A picture containing text, sign

Description automatically generated

College of Computing

Computer Science Department

CS3141 Team Software Project

Fall 2023

**Simulation Project**

Section: R01

Team #: Team 12

| Name | Role |
| --- | --- |
| Will | Scrum Master |
| Noah | Developer |
| Ben | Developer |
| Caleb | Developer |
| Chance | Developer |

Instructor:

Serein AL-Ratrout

# **Abstract**

(***Abstract*** *is one paragraph that summarizes your project, and describes the content and scope of the project objective, methodology, findings, and conclusion. So, you need to write one paragraph that gives an abstract idea about the entire project, the aim of the project, the process model you used, the tools, what you have done, the results, and your conclusion. If you think the project is worth extending to a Final Year Project (FYP) by you or other students or can be adopted and extended by industry/market, then mention that here and add it also as future work.)*

Example of abstract

In this project a student registration web application for university students and staff was developed, the aim of this application is to provide a simple set-up of programs for student enrolment, improve efficiencies and eliminate unnecessary paperwork. The system mainly has two modules: students and staff. Students can create an account then sign in/out, add, update, delete, and modify their data and schedule. Staff can also create accounts and then sign in, add, update, delete, and modify their data. The scrum agile process model is followed during project development and the implementation was realized by the use of object-oriented PHP, HTML, MYSQL, and Dreamweaver technologies.

It has been found that the final system was simple and user-friendly with an easy user interface, hence the end-users do not need to undergo extensive training or require any special skills. It was also secure, and reliable.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Table of Contents**

# **Table of Figures**

Chapter1

Specification

# **1.1 Introduction**

The program will be an application that allows the user to simulate the spread of disease. We will have a map where the user can decide where an outbreak of an infectious disease begins, as well as allowing for the user to change various factors of the disease such as infectivity, and potentially mortality rates. Time will progress and report the current statistics of each state on the map. The goal is to teach the user more about how disease can spread, and is especially relevant in a post-COVID world where much of the spread can be attributed to lack of understanding of the danger and rate of spread in a community.

# **1.2 Problem Statement**

The program will show a list of all 50 states and the number of diseased-infected individuals. The problem is the lack of knowledge as to the importance of preemptive measures to prevent disease spread. One example of this might be the Covid-19 pandemic. The user will pick the state the disease starts in and can watch the disease's infected population grow and by which state. The purpose of this is to show the end user how important it is to use preemptive measures.

# **1.3 Aim and Objectives**

Aim:

The aim of this project is to create an application to teach students and the general public about the spread of infectious disease. It will allow the user to customise a disease and watch it spread through a map, being able to read statistics about each state as time progresses.

Objectives:

* To allow a user to customise a disease
* To allow users to pick a disease starting location
* To report information on each state with each time update
* To simulate the spread of a disease over time
* To provide overall statistics after the simulated time period.
* To allow users to save statistics or rerun the simulation with the same outcome.

# **1.4 Stakeholders**

There are a few different stakeholders for a project like this.

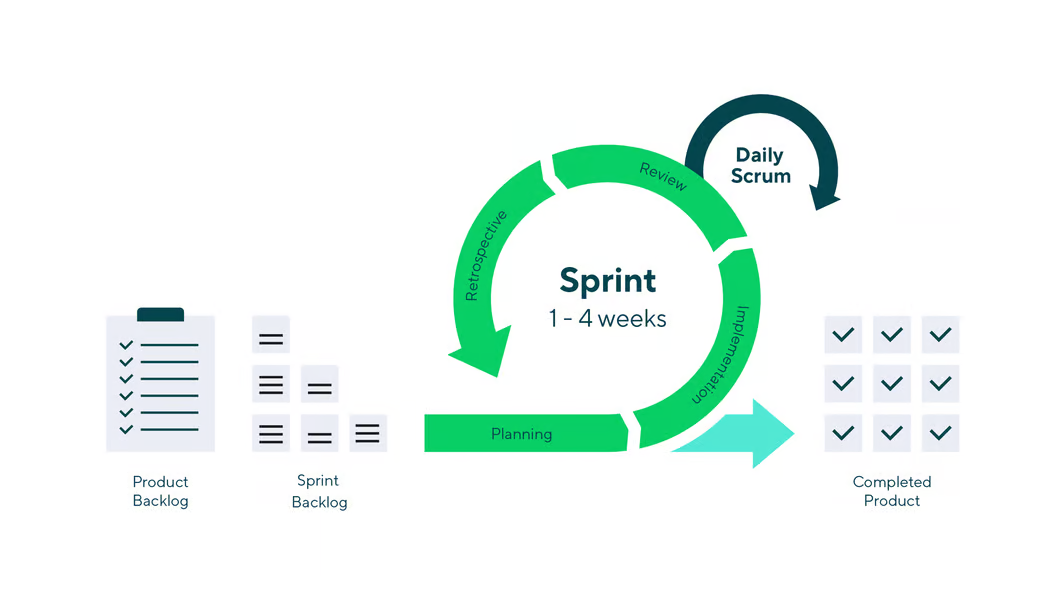
* The CDC
* Hospitals
* Doctors offices
* Pharmacies
* Patients
* Employers
* Insurance companies
* The Government

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **1.5** Methodology [ben]

We will be following the scrum software process, with 4, 2-week sprints. First, we will produce a product backlog of prioritized user requirements for our project. Then we will divide these up into sprints. For each sprint, we will delegate tasks and hold daily meetings, planning meetings at the beginning, and retrospective meetings at the end of each sprint. Each sprint will last 2 weeks.

