

# Visual Epidemic Simulation

Matthew Lakin

**Supervisor: Hossein Nevisi**

October 2024 - May 2025

# Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
<b>3</b>	<b>Problem Domain</b>	<b>4</b>
3.1	Minimum Viable Product . . . . .	4
3.2	Stretch Goals . . . . .	5
<b>4</b>	<b>Methodology</b>	<b>6</b>
4.1	Waterfall . . . . .	6
4.2	Gantt Chart . . . . .	8
<b>5</b>	<b>Research</b>	<b>9</b>
5.1	Existing Solutions . . . . .	9
5.1.1	COVID-19 Dashboard . . . . .	9
5.1.2	Plague Inc. . . . .	10
5.2	Compartmental Models . . . . .	10
<b>6</b>	<b>Technical Solution</b>	<b>11</b>
<b>7</b>	<b>Results and Analysis</b>	<b>12</b>
<b>8</b>	<b>Conclusion</b>	<b>13</b>
<b>9</b>	<b>Bibliography</b>	<b>14</b>
<b>10</b>	<b>Appendix</b>	<b>15</b>

# Chapter 1

## Abstract

"We've actually invested very little in a system to stop an epidemic. We're not ready for the next epidemic."

- Bill Gates, 2015 [4]

In recent years, we have all become too familiar with viruses and epidemics. From Ebola landing in Europe and America in the 2010s after ravaging through Africa, to Covid-19 in the 2020s. New epidemics seem to be springing up at a rate never seen before [5] and I think there are multiple causes to this. For one, access to medicine has been a huge help in increasing survival rates for all sorts of ailments. However, this has caused complacency in the world of medicine and Antibiotic Resistance, where medicine becomes ineffective to bacterial infections, has become one of the most prevalent issues the human race will face in the coming decades. Being able to use existing data to predict how epidemics will spread and what

Going back to the 14th Century, the Black Death, or the bubonic plague was one of the worst outbreaks in history. It killed between 75 and 200 million people in a few years time which was 30-50% of the total population of Europe at the time [3].

Smallpox still remains as the only human-affecting virus which has been functionally extinct as a result of human intervention [2]. This was a result of a combination of mass vaccination, containment and tracking. In this paper, I wanted to focus on the tracking aspect of containing diseases and how it can be a useful tool to not only teach us about what happened previously, but to predict and forecast what could happen in the future.

## Chapter 2

# Introduction

This project intends to create a visual simulation of an epidemic. Using publically available data from the COVID-19 epidemic, the simulation will represent the data in a visual format similar to a dashboard. The simulation will be created using a Dockerised Java environment, with the simulation itself being written in Javascript with node.js.

# Chapter 3

## Problem Domain

### 3.1 Minimum Viable Product

I have identified the minimal requirements for this project to be successful.

1. Front end
  - (a) The front end must show a map of the Earth.
  - (b) The map will render polygon and country data.
  - (c) The front end must show a timeline of the epidemic.
  - (d) The individual countries must be clickable.
  - (e) Clicking a country will show the number of cases and deaths per week.
  - (f) The user must be able to scrub through the timeline.
  - (g) The user must be able to select data from a CSV file to be displayed.
2. Back end
  - (a) The back end will be using a node.js server.
  - (b) The epidemic data will be stored in a csv file.
  - (c) The polygon data will be stored in a GeoJSON file.
  - (d) The node.js server will be dockerised.

## 3.2 Stretch Goals

For this project, the following aspirational requirements have been identified:

1. Zooming in on the map.
2. Hotspots on the map.
3. Predictctive modelling and forecasting.
4. Analysis of data with other data sources to search for correlations.

# Chapter 4

## Methodology

### 4.1 Waterfall

For this project I will incorporate a Waterfall model into my project since it will ensure my code and report as a whole is developed in a structured and ordered fashion.

The steps of Waterfall I will be following are:

#### 1. Requirements

- This section requires producing and refining a set of requirements. I plan on having 2 groups of requirements: Minimum Viable Product requirements and Stretch requirements.
- I plan on refining all the requirements in this stage so I can stay on this path plan.

