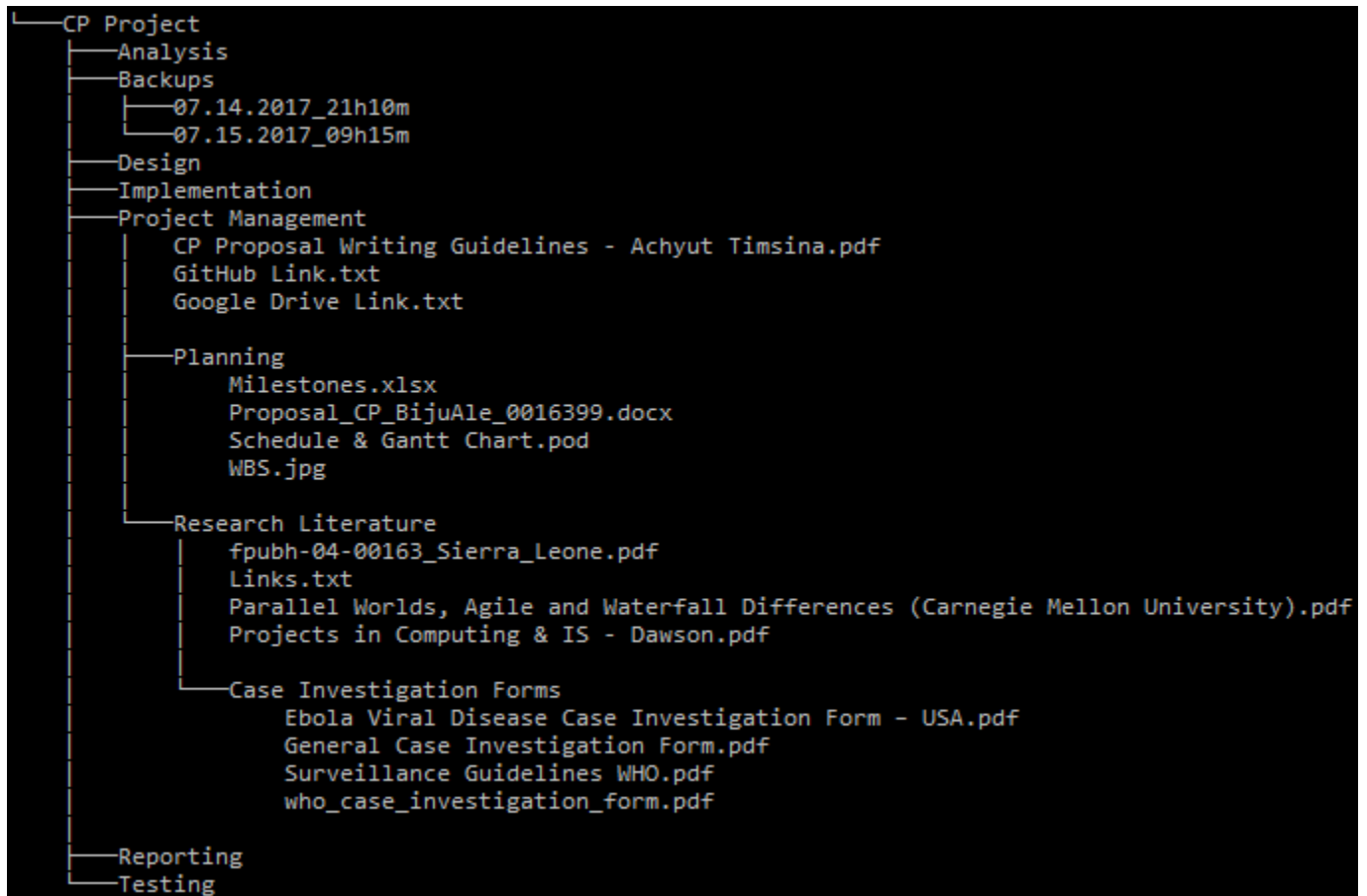


Figure 16: Directory Structure for Project in the local machine.

CHAPTER 7

CONCLUSION

EPIDEMIC ANALYZER

This report has sought to document all activities carried out in each of the phases of software development lifecycle. Firstly, a background of the system was provided stating the context for the project, where it would be deployed and how it would benefit humanitarian actors during epidemic scenarios. The aims and objectives were listed and an overview was given for the design along with a rich picture for the project. These were the contents explained in chapter 1.

Chapter 2 was opened with an introduction to analysis and the importance of analysis for better understanding of requirements and to reach transform it into proof of concept in modelling phase. The choice of analysis methodology was stated as object oriented analysis and the reasons for opting it over other techniques were expounded. Moreover, activities involved in analysis were described using the artefacts diagram in UML. The system's functional and non-functional requirements were identified along with the rational and justifications with MoSCoW prioritization. Then, all use cases of the system were presented with a use case diagram and scenario description for each of the cases. Lastly, the system's initial class diagram was shown to give an overview of domain level architecture.

In Chapter 3, the design phase was documented with dynamic and static models presented and explained. Dynamic model consisted of sequence diagram and activity diagram to show the object lifecycle and messaging at run time and to show the flow of execution respectively. Architecture level class diagram showed design pattern and supportive classes like the configuration and repository classes.

Chapter 4 focused on how the model was translated into actual implementation. The code snippets were referenced to the appendix section. It also explained how user manual would be integrated into the system and its importance for usability.

Chapter 5 documented the test report for the project. Unit testing with black box and white box testing and integration testing were done on each key methods of classes and their functionality was verified and validated.

In chapter 6, risk management, configuration management and change management was explained under other project issues. The references and appendix containing all test results, code snippets, and user manual were kept under chapter 8 and 9 respectively.

CHAPTER 8

REFERENCES & BIBLIOGRAPHY

References

Jones, D. and Sleeman, B. (2003). *Differential equations and mathematical biology*. 1st ed. Boca Raton, Fla.: Chapman and Hall/CRC.

Weisstein, Eric W. (2017). *Kermack-McKendrick Model -- from Wolfram MathWorld*. [online] Mathworld.wolfram.com.

Available at: <http://mathworld.wolfram.com/Kermack-McKendrickModel.html>

[Accessed 12 Aug. 2017].

Anderson, R.M. and May, R.M., 1979. Population biology of infectious diseases: Part I. *Nature*, 280 (5721), pp.361-367.

Available from: Research on Complex Systems, Humboldt University, Germany online resource- http://rocs.hu-berlin.de/complex_sys_2015/resources/Seminarpapers/Anderson_May_1979.pdf

[Accessed 12 Aug. 2017].

Larman, Craig (2012). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3/e. Pearson Education India.

Bruegge, B. and Dutoit, A. (2014). *Object-oriented software engineering*. 1st ed. Harlow: Pearson.

Data.humdata.org. (2016). Shelter Cluster Nepal Agency Data Tracker & Reports - Humanitarian Data Exchange. [online] Available at: <https://data.humdata.org/dataset/scnepal-agency-data> [Accessed 23 Aug. 2017].

Goo.gl. (2016). WASH Cluster. [online] Available at: <https://goo.gl/wh9wUI> [Accessed 23 Aug. 2017].

PMC NCBI. (2014). *Ebola Virus Disease in West Africa — The First 9 Months of the Epidemic and Forward Projections*. [online] Available at:

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4235004/>

[Accessed 16 Aug. 2017].

cdc.gov. (2016). *2014 Ebola Outbreak in West Africa - Case Counts | Ebola Hemorrhagic Fever | CDC*. [online] Available at: <https://www.cdc.gov/vhf/ebola/outbreaks/2014-west-africa/case-counts.html>

[Accessed 16 Aug. 2017].

Turner, J. (1999). *The handbook of project-based management*. London: McGraw-Hill.

Schach, S. (2011). *Object-oriented and classical software engineering*. 1st ed. Boston [u.a.]: McGraw-Hill.

Dawson, C. (2015). *Projects in computing and information systems*. Harlow: Pearson.

My.safaribooksonline.com. (2017). *UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition > Sequence Diagrams: Safari Books Online*. [online]

Available at: <http://my.safaribooksonline.com/book/software-engineering-and-development/uml/0321193687/sequence-diagrams/ch04>

[Accessed 03 Oct. 2017].

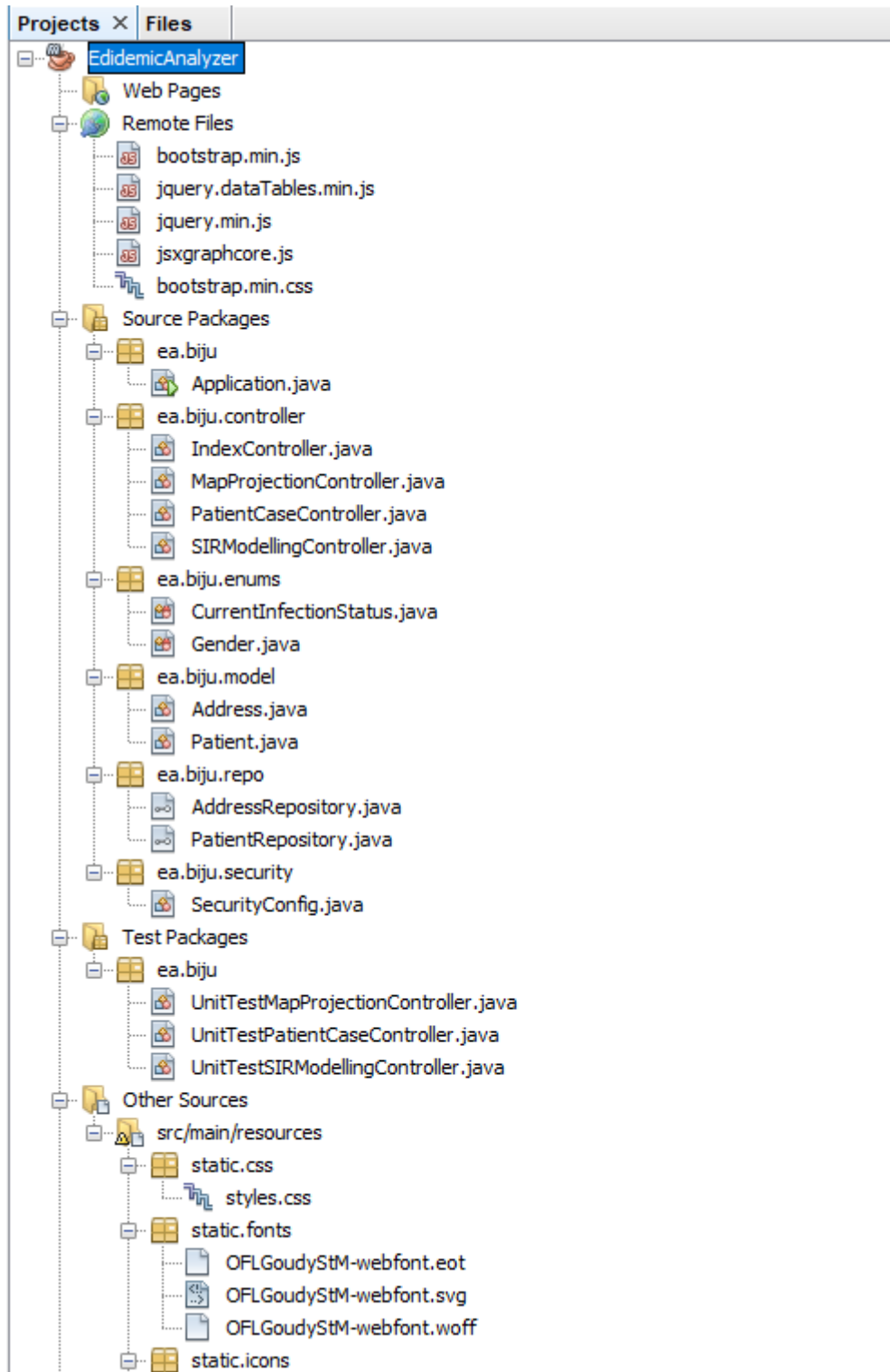
CHAPTER 9

APPENDIX

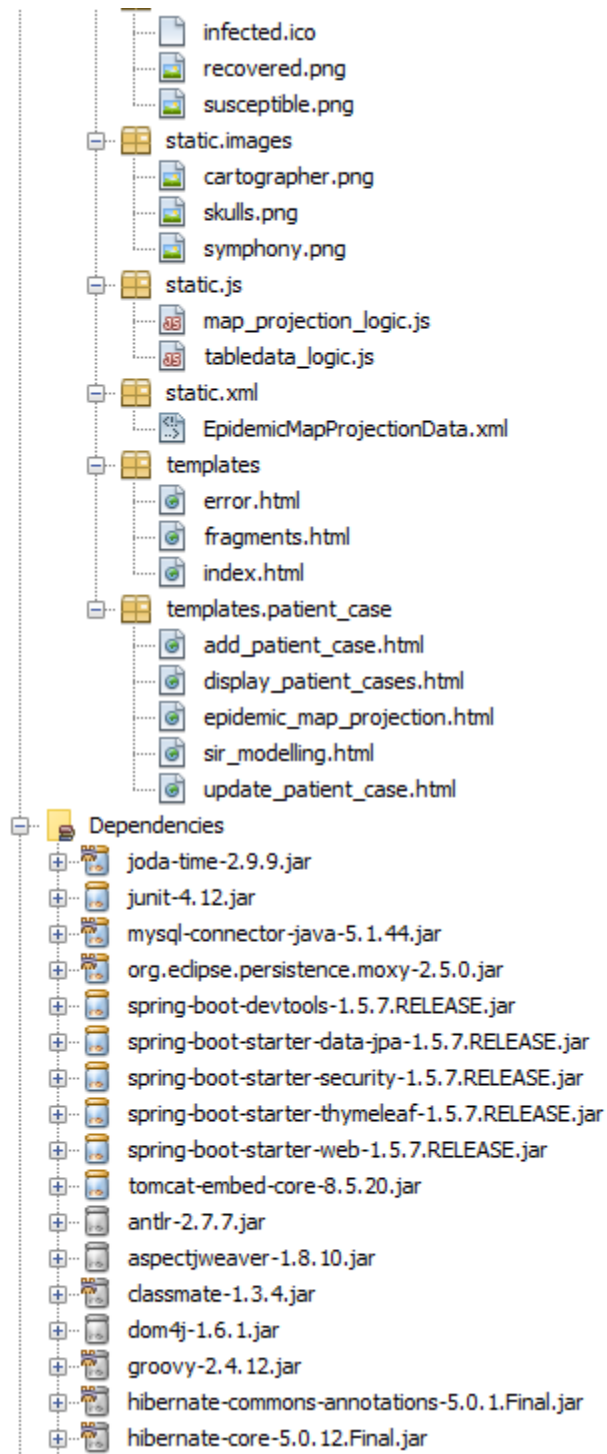
Code Snippets

Source code directory structure
(on next page)










