After each sprint, we will update our backlogs moving what we completed to finished and what we didn’t complete back into the product backlog before starting the next sprint. At the end of all of the sprints we will have completed our software process and will have our final product



# **1.6 Tools**

(Mention the Software and/or hardware you will need to develop your project)

* Software
  + An IDE for Java
  + Junit, or other testing framework
* Hardware
  + Computers to develop software on

# **1.7 High-Level Business Requirements**

## **Functional Requirements**

* + Simulate viral spread in population
  + Simulate viral spread between populations
  + Store viral spread data at regular intervals

## **Non-functional requirements**

* + User defined parameters for viral spread
  + Graphical depiction of viral spread

# **1.8 Product backlog [ben]**

(List of Prioritized user requirements written in user story format.)

**You can use the following table:**

| **Priority** | **User Story** | **Tasks** | **Estimated effort** | **Sprint** |
| --- | --- | --- | --- | --- |
| **\*\*\*** | **As a User, I want there to be multiple States with different populations, So that the disease can spread** | **Implement State class** | **1h** | **1** |
| **Implement population parameter and a function to get the population** | **~** |
| **\*\*\*** | **As a User, I want there to be a count of infected people at each State with logistic curve change over time, So that I can simulation disease change** | **Implement an infected parameter** | **~** | **1** |
| **Implement a time step function that updates the disease following a logistic curve** | **1h** |
|  |  |
| **\*\*** | **As a User, I want the States to have locations relative to each other with neighbors, So that later disease can be modeled to spread between them** | **Implement a graph data structure** | **2h** | **1** |
| **Implement a function to determine the neighbors** | **1h** |
| **\*** | **As a User, I want to be able to see the relative locations of the States in an understandable output, So that I can understand how they are located** | **Implement a way to display the layout either textually or visually** | **3h** | **1** |
| **\*\*\*** | **As a User, I want disease to spread from one State to other States based on their locations, So that disease spread can be modeled** | **Implement a spread function that infects neighboring States** | **3h** | **1** |
| **Implement a system for updating all of the States at once** | **4h** |
| **\*\*** | **As a developer, I want a crude way to print the number of infected people in each State, So that I can fix bugs while developing** | **Implement function to print the numbers in a semi readable format** | **2h** | **2** |
| **\*\*\*** | **As a user, I want a polished way of seeing the infected numbers in each State, So that I can understand the disease simulation** | **Implement a function to print the numbers in a polished format** | **3h** | **3** |
| **\*** | **As a User, I want a way to export the numbers in each state, So that I can see the disease simulaton data after it was already run** | **Implement a function to export the numbers in a polished format** | **3h** | **3** |
| **\*\*\*** | **As a User, I want a way to change the simulation parameters, So that I can see how the simulation would run under different circumstances** | **Implement a user way to change the starting location** | **2h** | **4** |
| **Implement a user way to change the infection rate** | **1h** |
| **Implement a user way to change the spread rate across states** | **1h** |
| **\*** | **As a User, I want a way to change the States and locations, So that I can see how the simulation would run under different circumstances** | **Implement a way for the user to import a different graph structure** | **4h** | **4** |

* 1. **Security Requirements engineering practice(s)**

***[you can apply one or more of the following practices]***

1. **Establish Security and Privacy Requirements**
   * **Identify key milestones and deliverables**
   * **Assign security experts**
   * **Define minimum security and privacy criteria for an application**
   * **Deploying a security vulnerability tracking system**
2. **Create Quality Gates/Bug Bars**
   * **Define minimum acceptable levels of security and privacy quality**
3. **Perform Security and Privacy Risk Assessments**
   * **Identify the need for threat modeling and security design reviews**

Chapter 2

Analysis and Design

For chapter 2 analysis and design,

Analysis specifies what the system should do, and design is the process of planning a new system or replacing an existing system by defining its components or modules to satisfy specific requirements.

In this chapter, you need to draw the following UML diagram + textual format to explain them  
  
UML use case diagram + textual formal to explain it  
  
UML class diagram +  textual formal to explain it  
  
UML sequence diagram + textual formal to explain it  
  
You can keep revisiting and updating this chapter (updating the design for the proposed system) during project development since the scrum software process allows this.