#### 2. Analysis

- In this section, I will be doing research into the problems which are discovered by my requirement refinement. I will investigate papers and other relevant published sources.
- I also want to investigate node packages to represent my data graphically. (This would continue into the design section)
- This section will likely take up a large section of my report, and I may return to this section midway through the project.

#### 3. Design

- I would be designing the layout and colour scheme of my application as it would appear in a browser window. It is worth noting that different size screens would render it differently, so I need to take that into account when planning.
- I am not using a database for this project, but I will be using JSON and creating a standard design for objects in JSON will be beneficial.
- I need to decide what values I use for graphs and what kind of graphs to use.

#### 4. Coding

- I will be making a web application with node.js. This will require me to create a valid work environment and get the node runtime working on my local machine. I will likely use Docker for this since it is a very easy to work with tool perfect for this application.
- I will mostly be using TypeScript for this project because of type safety and native compatibility with web development. There will also be HTML and CSS for loading the TS canvas and aligning elements.

- Node.js also comes with node package manager which will also be useful for loading the custom packages I plan on using in this project.

## 5. Testing

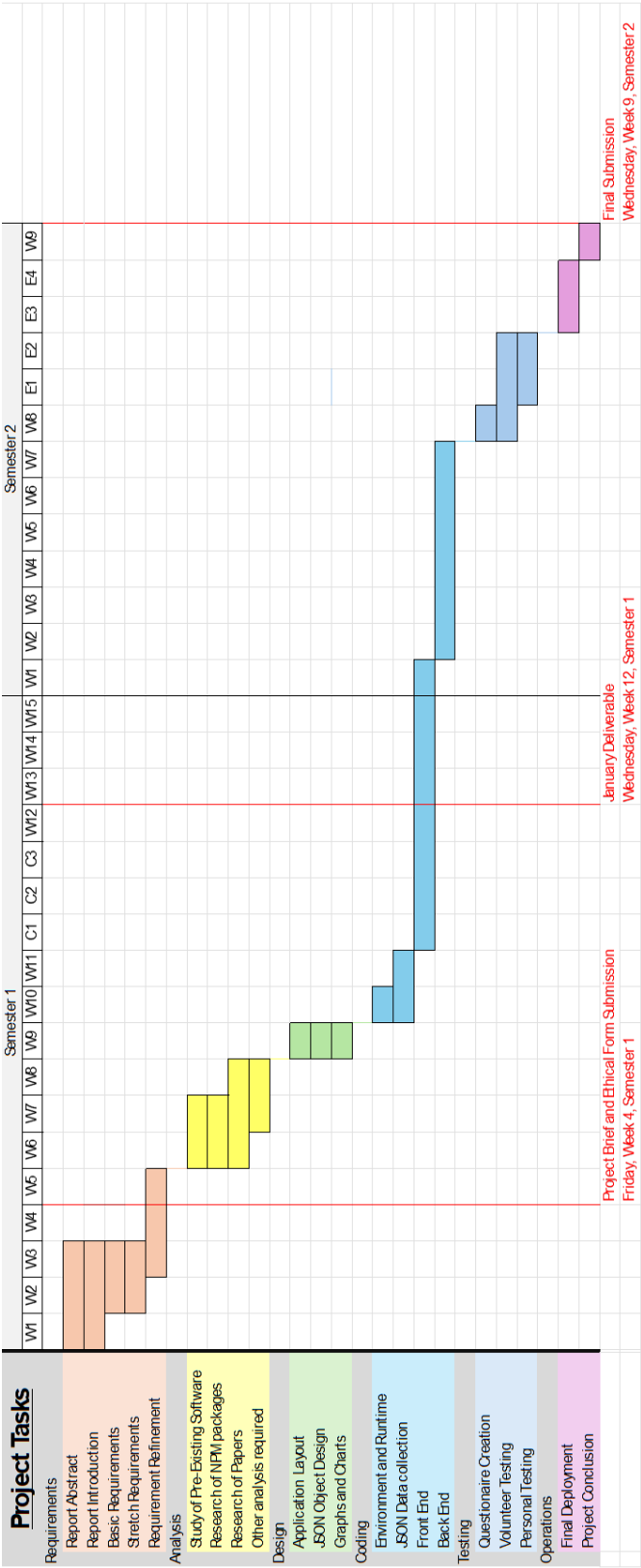
- For testing, I will be using a variety of normal, boundary and erroneous tests to check my software for crashes, unexpected results, and vulnerabilities. Using a type-safe language like TypeScript intends to minimise the chances of these happening.
- I also plan on having 3rd-party applicants to test my application. They would have a list of things to do in the application so I can analyse the ease of use.
- I would also ask them to fill in a questionnaire.