EPIDEMIC ANALYZER



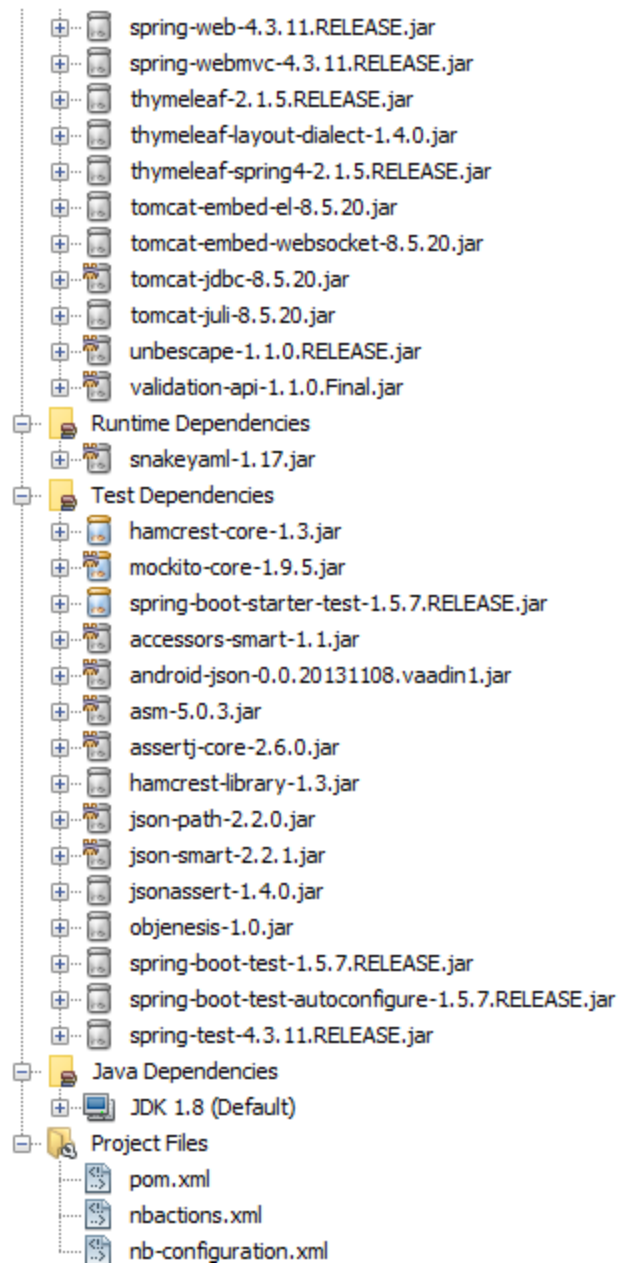
EPIDEMIC ANALYZER



EPIDEMIC ANALYZER

- +  hibernate-entitymanager-5.0.12.Final.jar
- +  hibernate-jpa-2.1-api-1.0.0.Final.jar
- +  hibernate-validator-5.3.5.Final.jar
- +  jackson-annotations-2.8.0.jar
- +  jackson-core-2.8.10.jar
- +  jackson-databind-2.8.10.jar
- +  jandex-2.0.0.Final.jar
- +  javassist-3.21.0-GA.jar
- +  javax.transaction-api-1.2.jar
- +  jboss-logging-3.3.1.Final.jar
- +  jcl-over-slf4j-1.7.25.jar
- +  jul-to-slf4j-1.7.25.jar
- +  log4j-over-slf4j-1.7.25.jar
- +  logback-classic-1.1.11.jar
- +  logback-core-1.1.11.jar
- +  ognl-3.0.8.jar
- +  org.eclipse.persistence.antlr-2.5.0.jar
- +  org.eclipse.persistence.asm-2.5.0.jar
- +  org.eclipse.persistence.core-2.5.0.jar
- +  slf4j-api-1.7.25.jar
- +  spring-aop-4.3.11.RELEASE.jar
- +  spring-aspects-4.3.11.RELEASE.jar
- +  spring-beans-4.3.11.RELEASE.jar
- +  spring-boot-1.5.7.RELEASE.jar
- +  spring-boot-autoconfigure-1.5.7.RELEASE.jar
- +  spring-boot-starter-1.5.7.RELEASE.jar
- +  spring-boot-starter-aop-1.5.7.RELEASE.jar
- +  spring-boot-starter-jdbc-1.5.7.RELEASE.jar
- +  spring-boot-starter-logging-1.5.7.RELEASE.jar
- +  spring-boot-starter-tomcat-1.5.7.RELEASE.jar
- +  spring-context-4.3.11.RELEASE.jar
- +  spring-core-4.3.11.RELEASE.jar
- +  spring-data-commons-1.13.7.RELEASE.jar
- +  spring-data-jpa-1.11.7.RELEASE.jar
- +  spring-expression-4.3.11.RELEASE.jar
- +  spring-jdbc-4.3.11.RELEASE.jar
- +  spring-orm-4.3.11.RELEASE.jar
- +  spring-security-config-4.2.3.RELEASE.jar
- +  spring-security-core-4.2.3.RELEASE.jar
- +  spring-security-web-4.2.3.RELEASE.jar
- +  spring-tx-4.3.11.RELEASE.jar

EPIDEMIC ANALYZER



ea.biju.Application.java

```
6   package ea.biju;
7
8   /**
9    *
10   * @author Biju Ale
11   */
12   import org.springframework.boot.SpringApplication;
13   import org.springframework.boot.autoconfigure.SpringBootApplication;↵
14
15   import org.springframework.boot.autoconfigure.domain.EntityScan;
16   import org.springframework.context.ApplicationContext;
17   import org.springframework.data.jpa.repository.config.↵
18   EnableJpaRepositories;
19
20   @SpringBootApplication(scanBasePackages = {"ea.biju"})
21   @EnableJpaRepositories
22   @EntityScan
23   public class Application {
24
25       public static void main(String[] args) {
26           ApplicationContext applicationContext = SpringApplication.↵
27           run(Application.class, args);
28       }
29   }
```

EPIDEMIC ANALYZER

ea.biju.controller.IndexController java

```

6   package ea.biju.controller;
7
8   import javax.servlet.http.HttpServletRequest;
9   import org.springframework.security.core.context.↵
    SecurityContextHolder;
10  import org.springframework.stereotype.Controller;
11  import org.springframework.web.bind.annotation.RequestMapping;
12
13  /**
14   *
15   * @author Biju Ale
16   */
17  @Controller
18  public class IndexController {
19
20      @RequestMapping(value = "")
21      public String index() {
22          return "index";
23      }
24
25      @RequestMapping(value = "logout")
26      public String closeSession(HttpServletRequest request) {
27          SecurityContextHolder.clearContext();
28          return "index";
29      }
30  }
31

```


EPIDEMIC ANALYZER

```

6 package ea.biju.controller;
7
8 import ea.biju.model.Patient;
9 import ea.biju.repo.PatientRepository;
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.web.bind.annotation.RequestMapping;
12
13 import java.io.File;
14 import java.util.Calendar;
15 import java.util.Date;
16 import javax.xml.parsers.DocumentBuilder;
17 import javax.xml.parsers.DocumentBuilderFactory;
18 import javax.xml.parsers.ParserConfigurationException;
19 import javax.xml.transform.Transformer;
20 import javax.xml.transform.TransformerException;
21 import javax.xml.transform.TransformerFactory;
22 import javax.xml.transform.dom.DOMSource;
23 import javax.xml.transform.stream.StreamResult;
24 import org.joda.time.Years;
25 import org.joda.time.LocalDate;
26 import org.springframework.stereotype.Controller;
27
28 import org.w3c.dom.Document;
29 import org.w3c.dom.Element;
30
31 /**
32  *
33  * @author Biju Ale
34  */
35 @Controller
36 @RequestMapping("patient_case")
37 public class MapProjectionController {
38
39     @Autowired
40     PatientRepository patientRepository;
41
42     @RequestMapping(value = "project_to_map")
43     public String getJSON() {

```

EPIDEMIC ANALYZER

```

44         createEpidemicMapProjectionData();
45         return "patient_case/epidemic_map_projection";
46     }
47
48     private void createEpidemicMapProjectionData() {
49         try {
50
51             DocumentBuilderFactory docFactory =
DocumentBuilderFactory.newInstance();
52             DocumentBuilder docBuilder = docFactory.
newDocumentBuilder();
53
54             // Create 1 root element - parent node = 'markers'
55             Document doc = docBuilder.newDocument();
56             Element rootElement = doc.createElement("markers");
57             doc.appendChild(rootElement);
58
59             //Create many marker element - child node = 'marker'
60             Iterable<Patient> resultSet = patientRepository.findAll
();
61             for (Patient p : resultSet) {
62                 Element marker = doc.createElement("marker");
63                 rootElement.appendChild(marker);
64
65                 marker.setAttribute("id", Integer.toString(p.getId
()));
66                 marker.setAttribute("name", p.getFirstName() + " "
+ p.getMiddleName() + " " + p.getLastName());
67                 marker.setAttribute("age", getAge(p.getDob()));
68                 marker.setAttribute("gender", p.getGender().
getDisplayNames());
69                 marker.setAttribute("address", p.getAddress().
getWardNo() + ", " + p.getAddress().getVdc() + ", " + p.getAddress().
getDistrict());
70                 marker.setAttribute("lat", p.getAddress().
getLatitude());
71                 marker.setAttribute("lng", p.getAddress().
getLongitude());

```

EPIDEMIC ANALYZER

```

72         marker.setAttribute("currentInfectionStatus", p.↵
getCurrentInfectionStatus().getDisplayName());
73     }
74
75     // write the content into xml file
76     TransformerFactory transformerFactory = ↵
TransformerFactory.newInstance();
77     Transformer transformer = transformerFactory.↵
newTransformer();
78     DOMSource source = new DOMSource(doc);
79
80     String filePathString = "src\\main\\resources\\static\\↵
xml\\EpidemicMapProjectionData.xml";
81     File xmlFile = new File(filePathString);
82     if (xmlFile.exists() && !xmlFile.isDirectory()) {
83         xmlFile.delete();
84     }
85     StreamResult result = new StreamResult(xmlFile);
86
87     // Output to console for testing
88     // StreamResult result = new StreamResult(System.out);
89     transformer.transform(source, result);
90     System.out.println("File saved!");
91     } catch (ParserConfigurationException pce) {
92         pce.printStackTrace();
93     } catch (TransformerException tfe) {
94         tfe.printStackTrace();
95     }
96 }
97
98 private String getAge(Date dob) {
99     Calendar cal = Calendar.getInstance();
100     cal.setTime(dob);
101     LocalDate birthdate = new LocalDate(cal.get(Calendar.YEAR), ↵
cal.get(Calendar.MONTH), cal.get(Calendar.DAY_OF_MONTH));
102     LocalDate now = new LocalDate();
103     String age = Integer.toString(Years.yearsBetween(birthdate, ↵
now).getYears());
104     return age;
105 }
106
107 }
108

```


EPIDEMIC ANALYZER

```

6 package ea.biju.controller;
7
8 import ea.biju.enums.CurrentInfectionStatus;
9 import ea.biju.enums.Gender;
10 import ea.biju.model.Patient;
11 import ea.biju.repo.AddressRepository;
12 import ea.biju.repo.PatientRepository;
13 import javax.validation.Valid;
14 import org.springframework.beans.factory.annotation.Autowired;
15 import org.springframework.stereotype.Controller;
16 import org.springframework.ui.Model;
17 import org.springframework.validation.Errors;
18 import org.springframework.web.bind.annotation.ModelAttribute;
19 import org.springframework.web.bind.annotation.PathVariable;
20 import org.springframework.web.bind.annotation.RequestMapping;
21 import org.springframework.web.bind.annotation.RequestMethod;
22 import org.springframework.web.bind.annotation.RequestParam;
23
24 /**
25  *
26  * @author Biju Ale
27  */
28 @Controller
29 @RequestMapping("patient_case")
30 public class PatientCaseController {
31
32     @Autowired
33     private PatientRepository patientRepository;
34
35     @Autowired
36     private AddressRepository addressRepository;
37
38     @RequestMapping(value = "add_patient_case", method = RequestMethod.GET)
39     public String displayAddPatientCaseView(Model model) {
40         model.addAttribute(new Patient());
41         model.addAttribute("genderList", Gender.values());
42         model.addAttribute("currentInfectionStatusList",

```

EPIDEMIC ANALYZER

```

43     CurrentInfectionStatus.values());
44     return "patient_case/add_patient_case";
45 }
46
47     @RequestMapping(value = "add_patient_case", method = RequestMethod.POST)
48     public String processAddPatientCaseView(@ModelAttribute @Valid Patient newPatient, Errors errors, Model model) {
49         if (errors.hasErrors()) {
50             model.addAttribute("genderList", Gender.values());
51             model.addAttribute("currentInfectionStatusList", CurrentInfectionStatus.values());
52             return "patient_case/add_patient_case";
53         }
54         patientRepository.save(newPatient); //persist logic here
55         return "redirect:add_patient_case";
56     }
57
58     @RequestMapping(value = "display_patient_cases", method = RequestMethod.GET)
59     public String displayAllPatientCases(Model model) {
60         model.addAttribute("patients", patientRepository.findAll());
61         return "patient_case/display_patient_cases";
62     }
63
64     @RequestMapping(value = "display_patient_cases", method = RequestMethod.POST)
65     public String removePatientCase(@RequestParam(required = false) int[] patientIds) {
66         if (patientIds == null) {
67             return "redirect:display_patient_cases";
68         }
69         for (int patientId : patientIds) {
70             patientRepository.delete(patientId);
71         }
72         return "redirect:display_patient_cases";
73     }

```

EPIDEMIC ANALYZER

```

73
74     @RequestMapping(value = "update_patient_case/{id}", method = ↵
RequestMapping.GET)
75     public String displayUpdatePatientCaseView(Model model, @↵
PathVariable(value = "id") Integer id) {
76         model.addAttribute("patient", patientRepository.findOne(id) ↵
);
77         model.addAttribute("genderList", Gender.values());
78         model.addAttribute("currentInfectionStatusList", ↵
CurrentInfectionStatus.values());
79         return "patient_case/update_patient_case";
80     }
81
82     @RequestMapping(value = "update_patient_case/{id}", method = ↵
RequestMethod.POST)
83     public String processUpdatePatientCaseView(
84         @PathVariable(value = "id") Integer id,
85         @ModelAttribute
86         @Valid Patient currentPatient,
87         Errors errors,
88         Model model) {
89         if (errors.hasErrors()) {
90             model.addAttribute("genderList", Gender.values());
91             model.addAttribute("currentInfectionStatusList", ↵
CurrentInfectionStatus.values());
92             return "patient_case/update_patient_case";
93         }
94         Patient updatedPatient = patientRepository.findOne(id);
95         updatedPatient.setFirstName(currentPatient.getFirstName());
96         updatedPatient.setMiddleName(currentPatient.getMiddleName() ↵
);
97         updatedPatient.setLastName(currentPatient.getLastName());
98         updatedPatient.setGender(currentPatient.getGender());
99         updatedPatient.setDob(currentPatient.getDob());
100        updatedPatient.setCurrentInfectionStatus(currentPatient.↵
getCurrentInfectionStatus());
101        updatedPatient.setAddress(currentPatient.getAddress());
102
103        patientRepository.save(updatedPatient);
104        return "redirect:../display_patient_cases";
105    }
106 }
107