* **Security Design practice(s)**

***[you can apply one or more of the following practices ]***

1. Establish Design Requirements
   * Address security and privacy concerns early to reduce a project's expense.
   * Validate all design specifications against a functional specification
2. Perform Attack Surface Analysis/Reduction
   * The points attacker can use to break your system
   * Analyze the overall attack surface (points)
   * disable or restrict access to system services
   * apply the principle of least privilege
   * Employing layered defenses
3. Use Threat Modeling

* Simulate attack scenarios.

Chapter 3

Implementation

For the implementation section, you add screenshots for your implementation, and interfaces of your product/ project with simple statements to explain how users will use them.

Keep revisiting and updating this chapter during project development since the scrum software process allows this. (updating with screenshots for the new increments/components )

**Security implementation practice(s)**

***[you can apply the following practice]***

1. Deprecate Unsafe Functions

* Analyze all project functions to ban unsafe ones
* Replace the unsafe functions with safer alternatives

Chapter 4

## Validation

**For Chapter 4 (Validation)**: here you need to write about the process of checking that your software system meets specifications and requirements so that it fulfils its intended purpose, and to confirm or prove the accuracy of your project.

Write about your testing and validation; **level of testing** you had, unit testing, integration testing, validation testing, and acceptance testing.   Did you have **manual or automated** testing or both? specify the part(s) that have automated testing and part(s) that have manual testing, and **What is your oracle?**

**Write the** **test cases** for valid and invalid **input** (please see Week3 Automated Testing/ slide 11),

then confirm that no errors in the code that the application is able to operate in the required condition (OS, web browsers) and that you have created the code correctly.

For validation and acceptance testing write who tested your system. MTU students? computer science student? other department students? your group only? other college students? public users? How many students/users? How many times? could they use it easily or did they make mistakes?

You keep revisiting and updating this chapter during project development since the scrum software process allows this. (updating with test cases/ test suites you have done for the new increments/components)

Create a **testing document** with a ***comprehensive list of actions to be performed and the expected results of those actions***



**Security validation practice(s)**

***[you can apply one or more of the following practices]***

1. Perform Dynamic Analysis
   * Perform run-time verification of the software
   * Penetration test.
2. Perform Fuzz Testing: an automated software testing technique that attempts to find hackable software bugs by randomly feeding invalid and unexpected inputs and data into a computer program in order to find coding errors and security loopholes.
3. Conduct Attack Surface Review

* Review attack surface upon code completion
* Ensure that any new attack vectors created as a result of the changes have been reviewed and mitigated including threat models

Chapter 5

## Limitations and Future Work

**For Chapter 5**

Limitations: address everything that the project left,  if some project backlog items/ features/ requirements have not been implemented then mention them in this part with an explanation/justification why you couldn't implement them (Time constraints the time was not enough, some developers were unavailable, because of COVID19, or  tool limitation ....etc.). Many students tend to feel that presenting the limits of their work makes work weaker. on the contrary, approaching this section shows maturity for the academic universe, and writing about them actually strengthens your work by identifying any problems before reviewers/readers find them.

Future work : if the limitations can be addressed in the future then add this in here in future work, moreover, if you believe this project can be extendable (add more features/ more parts) that the project is worth extending to a Final Year Project (FYP) by you or other students or can be adopted and extended by industry as a product so you can give directions for that in future work.

Chapter 6

Conclusion

**For  Chapter 6 (Conclusion),**: write what you have concluded.

Examples:

I solved many problems in the project…

This application/project/system was applied to improve the learning process.

The results of this project showed that the system significantly facilitated the students' learning process.

The system is useless, acceptable, usable, beneficial, or maybe enjoyable and why do you believe that?

# References

(Include any references to external documents or materials (for example, tutorials the team will be using, literature , web references or links to documentation of third-party tools you will use) here.

The references should be properly numbered and correctly used in the text.

The Reference section should be in the following fashion:

# References

Journal, Magazine/ Newspaper Article

| [1] | Author, "Title," *Journal name,* p. pages, year.  Book |
| --- | --- |
| [2] | Author, Book Title, publisher, year.  Internet Web page: |
| [3] | Author, "Name of the Web Page," [Online]. Available: URL. [Accessed Date]. |

# Appendix:

**Security Release practice(s)**

1. Create an Incident Response Plan

* Prepare an Incident Response Plan to address new threats that can emerge over time
* identify appropriate security emergency contacts
* establish security servicing from other groups

1. Conduct Final Security Review

* Review all security activities that were performed to ensure software release readiness. Examine threat models, tools outputs,and performance against the quality gates and bug bars defined during the Requirements Phase

1. Certify Release and Archive

* Certify software prior to a release to ensure security and privacy requirements were met
* Archive all specifications, source code, binaries, private symbols, threat models, documentation, emergency response plans, and license and servicing terms for any third-party software.