## 6. Operations

- I want to deploy my application to a server hosted by the university.
- Node.js applications are commonly deployed onto servers and having a production repository would be a good standard for myself. Pushing to a server from a local git repository is a highly transferable skill which is widely used. Test



4.2 Gantt Chart



## Chapter 5

# Research

### 5.1 Existing Solutions

#### 5.1.1 COVID-19 Dashboard

A common tool for data visualisation is a software called PowerBI, a Microsoft product which allows for the creation of dashboards. The dashboard in the figure shows a real-time representation of the COVID-19 pandemic.



Figure 5.1: COVID-19 Dashboard Example

This dashboard has a lot of information on it even at first glance. The map uses bubbles to denote hotspots of activity for the virus. The bubbles are sized by the number of cases in that country. The bubbles can also be clicked for more detailed information on that country. The user is also able to alter the time frame of the data, and the data is updated in real time.

### 5.1.2 Plague Inc.

Plague Inc. is a game published in 2012 by Ndemic Creations. It gives the user the ability to create and spread a virus across the world. The game was initially released on mobile devices and has since been released on PC and console.

Mobile games require accessible and easy to use interfaces, which is why Plague Inc. is a good example of a visual epidemic simulation. The game is simple to use and has a lot of information on the screen at once.



Figure 5.2: Plague Inc. Dashboard Example

Despite many systems being dramatised for the sake of gameplay, the game offers many features which could be useful for a real-world epidemic simulation.

The game uses a world map which gradually changes colour as the virus spreads, this is from many red dots appearing on the map. The user also has access to individual breakdowns of each country and can see the number of cases and deaths in each country.

The game also has a transmission system which changes from country to country.

## 5.2 Compartmental Models

Compartmental models are a type of mathematical model used to represent the different populations in a system. The most common compartmental system in epidemiology is the SIR model. The SIR model is a simple model which divides the population into three compartments: Susceptible, Infected and Recovered. The model is represented by the following differential equations:

$$\frac{dS}{dt} = -\alpha SI \quad (5.1)$$

$$\frac{dI}{dt} = \alpha SI - \beta I \quad (5.2)$$

$$\frac{dR}{dt} = \beta I \quad (5.3)$$

## Chapter 6

# Technical Solution

## Chapter 7

# Results and Analysis

## Chapter 8

## Conclusion

## Chapter 9

# Bibliography

- [1] Ross Beckley, Cametria Weatherspoon, Michael Alexander, Marissa Chandler, Anthony Johnson, and Ghan S Bhatt. Modeling epidemics with differential equations. *Tennessee State University Internal Report*, 2013.
- [2] Centers for Disease Control and Prevention. Smallpox, 2023. Accessed: 2023-10-01.
- [3] Sharon N DeWitte and James W Wood. Selectivity of black death mortality with respect to preexisting health. *Proceedings of the National Academy of Sciences*, 105(5):1436–1441, 2008.
- [4] Bill Gates. The next outbreak? we’re not ready, March 2015.
- [5] Serge Morand, Muriel Figuié, Frédéric Keck, Claude Gilbert, Nathalie Brender, François Roger, and Patrick Zylberman. *Emergence of infectious diseases*. Quae Editions, April 2018.

Chapter 10

Appendix