```


EPIDEMIC ANALYZER

ea.biju.controller.SIRModellingController.java

```

6  package ea.biju.controller;
7
8  import ea.biju.repo.PatientRepository;
9  import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.stereotype.Controller;
11 import org.springframework.ui.Model;
12 import org.springframework.web.bind.annotation.RequestMapping;
13 import org.springframework.web.bind.annotation.RequestMethod;
14
15 /**
16  *
17  * @author Biju Ale
18  */
19 @Controller
20 @RequestMapping(value = "patient_case")
21 public class SIRModellingController {
22
23     @Autowired
24     PatientRepository patientRepository;
25
26     @RequestMapping(value = "sir_modelling", method = RequestMethod.↵
27     GET)
28     public String getSIRModeller(Model model) {
29         model.addAttribute("population", patientRepository.↵
30         getPopulation());
31         return "patient_case/sir_modelling";
32     }
33 }

```

ea.biju.enums.CurrentInfectionStatus.java

```
6      package ea.biju.enums;
7
8      /**
9       *
10      * @author Biju Ale
11      */
12      public enum CurrentInfectionStatus {
13          SUSCEPTIBLE("Susceptible"),
14          INFECTED("Infected"),
15          RECOVERED("Recovered");
16
17          private final String displayName;
18
19          CurrentInfectionStatus(String displayname) {
20              this.displayName = displayname;
21          }
22
23          public String getDisplayName() {
24              return this.displayName;
25          }
26      }
27
```

ea.biju.enums.Gender.java

```
6 package ea.biju.enums;
7
8 /**
9  *
10  * @author Biju Ale
11  */
12 public enum Gender {
13     MALE("Male"),
14     FEMALE("Female");
15
16     private final String displayName;
17
18     Gender(String displayname) {
19         this.displayName = displayname;
20     }
21
22     public String getDisplayName() {
23         return this.displayName;
24     }
25 }
26
```

ea.biju.model.Address.java

```

6  package ea.biju.model;
7
8  import javax.persistence.Column;
9  import javax.persistence.Entity;
10 import javax.persistence.GeneratedValue;
11 import javax.persistence.Id;
12 import javax.persistence.JoinColumn;
13 import javax.persistence.OneToOne;
14 import javax.persistence.Table;
15 import javax.validation.constraints.Max;
16 import javax.validation.constraints.Min;
17 import javax.validation.constraints.NotNull;
18 import javax.validation.constraints.Pattern;
19 import javax.validation.constraints.Size;
20 import javax.xml.bind.annotation.XmlAccessType;
21 import javax.xml.bind.annotation.XmlAccessorType;
22 import javax.xml.bind.annotation.XmlElement;
23 import javax.xml.bind.annotation.XmlRootElement;
24 import org.eclipse.persistence.xml.annotations.XmlInverseReference;
25
26 /**
27  *
28  * @author Biju Ale
29  */
30 @Entity
31 @Table(name = "tbl_address")
32 @XmlRootElement
33 @XmlAccessorType(XmlAccessType.FIELD)
34 public class Address {
35
36     @Id
37     @GeneratedValue
38     private int id;
39
40     @Column
41     @NotNull
42     @Size(min = 1, message = "District cannot be null")
43     private String district;

```

EPIDEMIC ANALYZER

```

44
45     @Column
46     @NotNull
47     @Size(min = 1, message = "VDC cannot be null")
48     private String vdc;
49
50     @Column
51     @NotNull
52     @Min(1)
53     @Max(9)
54     private int wardNo;
55
56     @Column(precision = 10, scale = 6)
57     @Pattern(regexp = "([+-]?\\d+\\.?\\d+)",
58         message = "GPS Coordinates contains positive or ↵
negative floating points only.")
59     private String latitude;
60
61     @Column(precision = 10, scale = 6)
62     @Pattern(regexp = "([+-]?\\d+\\.?\\d+)",
63         message = "GPS Coordinates contains positive or ↵
negative floating points only.")
64     private String longitude;
65
66     @OneToOne
67     @XmlElement
68     @XmlInverseReference(mappedBy = "address")
69     @JoinColumn(name = "address_id")
70     private Patient patient;
71
72     public String getDistrict() {
73         return district;
74     }
75
76     public void setDistrict(String district) {
77         this.district = district;

```

EPIDEMIC ANALYZER

```

78  L      }
79
80  [      public String getVdc() {
81  |          return vdc;
82  |      }
83
84  [      public void setVdc(String vdc) {
85  |          this.vdc = vdc;
86  |      }
87
88  [      public int getWardNo() {
89  |          return wardNo;
90  |      }
91
92  [      public void setWardNo(int wardNo) {
93  |          this.wardNo = wardNo;
94  |      }
95
96  [      public int getId() {
97  |          return id;
98  |      }
99
100 [      public String getLatitude() {
101 |          return latitude;
102 |      }
103
104 [      public void setLatitude(String latitude) {
105 |          this.latitude = latitude;
106 |      }
107
108 [      public String getLongitude() {
109 |          return longitude;
110 |      }
111
112 [      public void setLongitude(String longitude) {
113 |          this.longitude = longitude;
114 |      }
115
116 [      public Patient getPatient() {
117 |          return patient;
118 |      }
119
120 [      public void setPatient(Patient patient) {
121 |          this.patient = patient;
122 |      }
123
124      }
125

```

ea.biju.model.Patient.java

```
6 package ea.biju.model;
7
8 import ea.biju.enums.CurrentInfectionStatus;
9 import ea.biju.enums.Gender;
10 import java.util.Date;
11 import javax.persistence.CascadeType;
12 import javax.persistence.Column;
13 import javax.persistence.Entity;
14 import javax.persistence.EnumType;
15 import javax.persistence.Enumerated;
16 import javax.persistence.GeneratedValue;
17 import javax.persistence.Id;
18 import javax.persistence.OneToOne;
19 import javax.persistence.Table;
20 import javax.persistence.Temporal;
21 import javax.persistence.TemporalType;
22 import javax.validation.constraints.NotNull;
23 import javax.validation.constraints.Pattern;
24 import javax.validation.constraints.Size;
25 import javax.xml.bind.annotation.XmlAccessType;
26 import javax.xml.bind.annotation.XmlAccessorType;
27 import javax.xml.bind.annotation.XmlElement;
28 import javax.xml.bind.annotation.XmlRootElement;
29 import org.eclipse.persistence.oxm.annotations.XmlInverseReference;
30 import org.springframework.format.annotation.DateTimeFormat;
31
32 /**
33  *
34  * @author Biju Ale
35  */
36 @Entity
37 @Table(name = "tbl_patient")
38 @XmlRootElement
39 @XmlAccessorType(XmlAccessType.FIELD)
40 public class Patient {
41
42     @Id
43     @GeneratedValue
```

EPIDEMIC ANALYZER

```
44     private int id;
45
46     @Column
47     @NotNull
48     @Size(min = 1, max = 20)
49     @Pattern(message = "Only alphabets, dots, and spaces allowed.",
50             regexp = "^[A-Za-z\\s]{1,}[\\.]{0,1}[A-Za-z\\s]{0,}$")
51     private String firstName;
52
53     @Column
54     @NotNull
55     @Size(min = 1, max = 20)
56     @Pattern(message = "Only alphabets, dots, and spaces allowed.",
57             regexp = "^[A-Za-z\\s]{1,}[\\.]{0,1}[A-Za-z\\s]{0,}$")
58     private String middleName;
59
60     @Column
61     @NotNull
62     @Size(min = 1, max = 20)
63     @Pattern(message = "Only alphabets, dots, and spaces allowed.",
64             regexp = "^[A-Za-z\\s]{1,}[\\.]{0,1}[A-Za-z\\s]{0,}$")
65     private String lastName;
66
67     @OneToOne(cascade = CascadeType.ALL)
68     @XmlElement
69     @XmlInverseReference(mappedBy = "patient")
70     private Address address;
71
72     @Column
73     @NotNull
74     @DateTimeFormat(pattern = "yyyy-MM-dd")
75     @Temporal(TemporalType.DATE)
76     private Date dob;
77
78     @Column
79     @Enumerated(EnumType.STRING)
80     private Gender gender;
81
```


EPIDEMIC ANALYZER

```

82     @Column
83     @Enumerated(EnumType.STRING)
84     private CurrentInfectionStatus currentInfectionStatus;
85
86     public int getId() {
87         return id;
88     }
89
90     public String getFirstName() {
91         return firstName;
92     }
93
94     public void setFirstName(String firstName) {
95         this.firstName = firstName;
96     }
97
98     public String getMiddleName() {
99         return middleName;
100    }
101
102    public void setMiddleName(String middleName) {
103        this.middleName = middleName;
104    }
105
106    public String getLastName() {
107        return lastName;
108    }
109
110    public void setLastName(String lastName) {
111        this.lastName = lastName;
112    }
113
114    public Address getAddress() {
115        return address;
116    }
117
118    public void setAddress(Address address) {
119        this.address = address;

```

EPIDEMIC ANALYZER

```

120     }
121
122     public Date getDob() {
123         return dob;
124     }
125
126     public void setDob(Date dob) {
127         this.dob = dob;
128     }
129
130     public Gender getGender() {
131         return gender;
132     }
133
134     public void setGender(Gender gender) {
135         this.gender = gender;
136     }
137
138     public CurrentInfectionStatus getCurrentInfectionStatus() {
139         return currentInfectionStatus;
140     }
141
142     public void setCurrentInfectionStatus(CurrentInfectionStatus ←
currentInfectionStatus) {
143         this.currentInfectionStatus = currentInfectionStatus;
144     }
145
146 }
147

```

ea.biju.repo.AddressRepository.java

```

6     package ea.biju.repo;
7
8     import ea.biju.model.Address;
9     import javax.transaction.Transactional;
10    import org.springframework.data.repository.CrudRepository;
11    import org.springframework.stereotype.Repository;
12
13    /**
14     *
15     * @author Biju Ale
16     */
17    @Repository
18    @Transactional
19    public interface AddressRepository extends CrudRepository<Address, ←
Integer> {
20
21    }

```

EPIDEMIC ANALYZER

Ea.biju.repo.PatientRepository.java

```

6   package ea.biju.repo;
7
8   import ea.biju.model.Patient;
9   import javax.transaction.Transactional;
10  import org.springframework.data.jpa.repository.Query;
11  import org.springframework.data.repository.CrudRepository;
12  import org.springframework.stereotype.Repository;
13
14  /**
15   *
16   * @author Biju Ale
17   */
18  @Repository
19  @Transactional
20  public interface PatientRepository extends CrudRepository<Patient, Integer> {
21
22      @Query("SELECT count(id) FROM Patient")
23      public int getPopulation();
24  }
25

```

EPIDEMIC ANALYZER

ea.biju.security.SecurityConfig.java

```

6  package ea.biju.security;
7
8  import org.springframework.context.annotation.Configuration;
9  import org.springframework.security.config.annotation.authentication↵
    .builders.AuthenticationManagerBuilder;
10 import org.springframework.security.config.annotation.web.builders.↵
    HttpSecurity;
11 import org.springframework.security.config.annotation.web.↵
    configuration.WebSecurityConfigurerAdapter;
12
13 /**
14  *
15  * @author Biju Ale
16  */
17 @Configuration
18 public class SecurityConfig extends WebSecurityConfigurerAdapter {
19     // Authentication : User --> Roles
20
21     protected void configure(AuthenticationManagerBuilder auth)
22         throws Exception {
23         auth.inMemoryAuthentication()
24             .withUser("user").password("user").roles("USER").and()
25             .withUser("admin").password("admin").roles("USER", "↵
    ADMIN");
26     }
27
28     @Override
29     // Authorization : Role -> Access
30     protected void configure(HttpSecurity http) throws Exception {
31         http.httpBasic().and().authorizeRequests()
32             .antMatchers("/**").hasRole("ADMIN").and()
33             .csrf().disable().headers().frameOptions().disable();
34     }
35 }
36

```

static.css.styles.css

```
1  /*Importing font*/
2  @font-face {
3      font-family: OFL Sorts Mill Goudy;
4      font-style: normal;
5      font-weight: normal;
6      src: url(../fonts/OFLGoudyStM-webfont.eot) format("embedded-↵
       opentype"),
7          url(../fonts/OFLGoudyStM-webfont.woff) format("woff"),
8          url(../fonts/OFLGoudyStM-webfont.svg) format("svg")
9  }
10
11  /*Basic Styling*/
12  body,nav,h1,h2,h3,legend,button, input[type="submit"],input[type="↵
       button"], input[type="reset"],caption{
13      font-family: "OFL Sorts Mill Goudy";
14      text-transform: uppercase;
15      color:#113654;
16  }
17  .indexContent{
18      font-family:"Georgia";
19      font-size: 12px;
20      text-align: justify;
21      text-transform: none;
22      text-indent: 10px;
23  }
24  p::first-letter {
25      font-size: 200%;
26      color: #8A2BE2;
27  }
28  #ea{
29      color: #8A2BE2;
30  }
31  blockquote{
32      text-align: right;
33  }
34  .InfectedMark{
35      background-color:#FFB2B2;
36  }
37  .RecoveredMark{
38      background-color:#E5E5FF;
39  }
40  .SusceptibleMark{
41      background-color:#DFF0D8;
42  }
43  body{
44      background:url("../images/symphony.png") fixed;
45  }
46  }
```

EPIDEMIC ANALYZER

static.js.map_projection_logic.js

```

1  function downloadUrl(url, callback) {
2      var request = window.ActiveXObject ?
3          new ActiveXObject('Microsoft.XMLHTTP') :
4          new XMLHttpRequest;
5
6      request.onreadystatechange = function () {
7          if (request.readyState == 4) {
8              request.onreadystatechange = doNothing;
9              callback(request, request.status);
10         }
11     };
12
13     request.open('GET', url, true);
14     request.send(null);
15 }
16

```

static.js.tabledata_logic.js

```

1  //Highlight patient case table's rows based on current infection status
2  $(function () {
3      $("tr").each(function () {
4          var col_val = $(this).find("td:eq(1)").text();
5          if (col_val.toUpperCase() === "INFECTED") {
6              $(this).addClass('InfectedMark');
7          } else if (col_val.toUpperCase() === "SUSCEPTIBLE") {
8              $(this).addClass('SusceptibleMark');
9          } else if (col_val.toUpperCase() === "RECOVERED") {
10             $(this).addClass('RecoveredMark');
11         }
12     });
13
14     $('#patientcasetable').DataTable({
15         "lengthMenu": [[10, 25, 50, -1], [10, 25, 50, "All"]],
16         "autoWidth": false,
17         language: {
18             search: "_INPUT_",
19             searchPlaceholder: "Search..."
20         }
21     });
22
23 });
24

