**Software Implementation and Testing Document**

**For**

**Group Germ Theory**

Version 1.0

**Authors**:

Hector Rizo

Adam Soccer

Andrew Koelsch

# Programming Languages (5 points)

*List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).*

* Python -> this language has support for data collection and visualization. It is intuitive and easy to read so we chose it as our primary language of implementation.

# Platforms, APIs, Databases, and other technologies used (5 points)

*List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).*

* Numpy -> for data manipulation and array functions
* Matplotlib -> for data visualization
* Possible API’s include: BeautifulSoup for webscraping, WHO api for gathering data
* Basic java script to show the results of our implementation

# Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the* ***functional requirements*** *listed in your RD).*

* A grid was created for the for the cell. It was tested via console output. It was possible to pass in the game board into a Cell object and to have the Board object mutated by its representative Cell object.
* There were issues getting correct seeding for random number generation.

# Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the* ***non-functional requirements*** *listed in your RD).*

# Non-Execution-based Testing (10 points)

*Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).*