```

EPIDEMIC ANALYZER

static.xml.EpidemicMapProjectionData.xml

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?><markers><↵
marker address="2, Kapan, Kathmandu" age="27" currentInfectionStatus=↵
"Recovered" gender="Female" id="13" lat="-33.840282" lng="151.207474"↵
name="Bikisha S Shrestha"/><marker address="2, Jiri, Dolakha" age="↵
39" currentInfectionStatus="Susceptible" gender="Male" id="14" lat="-↵
33.910751" lng="151.194168" name="Paul C Copan"/><marker address="1, ↵
asd, Asd" age="29" currentInfectionStatus="Susceptible" gender="Male"↵
id="15" lat="-33.879917" lng="151.210449" name="Pabin Moreland ↵
Roland"/><marker address="3, asdasd, asd" age="43" ↵
currentInfectionStatus="Infected" gender="Female" id="16" lat="-33.↵
906357" lng="151.263763" name="Radha Krishnan Roshan"/><marker ↵
address="4, jjj, ;klkj" age="51" currentInfectionStatus="Recovered" ↵
gender="Female" id="17" lat="" lng="" name="Manjushree Thapa Shrestha↵
"/><marker address="9, Dhapasi, Kathmandu" age="17" ↵
currentInfectionStatus="Susceptible" gender="Male" id="18" lat="" lng↵
="" name="Paul C Copan"/><marker address="6, Salyantar, Dhading" age↵
"37" currentInfectionStatus="Infected" gender="Male" id="19" lat="" ↵
lng="" name="John L. Lennox"/><marker address="8, Lokanthali, ↵
Bhaktapur" age="26" currentInfectionStatus="Infected" gender="Female"↵
id="20" lat="" lng="" name="Mary Jo Sharp"/><marker address="5, ↵
Walting, Kabhrepalanchok" age="24" currentInfectionStatus="Infected" ↵
gender="Male" id="21" lat="-33.898113" lng="151.174469" name="Frank T↵
Turek"/><marker address="2, Sankhu, Lalitpur" age="38" ↵
currentInfectionStatus="Infected" gender="Male" id="22" lat="-33.↵
898113" lng="151.174469" name="James Porter Moreland"/><marker ↵
address="2, Langtang, Rasuwa" age="46" currentInfectionStatus="↵
Infected" gender="Male" id="23" lat="" lng="" name="William Lane ↵
Craig"/><marker address="1, cccc, cccc" age="18" ↵
currentInfectionStatus="Infected" gender="Female" id="24" lat="" lng↵
="" name="Brian Carroll Patrick"/><marker address="1, cccc, cccc" age="↵
38" currentInfectionStatus="Infected" gender="Female" id="25" lat="" ↵
lng="" name="Bucket Carroll Brian"/><marker address="2, asd, Aasd" age↵
="28" currentInfectionStatus="Recovered" gender="Female" id="27" lat=↵
"" lng="" name="Deepak Thapa Kumar"/></markers>
```

EPIDEMIC ANALYZER

static.templates.error.html

```

1 <!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project
4 Properties.
5 To change this template file, choose Tools | Templates
6 and open the template in the editor.
7 -->
8 <html>
9 <head th:replace="fragments::head">
10 </head>
11 <body class = "container">
12 <nav th:replace="fragments::navigation"></nav>
13 <h2>Access Denied!</h2>
14 </body>
15 </html>
16

```


EPIDEMIC ANALYZER

static.templates.fragments.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org/">
3   <head th:fragment="head">
4     <meta charset="UTF-8"/>
5     <title>Epidemic Analyzer</title>
6     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
7 bootstrap/3.3.7/css/bootstrap.min.css" />
8     <link rel="stylesheet" th:href="@{/css/styles.css}" />
9     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.
10 2.0/jquery.min.js"></script>
11     <script src="https://cdn.datatables.net/1.10.16/js/jquery.
12 datatables.min.js"></script>
13     <script type="text/javascript" src="https://maxcdn.
14 bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
15     <script type="text/javascript" th:src="@{/js/taledata_logic
16 .js}"></script>
17   </head>
18   <body class = "container">
19     <nav class="navbar navbar-inverse" th:fragment="navigation">
20       <div class="container-fluid">
21         <div class="navbar-header">
22           <a class="navbar-brand" th:href="@{/}">Epidemic
23 Analyzer</a>
24         </div>
25         <ul class="nav navbar-nav">
26           <li class="dropdown">
27             <a class="dropdown-toggle" data-toggle="
28 dropdown" href="#">Patient case
29               <span class="caret"></span></a>
30               <ul class="dropdown-menu">
31                 <li><a th:href="@{/patient_case/add_
32 patient_case}">Add new patient case</a></li>
33                 <li><a th:href="@{/patient_case/display_
34 patient_cases}">Display all patient cases</a></li>
35                 <li><a th:href="@{/patient_case/project_
36 to_map}">Project cases on map</a></li>
37               </ul>
38             </li>

```

EPIDEMIC ANALYZER

```
29 | <li>
30 |     <a th:href="@{/patient_case/sir_modelling}">↵
    |     SIR Modelling</a>
31 | </li>
32 | <li>
33 |     <a href="#">User Manual</a>
34 | </li>
35 | <li>
36 |     <a th:href="@{/}">Logout</a>
37 | </li>
38 | </ul>
39 | </div>
40 | </nav>
41 | </body>
42 | </html>
```

EPIDEMIC ANALYZER

static.templates.index.html

```

1  <!DOCTYPE html>
2  <!--
3  To change this license header, choose License Headers in Project
4  Properties.
5  To change this template file, choose Tools | Templates
6  and open the template in the editor.
7  -->
8  <html>
9      <head th:replace="fragments::head">
10     </head>
11     <body class = "container">
12         <nav th:replace="fragments::navigation"></nav>
13         <h2>Welcome!</h2>
14
15         <div class="container">
16             <div class="row">
17                 <div class="col-sm-4">
18                     <p class = "indexContent">
19                         In the milieu of a humanitarian crisis (such
20                         as the 2014 Ebola epidemic), responders including public health
21                         personnels require epidemic data and its analysis. Such
22                         information is paramount for decision makers whose discernments
23                         and actions have direct impact on the
24                         control of the outbreak, consequently saving lives.
25                     </p>
26                     <h4>Aims</h4>
27                     <p>
28                         <ul class = "indexContent">
29                             <li>To build a system that can capture
30                                 epidemic data on ground zero and provide analytics for
31                                 provincial and national level decision
32                                 makers.</li>
33                             <li>To contribute to the understanding of
34                                 public health practitioners and humanitarian agencies
35                                 about an epidemic situation through the
36                                 introduction of information technologies.</li>
37                         </ul>

```

EPIDEMIC ANALYZER

```

30         </p>
31         <h4>Objectives</h4>
32         <p>
33         <ul class = "indexContent">
34             <li>To provide a robust data entry form that↵
captures a broad range of health data dynamics.</li>
35             <li>To provide a robust data entry form that↵
captures a broad range of health data dynamics.</li>
36             <li>To provide a graphical representation of↵
data for better comprehension & decision making.</li>
37             <li>To provide a graphical representation of↵
data for better comprehension & decision making.</li>
38             <li>To learn about the impact of information↵
technology in international humanitarian interventions.</li>
39             <li>To apply the principles from academic ↵
learnings to the real world scenarios and garner
40                 experience in research and development.<↵
/li>
41         </ul>
42         </p>
43     </div>
44
45     <div class="col-sm-4">
46         <blockquote>
47             <p>
48                 <span id = "ea">Epidemic analyzer</span>↵
- a web-based epidemic data management cum analysis system that ↵
aids humanitarian actors in information acquisition & analysis. ↵
49             </p>
50         </blockquote>
51         <h4>Get Started</h4>
52         <p>
53         <ul class = "indexContent">
54             <li><a href="#">Read the user manual</a></li>
55             <li><a href="/patient_case/add_patient_case"↵
>Add new epidemic case</a></li>
56             <li><a href="/patient_case/display_patient_↵
cases">View/Edit existing cases</a></li>
57             <li><a href="/patient_case/project_to_map">↵
Project cases on map</a></li>
58         </ul>
59         </p>
60     </div>
61 </div>
62 </div>
63 </body>
64 </html>
65

```

EPIDEMIC ANALYZER

static.templates.patient_case.add_patient_case.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org/">
3   <head th:replace="fragments :: head"></head>
4   <body class="container">
5     <nav th:replace="fragments::navigation"></nav>
6     <h1>Add new patient case</h1>
7     <form method = "POST" th:object="${patient}">
8       <div class="form-group">
9         <input th:field="*{firstName}" placeholder = "First
10        name" />
11         <span th:errors="*{firstName}" class = "error"></span>
12       </div>
13       <div class="form-group">
14         <input th:field="*{middleName}" placeholder = "
15        Middle name" />
16         <span th:errors="*{middleName}" class = "error"></
17        span>
18       </div>
19       <div class="form-group">
20         <input th:field="*{lastName}" placeholder = "Last
21        name" />
22         <span th:errors="*{lastName}" class = "error"></span>
23       </div>
24       <div class="form-group">
25         <input th:field = "${patient.address.district}"
26        placeholder = "District" required = "required" />
27         <span th:errors="${patient.address.district}" class
28        = "error" ></span>
29       </div>
30       <div class="form-group">
31         <input th:field = "${patient.address.vdc}"
32        placeholder = "VDC" required = "required" />
33         <span th:errors="${patient.address.vdc}" class = "
34        error"></span>
35       </div>
36       <div class="form-group">
37         Ward No
38         <input th:field = "${patient.address.wardNo}"
39       </div>

```

EPIDEMIC ANALYZER

```

placeholder = "Ward No" type = "number" min = "1" max = "9"/>
31      <span th:errors="{patient.address.wardNo}" class =
"error"></span>
32      </div>
33
34      <div class="form-group">
35      <label class="label label-default">GPS Latitude</
label>
36      <input th:field = "{patient.address.latitude}"
placeholder = "Latitude"/>
37      <span th:errors="{patient.address.latitude}" class
= "error"></span>
38      </div>
39      <div class="form-group">
40      <label class="label label-default">GPS Longitude</
label>
41      <input th:field = "{patient.address.longitude}"
placeholder = "Longitude"/>
42      <span th:errors="{patient.address.longitude}" class
= "error"></span>
43      </div>
44      <div class="form-group">
45      <input th:field="{dob}" type = "date" placeholder =
"Date of birth"/>
46      <span th:errors="{dob}" class = "error"></span>
47      </div>
48      <div class="form-group">
49      <select th:field="{gender}" placeholder = "Gender">
50      <option th:each = "gender: {genderList}"
th:text="{gender.displayName}"
th:value = "{gender}"></option>
51      </select>
52      </div>
53      <div class="form-group">
54      <select th:field="{currentInfectionStatus}"
placeholder = "Current Infection Status">
55      <option th:each = "currentInfectionStatus: {
currentInfectionStatusList}"
th:text="{currentInfectionStatus.
displayName}"
th:value = "{currentInfectionStatus}"></
option>
56      </select>
57      </div>
58      <input type = "submit" value = "Add case"/>
59      </form>
60      </body>
61      </html>

```

EPIDEMIC ANALYZER

static.templates.patient_case.display_patient_cases.html

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org/">
3 <head th:replace="fragments :: head"></head>
4 <body class="container">
5     <nav th:replace="fragments::navigation"></nav>
6     <h1>Patient case entries</h1>
7     <form method = "POST">
8         <table id = "patientcasetable" class = "table-condensed">
9             <thead>
10                 <tr>
11                     <th role="button">Patient ID</th>
12                     <th role="button">Current infection status</th>
13                     <th role="button">First Name</th>
14                     <th role="button">Middle Name</th>
15                     <th role="button">Last Name</th>
16                     <th role="button">Date of Birth</th>
17                     <th role="button">Gender</th>
18                     <th role="button">District</th>
19                     <th role="button">VDC</th>
20                     <th role="button">Ward No</th>
21                     <th role="button">Lat.</th>
22                     <th role="button">Long.</th>
23                     <th role="button">Select</th>
24                     <th></th>
25                 </tr>
26             </thead>
27             <tbody>
28                 <tr th:each="patient : ${patients}">
29                     <td th:text="${patient.id}"></td>
30                     <td th:text="${patient.currentInfectionStatus}">↵
31
32                     <td th:text="${patient.firstName}"></td>
33                     <td th:text="${patient.middleName}"></td>
34                     <td th:text="${patient.lastName}"></td>
35                     <td th:text="${patient.dob}"></td>
36                     <td th:text="${patient.gender}"></td>
37                     <td th:text="${patient.address.district}"></td>
38                     <td th:text="${patient.address.vdc}"></td>
```

EPIDEMIC ANALYZER

```

38         <td th:text="{patient.address.wardNo}"></td>
39         <td th:text="{patient.address.latitude}"></td>
40         <td th:text="{patient.address.longitude}"></td>
41         <td>
42             <input type="checkbox" name="patientIds" th:↵
value="{patient.id}" th:id="{patient.id}" />
43         </td>
44         <td>
45             <a class ="btn btn-success" th:href="{↵
update_patient_case/} + {patient.id}">Update</a>
46         </td>
47     </tr>
48 </tbody>
49 </table>
50     <input class="btn btn-danger" type="submit" value="Delete ↵
selected" onclick="return confirm('Confirm deletion?');"/>
51 </form>
52 </body>
53 </html>

```


EPIDEMIC ANALYZER

static.templates.patient_case.epidemic_map_projection.html

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8"/>
5      <title>Epidemic Analyzer</title>
6      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
7      <link rel="stylesheet" th:href="@{/css/styles.css}" />
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
9      <script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
10     <script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
11     <script type="text/javascript" th:src="@{/js/tabledata_logic.js}"></script>
12     <style>
13       /* element that contains the map. */
14       #map {
15         height: 100%;
16         width: 100%;
17       }
18       /* Optional: Makes the sample page fill the window. */
19       html, body {
20         height: 100%;
21         margin: 0;
22         padding: 0;
23       }
24       #legend {
25         background: #fff;
26         padding: 10px;
27         margin: 10px;
28         border: 2px solid #000;
29       }
30     </style>
31   </head>
32   <body class = "container">

```

EPIDEMIC ANALYZER

```

33 <body class = "container">
34   <nav th:replace="fragments::navigation"></nav>
35   <h1>Geographic projection - Epidemic cases</h1>
36   <div id="map">
37     <script>
38       var customLabel = {
39         restaurant: {
40           label: 'R'
41         },
42         bar: {
43           label: 'B'
44         }
45       };
46       function initMap() {
47         var map = new google.maps.Map(document.↵
getElementById('map'), {
48           center: new google.maps.LatLng(-33.863276, ↵
151.207977),
49           zoom: 12
50         });
51         var infoWindow = new google.maps.InfoWindow;
52         map.controls[google.maps.ControlPosition.RIGHT_↵
BOTTOM].push
53           (document.getElementById('legend'));
54
55           // Change this depending on the name of your ↵
PHP or XML file
56           //           downloadUrl('src/main/↵
resources/static/xml/EpidemicMapProjectionData.xml', function (data↵
) {
57             downloadUrl('http://localhost:8080/xml/↵
EpidemicMapProjectionData.xml', function (data) {
58               var xml = data.responseXML;
59               var markers = xml.documentElement.↵
getElementsByTagName('marker');
60               Array.prototype.forEach.call(markers, ↵
function (markerElem) {

```

EPIDEMIC ANALYZER

```

61
62         //Getting model attributes and values
63         var id = markerElem.getAttribute('id');
64         var name = markerElem.getAttribute('↵
name');
65         var age = markerElem.getAttribute('age'↵
);
66         var gender = markerElem.getAttribute('↵
gender');
67         var address = markerElem.getAttribute('↵
address');
68         var currentInfectionStatus = markerElem↵
.getAttribute('currentInfectionStatus');
69
70         var point = new google.maps.LatLng(
71             parseFloat(markerElem.↵
getAttribute('lat')),
72             parseFloat(markerElem.↵
getAttribute('lng')));
73
74         var infowincontent = document.↵
createElement('div');
75
76         //Adding info content
77         var strong = document.createElement('↵
strong');
78         strong.textContent = name;
79         infowincontent.appendChild(strong);
80
81         var cis = document.createElement('em');
82         cis.textContent = ↵
currentInfectionStatus;
83
84         infowincontent.appendChild(document.↵
createElement('br'));
85         infowincontent.appendChild(cis);

```

EPIDEMIC ANALYZER

```

86         infowincontent.appendChild(document.↵
createElement('br'));
87
88         var info_id = document.createTextNode('↵
ID: ' + id);
89         infowincontent.appendChild(document.↵
createElement('br'));
90         infowincontent.appendChild(info_id);
91
92         var info_age = document.createTextNode(↵
'Age: ' + age);
93         infowincontent.appendChild(document.↵
createElement('br'));
94         infowincontent.appendChild(info_age);
95
96         var info_gender = document.↵
createTextNode('Gender: ' + gender);
97         infowincontent.appendChild(document.↵
createElement('br'));
98         infowincontent.appendChild(info_gender);
99
100        var info_address = document.↵
createTextNode('Address: ' + address);
101        infowincontent.appendChild(document.↵
createElement('br'));
102        infowincontent.appendChild(info_address↵
);
103
104
105        //Adding icons
106        var icon = customLabel[↵
currentInfectionStatus] || {};
107        var iconBase = 'http://localhost:8080/↵
icons/';
108        var icons = {
109            Infected: {

```

EPIDEMIC ANALYZER

```

110         icon: iconBase + 'infected.ico'
111     },
112     Susceptible: {
113         icon: iconBase + 'susceptible.
png'
114     },
115     Recovered: {
116         icon: iconBase + 'recovered.png'
117     }
118 };
119
120 //Adding marker
121 var marker = new google.maps.Marker({
122     map: map,
123     position: point,
124     icon: icons[currentInfectionStatus]
.icon,
125     label: icon.label
126 });
127
128 marker.addListener('click', function ()
{
129     infoWindow.setContent(
infowincontent);
130     infoWindow.open(map, marker);
131 });
132
133 });
134 });
135 //Adding Legend
136 var legend = document.getElementById('legend');
137 var div = document.createElement('div');
138 div.innerHTML = '<p> Infected</p>' +
    '<p>Susceptible</p>' +
    'Recovered'
    ;
142     legend.appendChild(div);
143 }
144 function downloadUrl(url, callback) {
145     var request = window.ActiveXObject ?
146         new ActiveXObject('Microsoft.XMLHTTP') :
147         new XMLHttpRequest;
148
149     request.onreadystatechange = function () {
150         if (request.readyState == 4) {
151             request.onreadystatechange = doNothing;
152             callback(request, request.status);
153         }
154     };
155
156     request.open('GET', url, true);
157     request.send(null);
158 }
159 function doNothing() {}
160 </script>
161 <script async="async" defer="defer"
162     src="https://maps.googleapis.com/maps/api/js?
163     key=AIzaSyD9fBatjhGfzLW-eglk4SAkkiTyVGHD9AY&callback=initMap">
164 </script>
165 </div>
166 <div id="legend"><h3>Legend</h3></div>
167 </body>
</html>

```

EPIDEMIC ANALYZER

static.templates.patient_case.sir_modelling.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org/">
3   <head>
4     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"/>
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
6     <script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
7     <link rel="stylesheet" href="http://jsxgraph.uni-bayreuth.de/distrib/jsxgraph.css" />
8     <script type="text/javascript" src="http://cdnjs.cloudflare.com/ajax/libs/jsxgraph/0.99.6/jsxgraphcore.js"></script>
9     <style>
10       /*Importing font*/
11       @font-face {
12         font-family: OFL Sorts Mill Goudy;
13         font-style: normal;
14         font-weight: normal;
15         src: url(../fonts/OFLGoudyStM-webfont.eot) format("embedded-opentype"),
16              url(../fonts/OFLGoudyStM-webfont.woff) format("woff"),
17              url(../fonts/OFLGoudyStM-webfont.svg) format("svg")
18       }
19
20       /*Basic Styling*/
21       nav,h1,h2,h3,legend,button, input[type="submit"],input[type="button"], input[type="reset"],caption{
22         font-family: "OFL Sorts Mill Goudy";
23         text-transform: uppercase;
24         color:#113654;
25       }
26       body{
27         background:url("../images/symphony.png") fixed;
28         font-family: "Georgia";
29       }

```

EPIDEMIC ANALYZER

```

30     </style>
31 </head>
32 <body class="container">
33     <nav th:replace="fragments::navigation"></nav>
34     <h1>SIR Modelling</h1>
35     <form><input class = "btn btn-primary" type="button" value=↵
"clear and run a simulation of 200 days"    onClick="clearturtle()↵
;run()" />
36         <input class = "btn" type="button" value="stop" onClick=↵
="stop()" />
37         <input class = "btn" type="button" value="continue" ↵
onClick="goOn()" />
38     </form>
39     <input id="myInput" type="number" th:value="{population}" />
40     <button class = "btn" onclick = "loadPopulation()">Set ↵
population</button>
41     <div id='jxgbox' class='jxgbox' style='width:700px; height:↵
500px;'></div>
42     <div>Legend:
43         <div>* <span style="color:Blue">Blue: Rate of ↵
susceptible population</span></div>
44         <div>* <span style="color:red">Red: Rate of infected ↵
population</span></div>
45         <div>* <span style="color:green">Green: Rate of ↵
recovered population</span></div>
46     </div>
47     <script>
48         /*<![CDATA[* /
49         /*
50         * Author: @Biju Ale
51         var brd -boundingbox parameters:
52         [a, b, c, d]
53         a = y-axis horizontal pos ('-' pull in),
54         b = y-axis height limit
55         c = x-axis width limit
56         d = y-axis scale (division size)
57         */
58         var population = 1000000;

```


EPIDEMIC ANALYZER

```

59 function loadPopulation() {
60     var _population = document.getElementById("myInput").value;
61     if (_population * 1 !== population * 1) {
62         population = _population * 1;
63     }
64 }
65 var brd = JXG.JSXGraph.initBoard('jxgbox', {boundingbox: [-9.66, 1.1, 246.66, -0.5], axis: true});
66
67 var S = brd.create('turtle', [], {strokeColor: 'blue', strokeWidth: 3});
68 var I = brd.create('turtle', [], {strokeColor: 'red', strokeWidth: 3});
69 var R = brd.create('turtle', [], {strokeColor: 'green', strokeWidth: 3});
70
71 /*
72  * Author: @Biju Ale
73  * var brd - slider position parameters:
74  * [[a, b], [c, d], [e, f, g]]
75  * a = horizontal beginning edge
76  * b = x-axis coordinate
77  * c = slider length
78  * d = y-axis coordinate
79  * e = min value for pointerr slider
80  * f = current value of slider pointer
81  * g = max value for slider
82  */
83 var s = brd.create('slider', [[0, -0.1], [60, -0.1], [0, 1E-6, 1]], {name: 's'});
84 var beta = brd.create('slider', [[0, -0.2], [60, -0.2], [0, 0.2857, 1]], {name: '&beta;'});
85 var gamma = brd.create('slider', [[0, -0.3], [60, -0.3], [0, 0.1428, 0.5]], {name: '&gamma;'});
86 var mort = brd.create('slider', [[0, -0.4], [60, -0.4], [0, 0.0, 10.0]], {name: '% mortality'});
87

```

EPIDEMIC ANALYZER

```

88      /*
89      * Author: @Biju Ale
90      var brd - text position parameters:
91      a = horizontal positioning (+: push-out, -: pull-in )
92      b = y-axis height limit
93      */
94      brd.create('text', [90, -0.1, "initially infected ←
population rate"]);
95      brd.create('text', [90, -0.2, function () {
96          return "&beta;:: infection rate, R<sub>0</sub>=" +
+ (beta.Value() / gamma.Value()).toFixed(2);
97      }]);
98      brd.create('text', [90, -0.3, function () {
99          return "&gamma;:: recovery rate = 1/(days of ←
infection), days of infection=" + (1 / gamma.Value()).toFixed(1);
100      }]);
101
102      var t = 0; // global
103
104      brd.create('text', [90, 1.06,
105          function () {
106              return "Day " + t +
107                  ": infected=" + (population * I.Y()).←
toFixed(1) +
108                  " recovered=" + (population * R.Y()).←
toFixed(1) +
109                  " dead=" + (population * R.Y() * mort.←
Value() * 0.01).toFixed(0);
110          }]);
111
112      S.hideTurtle();
113      I.hideTurtle();
114      R.hideTurtle();
115
116      function clearturtle() {
117          S.cs();
118          I.cs();
119          R.cs();

```

EPIDEMIC ANALYZER

```

120
121     S.hideTurtle();
122     I.hideTurtle();
123     R.hideTurtle();
124 }
125 function run() {
126     S.setPos(0, 1.0 - s.Value());
127     R.setPos(0, 0);
128     I.setPos(0, s.Value());
129
130     delta = 1; // global
131     t = 0; // global
132     loop();
133 }
134 function loop() {
135     var dS = -beta.Value() * S.Y() * I.Y();
136     var dR = gamma.Value() * I.Y();
137     var dI = -(dS + dR);
138     turtleMove(S, delta, dS);
139     turtleMove(R, delta, dR);
140     turtleMove(I, delta, dI);
141
142     t += delta;
143     if (t < 200.0) {
144         active = setTimeout(loop, 10);
145     }
146 }
147 function turtleMove(turtle, dx, dy) {
148     turtle.moveTo([dx + turtle.X(), dy + turtle.Y()]);
149 }
150 function stop() {
151     if (active)
152         clearTimeout(active);
153     active = null;
154 }
155 function goOn() {
156     if (t > 0) {
157         if (active == null) {
158             active = setTimeout(loop, 10);
159         }
160     } else {
161         run();
162     }
163 }
164 }
165 /*]]>*/
166 </script>
167 </body>
168 </html>

```

EPIDEMIC ANALYZER

static.templates.patient_case.update_patient_case.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org/">
3   <head th:replace="fragments :: head"></head>
4   <body class="container">
5     <nav th:replace="fragments::navigation"></nav>
6     <h1>Update patient case</h1>
7     <form method = "POST" th:object="${patient}">
8       <div class="form-group">
9         <input th:field="*{firstName}" placeholder = "First
10        name" th:value="${patient.firstName}"/>
11         <span th:errors="*{firstName}" class = "error"></span>
12       </div>
13       <div class="form-group">
14         <input th:field="*{middleName}" placeholder = "
15        Middle name" th:value="${patient.middleName}"/>
16         <span th:errors="*{middleName}" class = "error"></
17        span>
18       </div>
19       <div class="form-group">
20         <input th:field="*{lastName}" placeholder = "Last
21        name" th:value="${patient.lastName}"/>
22         <span th:errors="*{lastName}" class = "error"></span>
23       </div>
24       <div class="form-group">
25         <input th:field = "${patient.address.district}"
26        placeholder = "District" required = "required"/>
27         <span th:errors="${patient.address.district}" class
28        = "error" ></span>
29       </div>
30       <div class="form-group">
31         <input th:field = "${patient.address.vdc}"
32        placeholder = "VDC" required = "required"/>
33         <span th:errors="${patient.address.vdc}" class = "
34        error"></span>
35       </div>
36       <div class="form-group">
37         <input th:field = "${patient.address.wardNo}"
38        placeholder = "Ward No" type = "number" min = "1" max = "9"/>

```

EPIDEMIC ANALYZER

```

30         <span th:errors="${patient.address.wardNo}" class = ↵
"error"></span>
31     </div>
32     <div class="form-group">
33         <label class="label label-default">GPS Latitude</↵
label>
34         <input th:field = "${patient.address.latitude}" ↵
placeholder = "Latitude"/>
35         <span th:errors="${patient.address.latitude}" class ↵
= "error"></span>
36     </div>
37     <div class="form-group">
38         <label class="label label-default">GPS Longitude</↵
label>
39         <input th:field = "${patient.address.longitude}" ↵
placeholder = "Longitude"/>
40         <span th:errors="${patient.address.longitude}" class↵
= "error"></span>
41     </div>
42     <div class="form-group">
43         <input th:field="*{dob}" type = "date" placeholder =↵
"Date of birth" th:value="${patient.dob}"/>
44         <span th:errors="*{dob}" class = "error"></span>
45     </div>
46     <div class="form-group">
47         <select th:field="*{gender}" placeholder = "Gender">
48             <option th:each = "gender: ${genderList}"
49                 th:text="${gender.displayName}"
50                 th:value = "${gender}"></option>
51         </select>
52     </div>
53     <div class="form-group">
54         <select th:field="*{currentInfectionStatus}" ↵
placeholder = "Current Infection Status">
55             <option th:each = "currentInfectionStatus: ${↵
currentInfectionStatusList}"
56                 th:text="${currentInfectionStatus.↵
displayName}"
57                 th:value = "${currentInfectionStatus}"></↵
option>
58     </select>
59     </div>
60     <input class = "btn btn-success" type = "submit" value =↵
"Update case"/>
61 </form>
62 </body>
63 </html>

```

EPIDEMIC ANALYZER

pom.xml

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2       xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3     <modelVersion>4.0.0</modelVersion>
4     <groupId>ea.biju</groupId>
5     <artifactId>EdidemicAnalyzer</artifactId>
6     <version>0.0.1-SNAPSHOT</version>
7
8     <parent>
9       <groupId>org.springframework.boot</groupId>
10      <artifactId>spring-boot-starter-parent</artifactId>
11      <version>1.5.7.RELEASE</version>
12    </parent>
13
14    <!-- Java Version -->
15    <properties>
16      <java.version>1.8</java.version>
17    </properties>
18
19    <dependencies>
20
21      <!-- Spring Web -->
22      <dependency>
23        <groupId>org.springframework.boot</groupId>
24        <artifactId>spring-boot-starter-web</artifactId>
25      </dependency>
26
27      <!-- Spring Data JPA -->
28      <dependency>
29        <groupId>org.springframework.boot</groupId>
30        <artifactId>spring-boot-starter-data-jpa</artifactId>
31      </dependency>
32
33      <!-- Spring Security -->
34      <dependency>
35        <groupId>org.springframework.boot</groupId>
36        <artifactId>spring-boot-starter-security</artifactId>
```

EPIDEMIC ANALYZER

```

37     </dependency>
38
39     <!-- Spring Test -->
40     <dependency>
41         <groupId>org.springframework.boot</groupId>
42         <artifactId>spring-boot-starter-test</artifactId>
43         <scope>test</scope>
44     </dependency>
45
46     <!-- MySQL -->
47     <dependency>
48         <groupId>mysql</groupId>
49         <artifactId>mysql-connector-java</artifactId>
50     </dependency>
51
52     <!--J-Unit-->
53     <dependency>
54         <groupId>junit</groupId>
55         <artifactId>junit</artifactId>
56         <type>jar</type>
57     </dependency>
58
59     <!--Hamcrest-->
60     <dependency>
61         <groupId>org.hamcrest</groupId>
62         <artifactId>hamcrest-core</artifactId>
63         <scope>test</scope>
64     </dependency>
65
66     <!--HttpServletRequest-->
67     <dependency>
68         <groupId>org.apache.tomcat.embed</groupId>
69         <artifactId>tomcat-embed-core</artifactId>
70         <type>jar</type>
71     </dependency>
72
73     <!-- hot swapping, disable cache for template, enable liv
reload -->

```

EPIDEMIC ANALYZER

```

74 <dependency>
75     <groupId>org.springframework.boot</groupId>
76     <artifactId>spring-boot-devtools</artifactId>
77     <optional>true</optional>
78 </dependency>
79
80 <!--Thymeleaf-->
81 <dependency>
82     <groupId>org.springframework.boot</groupId>
83     <artifactId>spring-boot-starter-thymeleaf</artifactId>
84 </dependency>
85
86 <!-- Moxy - XML Binding-->
87 <dependency>
88     <groupId>org.eclipse.persistence</groupId>
89     <artifactId>org.eclipse.persistence.moxy</artifactId>
90     <version>2.5.0</version>
91 </dependency>
92
93 <!-- Mockito-->
94 <dependency>
95     <groupId>org.mockito</groupId>
96     <artifactId>mockito-core</artifactId>
97     <version>1.9.5</version>
98     <scope>test</scope>
99 </dependency>
100
101 <!--Joda Time-->
102 <dependency>
103     <groupId>joda-time</groupId>
104     <artifactId>joda-time</artifactId>
105     <version>2.9.9</version>
106 </dependency>
107
108 </dependencies>
109
110 <build>
111     <plugins>
112         <!-- Package as an executable jar/war -->
113         <plugin>
114             <groupId>org.springframework.boot</groupId>
115             <artifactId>spring-boot-maven-plugin</artifactId>
116         </plugin>
117     </plugins>
118 </build>
119 <name>EdidemicAnalyzer</name>
120 </project>

```


EPIDEMIC ANALYZER

nbbactions.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <actions>
3      <action>
4          <actionName>run</actionName>
5          <packagings>
6              <packaging>jar</packaging>
7          </packagings>
8          <goals>
9              <goal>process-classes</goal>
10             <goal>org.codehaus.mojo:exec-maven-plugin:1.2.1:exec↵
11         </goal>
12         </goals>
13         <properties>
14             <exec.args>-classpath %classpath ea.biju.Application↵
15         </exec.args>
16         <exec.executable>java</exec.executable>
17         </properties>
18     </action>
19     <action>
20         <actionName>debug</actionName>
21         <packagings>
22             <packaging>jar</packaging>
23         </packagings>
24         <goals>
25             <goal>process-classes</goal>
26             <goal>org.codehaus.mojo:exec-maven-plugin:1.2.1:exec↵
27         </goal>
28         </goals>
29         <properties>
30             <exec.args>-Xdebug -Xrunjdpw:transport=dt_socket,↵
31             server=n,address=${jpda.address} -classpath %classpath ea.biju.↵
32             Application</exec.args>
33             <exec.executable>java</exec.executable>
34             <jpda.listen>true</jpda.listen>
35         </properties>
36     </action>
37     <action>
38         <actionName>profile</actionName>

```

EPIDEMIC ANALYZER

```

34  <packagings>
35      <packaging>jar</packaging>
36  </packagings>
37  <goals>
38      <goal>process-classes</goal>
39      <goal>org.codehaus.mojo:exec-maven-plugin:1.2.1:exec<
40  </goal>
41  </goals>
42  <properties>
43      <exec.args>-classpath %classpath ea.biju.Application<
44  </exec.args>
45      <exec.executable>java</exec.executable>
46  </properties>
47  </action>
48  </actions>

```

nb-configuration.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <project-shared-configuration>
3      <!--
4      This file contains additional configuration written by modules in
5      the NetBeans IDE.
6      The configuration is intended to be shared among all the users of
7      project and
8      therefore it is assumed to be part of version control checkout.
9      Without this configuration present, some functionality in the IDE
10     may be limited or fail altogether.
11     -->
12     <properties xmlns="http://www.netbeans.org/ns/maven-properties-
13     data/1">
14         <!--
15         Properties that influence various parts of the IDE, especially code
16         formatting and the like.
17         You can copy and paste the single properties, into the pom.xml file
18         and the IDE will pick them up.
19         That way multiple projects can share the same settings (useful for
20         formatting rules for example).
21         Any value defined here will override the pom.xml file value but is
22         only applicable to the current project.
23         -->
24         <org-netbeans-modules-maven-j2ee.netbeans_2e_hint_2e_
25         j2eeVersion>1.7-web</org-netbeans-modules-maven-j2ee.netbeans_2e_
26         hint_2e_j2eeVersion>
27     </properties>
28 </project-shared-configuration>
29

```

Test Scripts

Patient Case CRUD operations
(on next page)

EPIDEMIC ANALYZER

```

6 package ea.biju;
7
8 import org.junit.Before;
9 import org.junit.Test;
10 import org.mockito.InjectMocks;
11 import org.mockito.Mock;
12 import org.mockito.MockitoAnnotations;
13 import org.springframework.test.web.servlet.MockMvc;
14 import org.springframework.test.web.servlet.setup.MockMvcBuilders;
15 import ea.biju.controller.PatientCaseController;
16 import ea.biju.enums.CurrentInfectionStatus;
17 import ea.biju.enums.Gender;
18 import ea.biju.model.Patient;
19 import ea.biju.repo.PatientRepository;
20 import java.util.ArrayList;
21 import java.util.List;
22 import static org.hamcrest.Matchers.hasSize;
23 import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*;
24 import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.model;
25 import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;
26 import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.view;
27 import static org.hamcrest.Matchers.instanceOf;
28 import static org.mockito.Mockito.when;
29
30 /**
31  *
32  * @author Biju Ale
33  */
34 public class UnitTestPatientCaseController {
35
36     @Mock
37     private PatientRepository patientRepository;

```

EPIDEMIC ANALYZER

```

38
39     @InjectMocks
40     private PatientCaseController patientCaseController;
41
42     private MockMvc mockMvc;
43
44     @Before
45     public void setup() {
46         MockitoAnnotations.initMocks(this);
47         mockMvc = MockMvcBuilders.standaloneSetup(↵
patientCaseController).build();
48         MockitoAnnotations.initMocks(this);
49     }
50
51     @Test
52     public void testDisplayAddPatientCaseView() throws Exception {
53         mockMvc.perform(get("/patient_case/add_patient_case"))
54             .andExpect(status().isOk())
55             .andExpect(model().attribute("↵
currentInfectionStatusList", CurrentInfectionStatus.values()))
56             .andExpect(model().attribute("genderList", Gender.↵
values()))
57             .andExpect(view().name("patient_case/add_patient_case↵
"));
58     }
59
60     @Test
61     public void testProcessAddPatientCaseView() throws Exception {
62
63         Patient p = new Patient();
64         p.setFirstName("Test");
65         patientRepository.save(p);
66         mockMvc.perform(post("/patient_case/add_patient_case"))
67             .andExpect(status().isOk())
68             .andExpect(model().attribute("patient", instanceOf(↵
Patient.class)))
69             .andExpect(model().attribute("genderList", Gender.↵
values()))
70             .andExpect(view().name("patient_case/add_patient_case↵
"));
71     }

```

EPIDEMIC ANALYZER

```

72
73     @Test
74     public void testDisplayAllPatientCases() throws Exception {
75
76         List<Patient> patients = new ArrayList<>();
77         patients.add(new Patient());
78         patients.add(new Patient());
79
80         when(patientRepository.findAll()).thenReturn((List) patients);
81
82         mockMvc.perform(get("/patient_case/display_patient_cases"))
83             .andExpect(status().isOk())
84             .andExpect(view().name("patient_case/display_patient_↵
cases"))
85             .andExpect(model().attribute("patients", hasSize(2)));
86
87     }
88
89     @Test
90     public void testProcessUpdatePatientCaseView() throws Exception {
91         Integer id = 1;
92         when(patientRepository.findOne(id)).thenReturn(new Patient());
93
94         mockMvc.perform(post("/patient_case/update_patient_case/1"))
95             .andExpect(status().isOk())
96             .andExpect(model().attribute("patient", instanceof(↵
Patient.class)))
97             .andExpect(model().attribute("↵
currentInfectionStatusList", CurrentInfectionStatus.values()))
98             .andExpect(model().attribute("genderList", Gender.↵
values()))
99             .andExpect(view().name("patient_case/update_patient_↵
case"));
100     }
101
102     @Test
103     public void removePatientCase() throws Exception {
104         Integer[] id = new Integer[1];
105         id[0] = 1;
106         Patient p = new Patient();
107         patientRepository.delete(p);

```

EPIDEMIC ANALYZER

```

108         List<Patient> patients = new ArrayList<>();
109         patients.add(new Patient());
110         patients.add(new Patient());
111         when(patientRepository.findAll()).thenReturn((List) patients);
112         mockMvc.perform(get("/patient_case/display_patient_cases"))
113             .andExpect(status().isOk())
114             .andExpect(view().name("patient_case/display_patient_
cases"))
115             .andExpect(model().attribute("patients", hasSize(2)));
116
117     }
118 }
119











```

Test Results × Output - Test (UnitTestPatientCaseController)

ea.biju:EdidemicAnalyzer:jar:0.0.1-SNAPSHOT ×

Tests passed: 100.00 %

All 5 tests passed. (3.218 s)

EPIDEMIC ANALYZER

Patient case data projection on map

```

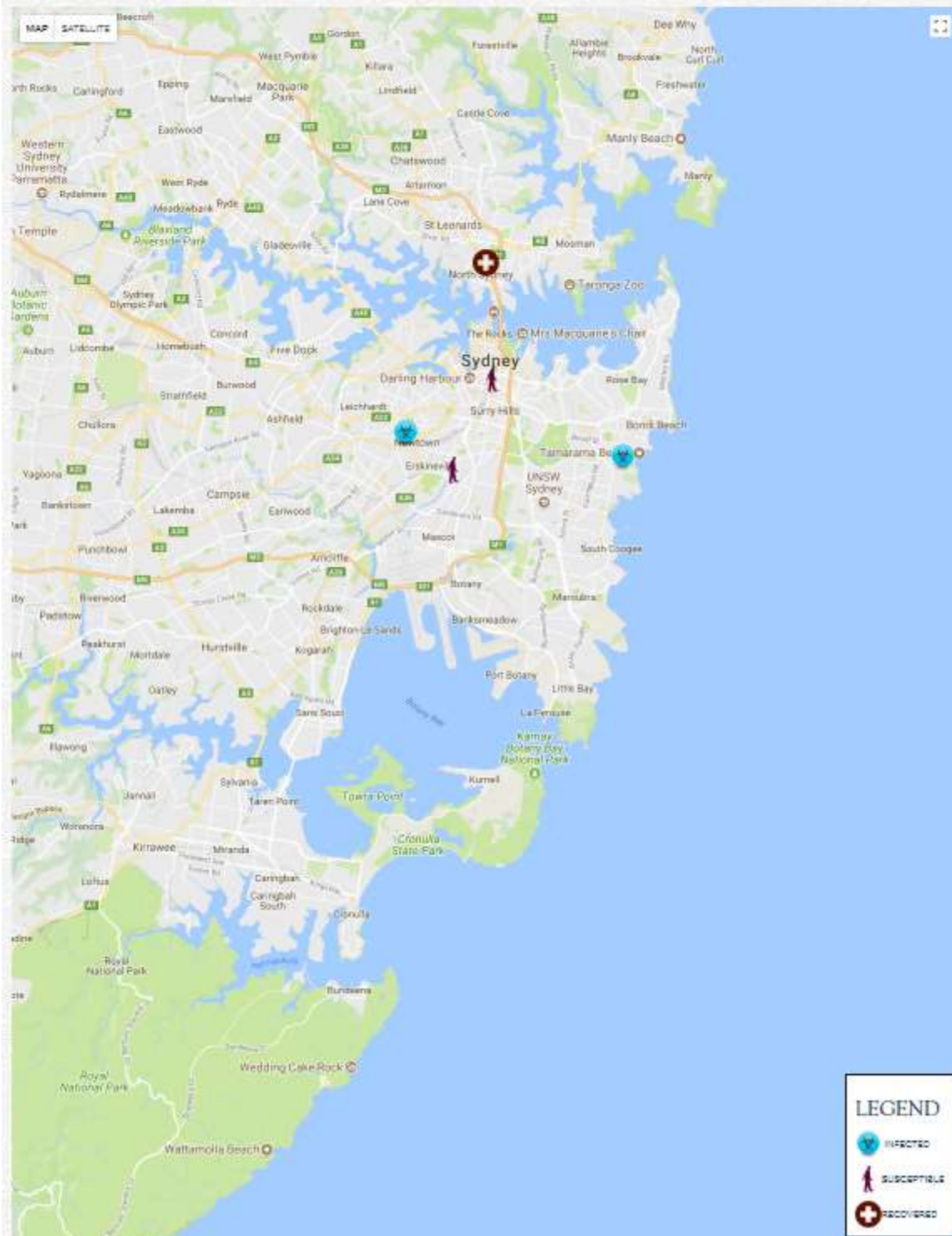
6  package ea.biju;
7
8  import org.junit.Before;
9  import org.mockito.InjectMocks;
10 import org.mockito.Mock;
11 import org.mockito.MockitoAnnotations;
12 import org.springframework.test.web.servlet.MockMvc;
13 import org.springframework.test.web.servlet.setup.MockMvcBuilders;
14 import ea.biju.controller.PatientCaseController;
15 import ea.biju.repo.PatientRepository;
16
17 /**
18  *
19  * @author Biju Ale
20  */
21 public class UnitTestMapProjectionController {
22
23     @Mock
24     private PatientRepository patientRepository;
25
26     @InjectMocks
27     private PatientCaseController patientCaseController;
28
29     private MockMvc mockMvc;
30
31     @Before
32     public void setup() {
33         MockitoAnnotations.initMocks(this);
34         mockMvc = MockMvcBuilders.standaloneSetup(patientCaseController).build();
35         MockitoAnnotations.initMocks(this);
36     }
37
38     //All tests were run & verified with walkthrough and documented
39     //because test needs visual verification due to graphics rendering
40
41 }

```


EPIDEMIC ANALYZER

EPIDEMIC ANALYZER PATIENT CASE - SIR MODELLING USER MANUAL LOGOUT

GEOGRAPHIC PROJECTION - EPIDEMIC CASES



EPIDEMIC ANALYZER

SIR Modelling

```

6   package ea.biju;
7
8   import org.junit.Before;
9   import org.mockito.InjectMocks;
10  import org.mockito.Mock;
11  import org.mockito.MockitoAnnotations;
12  import org.springframework.test.web.servlet.MockMvc;
13  import org.springframework.test.web.servlet.setup.MockMvcBuilders;
14  import ea.biju.controller.PatientCaseController;
15  import ea.biju.repo.PatientRepository;
16
17  /**
18   *
19   * @author Biju Ale
20   */
21  public class UnitTestSIRModellingController {
22
23      @Mock
24      private PatientRepository patientRepository;
25
26      @InjectMocks
27      private PatientCaseController patientCaseController;
28
29      private MockMvc mockMvc;
30
31      @Before
32      public void setup() {
33          MockitoAnnotations.initMocks(this);
34          mockMvc = MockMvcBuilders.standaloneSetup(patientCaseController).build();
35          MockitoAnnotations.initMocks(this);
36      }
37
38      //All tests were run & verified with walkthrough and documented
39      //because test needs visual verification due to graphics rendering
40  }
41

```

EPIDEMIC ANALYZER

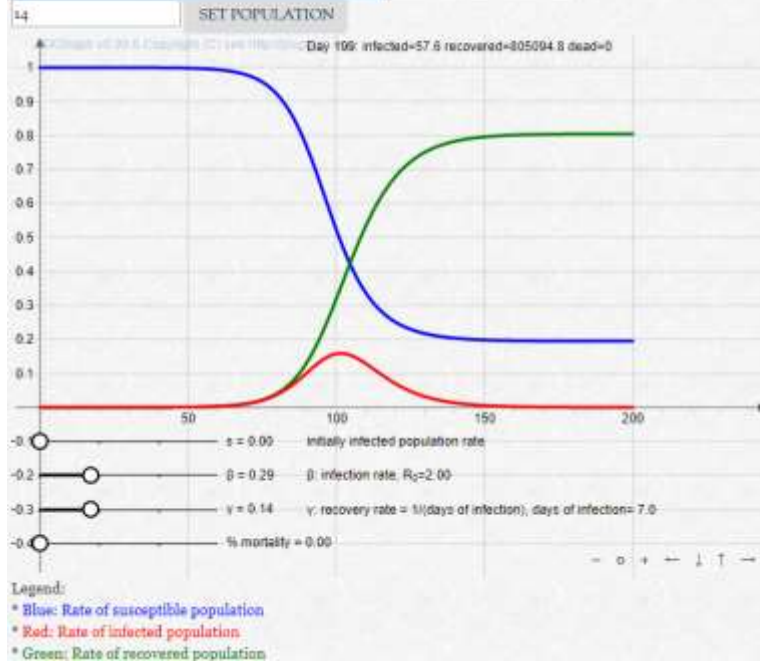
EPIDEMIC ANALYZER PATIENT CASE + SIR MODELLING USER MANUAL LOGOUT

SIR MODELLING

CLEAR AND RUN A SIMULATION OF 200 DAYS

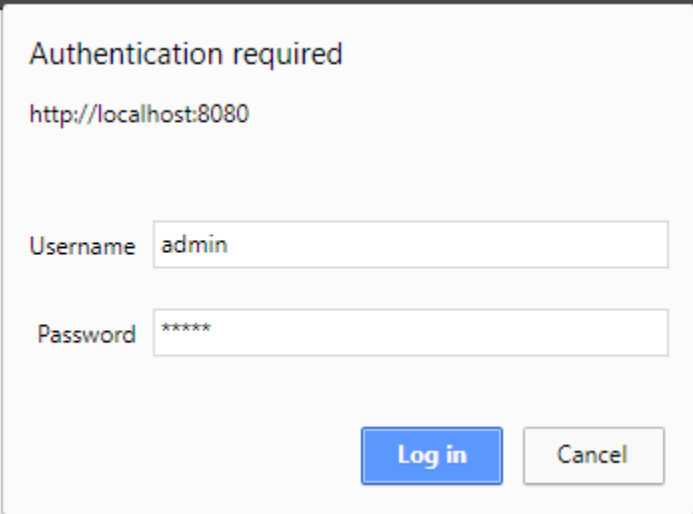
STOP

CONTINUE



User Manual

Logging in



The image shows a web browser's authentication dialog box. At the top, it says "Authentication required" in red. Below that is the URL "http://localhost:8080". There are two input fields: "Username" with the text "admin" and "Password" with masked characters "*****". At the bottom right are two buttons: "Log in" (blue) and "Cancel" (grey).

1. Acquire valid credentials from the site's owner.
2. Enter site url.
3. As shown in figure above, an input dialog box will appear.
4. Enter username and password in respective fields.
5. Click on 'Log in' button.

Adding new patient case

EPIDEMIC ANALYZER PATIENT CASE SIR MODELLING USER MANUAL LOGOUT

WELCOME!

In the milieu of a humanitarian crisis (e.g., epidemic), responders including public health personnels require epidemic data and its analysis. Such information is paramount for decision makers whose discernments and actions have direct impact on the control of the outbreak, consequently saving lives.

AIMS

- To build a system that can capture epidemic data on ground zero and provide analytics for provincial and national level decision makers.
- To contribute to the understanding of public health practitioners and humanitarian agencies about an epidemic situation through the introduction of information technologies.

OBJECTIVES

- To provide a robust data entry form that captures a broad range of health data dynamics.
- To provide a robust data entry form that captures a broad range of health data dynamics.
- To provide a graphical representation of data for better comprehension & decision making.
- To provide a graphical representation of data for better comprehension & decision making.
- To learn about the impact of information technology in international humanitarian interventions.
- To apply the principles from academic learnings to the real world scenarios and garner experience in research and development.

GET STARTED

- [Read the user manual](#)
- [Add new epidemic case](#)
- [View/Edit existing cases](#)
- [Project cases on map](#)

EPIDEMIC ANALYZER - A WEB-BASED EPIDEMIC DATA MANAGEMENT CUM ANALYSIS SYSTEM THAT AIDS HUMANITARIAN ACTORS IN INFORMATION ACQUISITION & ANALYSIS.

EPIDEMIC ANALYZER **PATIENT CASE** ▾ **SIR MODEL**

ADD NEW PATIENT CASE

WARD NO

GPS LATITUDE

GPS LONGITUDE

Male ▾

Susceptible ▾

ADD CASE

1. Enter site url and log in using valid credentials.
2. On the navigation bare, click on 'Patient Case'.
3. On the dropdown menu, click on 'Add new Patient Case'.
4. Enter patient case data on respective fields.
5. Click on 'Add Case' button.

EPIDEMIC ANALYZER

Displaying all patient cases

EPIDEMIC ANALYZER PATIENT CASE ▾ SIR MODELLING USER MANUAL LOGOUT

WELCOME!

ADD NEW PATIENT CASE
DISPLAY ALL PATIENT CASES
PROJECT CASES ON MAP

In the milieu of a humanitarian crisis (e.g., epidemic), responders including public health personnels require epidemic data and its analysis. Such information is paramount for decision makers whose discernments and actions have direct

EPIDEMIC ANALYZER - A WEB-BASED EPIDEMIC DATA

EPIDEMIC ANALYZER PATIENT CASE ▾ SIR MODELLING USER MANUAL LOGOUT

PATIENT CASE ENTRIES

SHOW 10 ▾ ENTRIES

Search...

PATIENT ID	CURRENT INFECTION STATUS	FIRST NAME	MIDDLE NAME	LAST NAME	DATE OF BIRTH	GENDER	DISTRICT	VDC	WARD NO	LAT.	LONG.	SELECT
13	RECOVERED	BIKISHA	S	SHRESTHA	1990-10-24	FEMALE	KATHMANDU	KAPAN	2	-27.840282	85.1207474	<input type="checkbox"/> UPDATE
14	SUSCEPTIBLE	PAUL	C	COPAN	1978-10-12	MALE	DOLAKHA	JIRU	2	-27.902791	85.1194658	<input type="checkbox"/> UPDATE
15	SUSCEPTIBLE	PABIN	MORELAND	ROLAND	1988-10-03	MALE	ASD	ASD	1	-27.879917	85.220449	<input type="checkbox"/> UPDATE
16	INFECTED	RADHA	KRISHNAN	ROSHAN	1974-10-03	FEMALE	ASD	ASDASD	3	-27.908317	85.263263	<input type="checkbox"/> UPDATE
17	RECOVERED	MANJUSHREE	THAPA	SHRESTHA	1986-11-01	FEMALE	[LKLK]	III	4			<input type="checkbox"/> UPDATE
18	SUSCEPTIBLE	PAUL	C	COPAN	2000-10-04	MALE	KATHMANDU	DHAPASI	9			<input type="checkbox"/> UPDATE
19	INFECTED	JOHN	L	LENNOX	1980-10-04	MALE	DHADING	SALYANTAR	6			<input type="checkbox"/> UPDATE
20	INFECTED	MARY	JO	SHARJ	1998-10-13	FEMALE	BHAKTAPUR	LOKANTHALI	8			<input type="checkbox"/> UPDATE
21	INFECTED	BLANK	T	TUREK	1993-10-11	MALE	KABHREPALANCHOK	WALTING	3	-27.828013	85.1274469	<input type="checkbox"/> UPDATE
22	INFECTED	JAMES	PORTER	MORELAND	1979-10-15	MALE	LALITPUR	SANKHU	1	-27.828013	85.1274469	<input type="checkbox"/> UPDATE

SHOWING 1 TO 10 OF 14 ENTRIES
PREVIOUS12NEXT

1. Enter site url and log in using valid credentials.
2. On the navigation bar, click on 'Patient Case'.
3. On the dropdown menu, click on 'Display all patient cases'.

EPIDEMIC ANALYZER

Searching and sorting results – patient cases

EPIDEMIC ANALYZER PATIENT CASE SIR MODELLING USER MANUAL LOGOUT

PATIENT CASE ENTRIES

SHOW 10 ENTRIES

PATIENT ID	CURRENT INFECTION STATUS	FIRST NAME	MIDDLE NAME	LAST NAME	DATE OF BIRTH	GENDER	DISTRICT	VDC	WARD NO	LAT.	LONG.	SELECT	
14	SUSCEPTIBLE	PAUL	C	COPAN	1979-10-12	MALE	DOLAKHA	BRI	2	-33.910751	151.194168	<input type="checkbox"/>	UPDATE
15	SUSCEPTIBLE	PAUL	C	COPAN	2000-10-04	MALE	KATHMANDU	DHAPASI	9			<input type="checkbox"/>	UPDATE

SHOWING 1 TO 2 OF 2 ENTRIES (FILTERED FROM 14 TOTAL ENTRIES)

PREVIOUS NEXT

DELETE SELECTED

1. Enter site url and log in using valid credentials.
2. On the navigation bar, click on 'Patient Case'.
3. On the dropdown menu, click on 'Display all patient cases'.
4. On the top-right area, in search box, input search string.
5. Click on any field's header on the results, to sort by that field.

Deleting patient case

WARD NO	LAT.	LONG.	SELECT	
2	-33.840282	151.207474	<input checked="" type="checkbox"/>	UPDATE
2	-33.910751	151.194168	<input type="checkbox"/>	UPDATE

SHOWING 1 TO 10 OF 14 ENTRIES

PREVIOUS NEXT

DELETE SELECTED

localhost:8080 says:

Confirm deletion?

OK Cancel

EPIDEMIC ANALYZER

1. Enter site url and log in using valid credentials.
2. On the navigation bar, click on 'Patient Case'.
3. On the dropdown menu, click on 'Display all patient cases'.
4. Tick on the check box of record that you want to delete.
5. Batch delete is supported. Click on multiple record's checkboxes.
6. On bottom left area, click on 'Delete Case' button.
7. Confirm deletion by clicking on 'OK' button as a confirmation dialog box appears.

EPIDEMIC ANALYZER

Updating patient case

WARD NO	LAT.	LONG.	SELECT	
2	-33.840282	151.207474	<input checked="" type="checkbox"/>	UPDATE
2	-33.910751	151.194168	<input type="checkbox"/>	UPDATE

EPIDEMIC ANALYZER

PATIENT CASE ▾

SIR MOI

UPDATE PATIENT CASE

William

Lane

Craig

Rasuwa

Langtang

2

GPS LATITUDE

Latitude

GPS LONGITUDE

Longitude

10/11/1971

Male ▾

Infected ▾

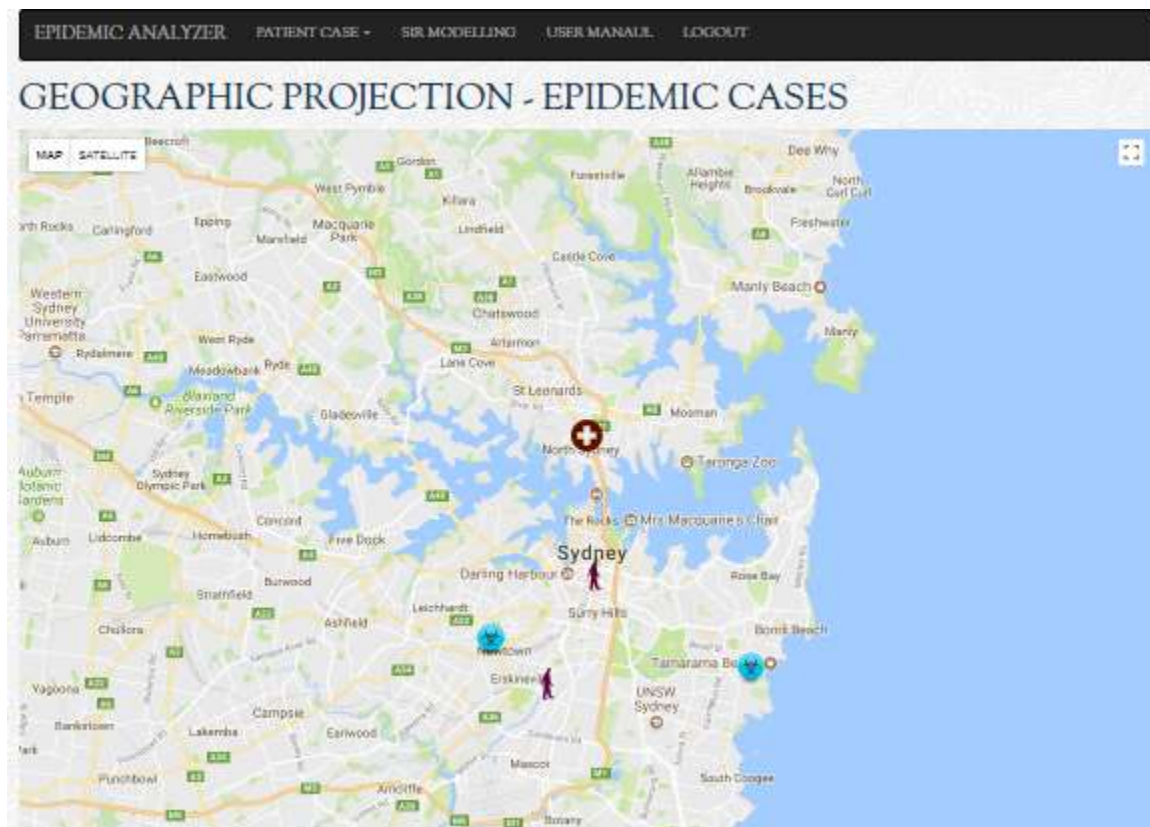
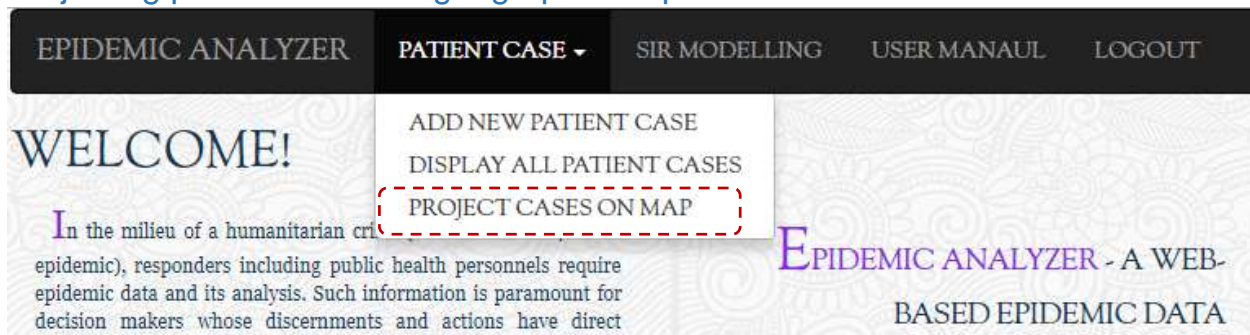
UPDATE CASE

1. Enter site url and log in using valid credentials.
2. On the navigation bar, click on 'Patient Case'.

EPIDEMIC ANALYZER

3. On the dropdown menu, click on 'Display all patient cases'.
4. Tick on the check box of record that you want to update.
5. On the update page, modify existing data in respective fields.
6. Click on 'update case' button.

Projecting patient cases on geographic map



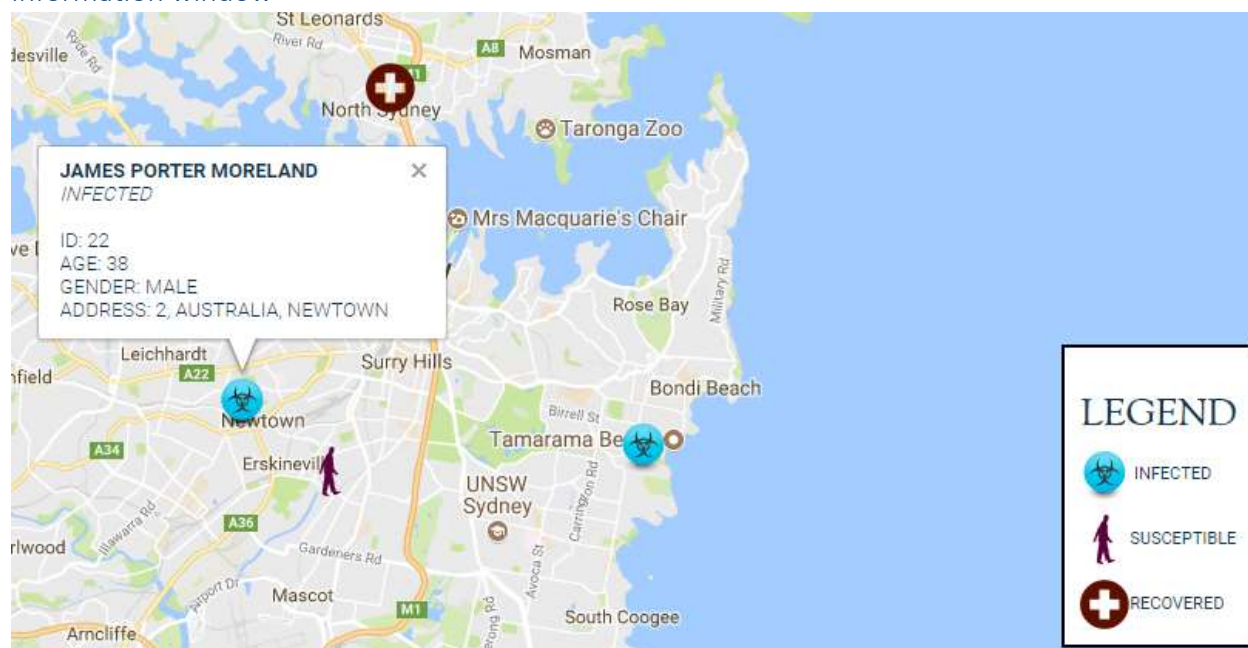
EPIDEMIC ANALYZER

1. Enter site url and log in using valid credentials.
2. On the navigation bar, click on 'Patient Case'.
3. On the dropdown menu, click on 'Project cases on map'.

Interacting with map

1. Enter site url and log in using valid credentials.
2. On the navigation bar, click on 'Patient Case'.
3. On the dropdown menu, click on 'Project cases on map'.

Information window

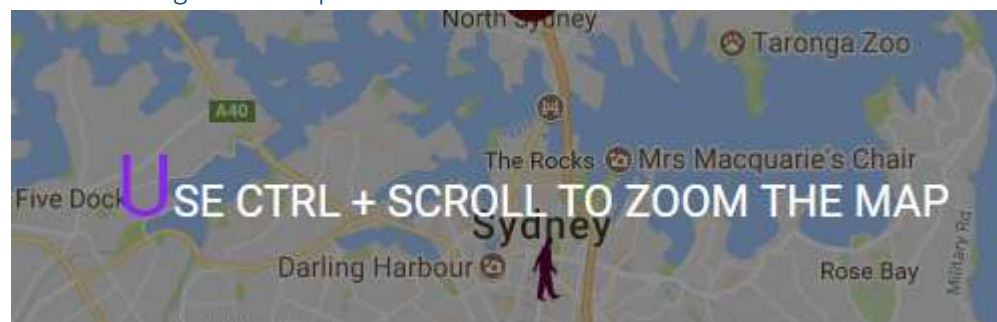


To view information on patient case on the map itself, click on the respective marker.

Legend

View legend at bottom right area, for marker legend.

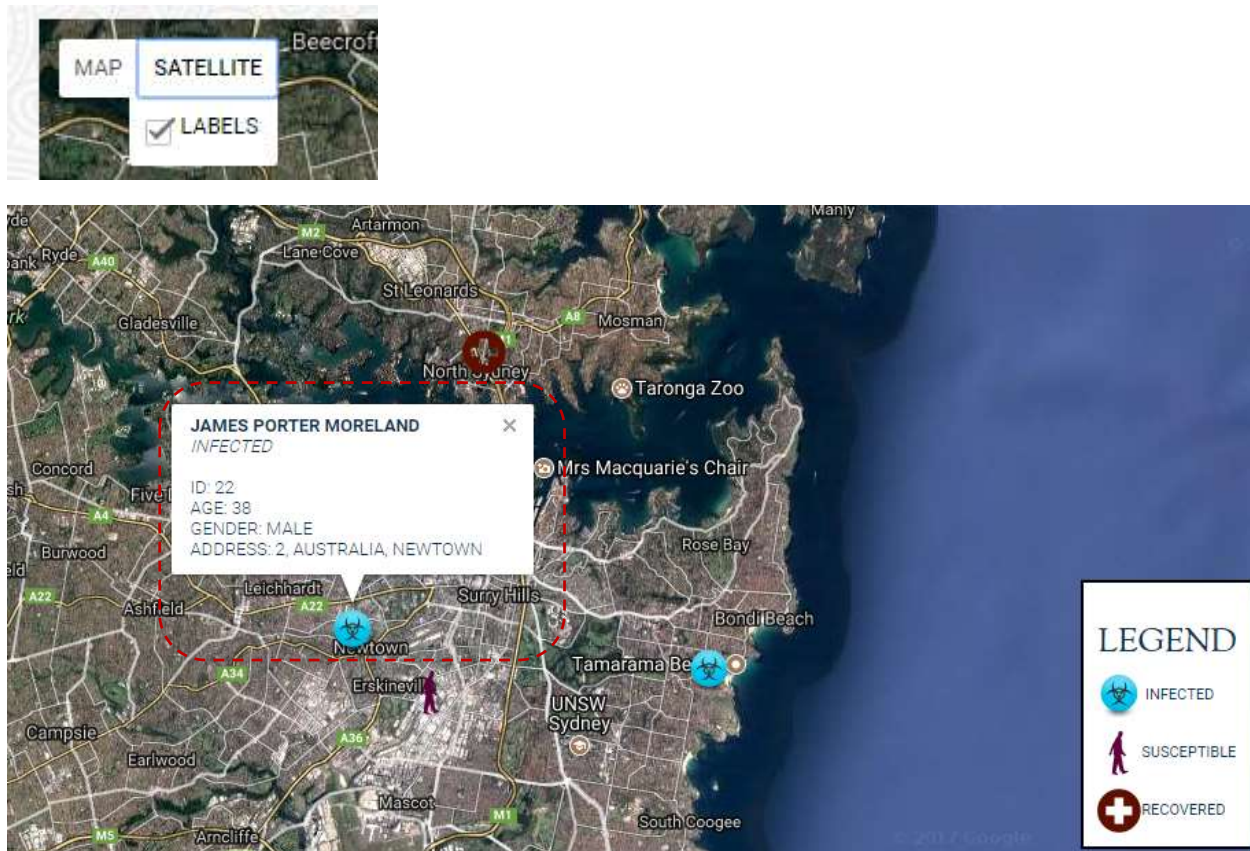
Zoom & navigate on map



Perform Ctrl+mouse-scroll to zoom in and out and drag pointer on map to navigate.

EPIDEMIC ANALYZER

Satellite view



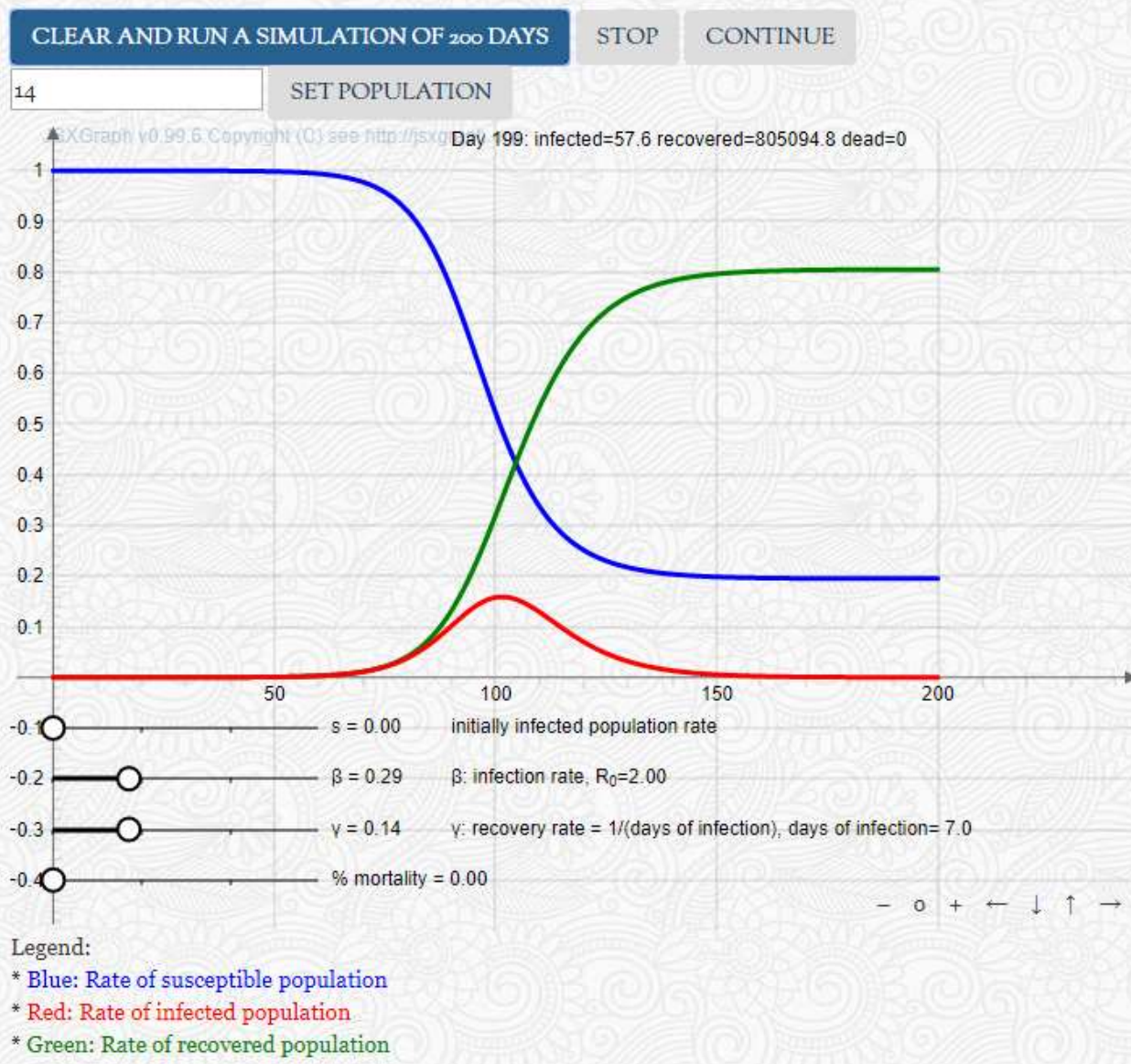
On the top left area of map, click on 'Satellite' to change terrain view.

SIR Modelling



EPIDEMIC ANALYZER

SIR MODELLING



1. Enter site url and log in using valid credentials.
2. On the navigation bar, click on 'SIR Modelling'.
3. Click on 'Set Population'.
4. Click on 'Clear and run a simulation of 200 days' to run the simulation on model.
5. Change parameters by using the respective slider-controls provided below the graph.