# A picture containing text, clipart Description automatically generated CSC 431 Coronapyrus Software Requirements Specification (SRS)

**Team 11**

|  |  |
| --- | --- |
| Alexander Claman | <Role> |
| Noah Jaccard | <Role> |
| James McSweeney | <Role> |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1.0 | 2.23.2021 | Team 11 | Rough Draft #1 |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

Version History 2

Table of Contents 3

1. System Requirements 4

1.1. Functional Requirements 4

1.1.1. User Request Processing 4

1.1.2. COVID Information Retrieval 4

1.1.3. Data Processing and Visualization 5

1.2. Nonfunctional Requirements 5

1.2.1. Request Scope 5

1.2.2. Request Format 5

2. Constraints 6

2.1. Tool Constraints 6

2.1.1. Required Python Packages 6

2.2. Language Constraints 6

2.2.1. Python Constraint 6

2.3. Platform Constraints 6

2.3.1. Python Package Management Platform 6

2.4. Network Constraints 6

2.4.1. Request COVID Information 6

2.5. Deployment Constraints 7

2.5.1. Python Environment 7

2.6. Budget and Schedule Constraints 7

2.6.1. Time Constraint 7

2.6.2. Funding Constraint 7

3. Requirements Modeling 8

3.1. User Request Processing 8

3.2. Class Diagram 9

# System Requirements

## Functional Requirements

### User Request Processing

|  |  |
| --- | --- |
| ID | FR1 |
| Title | User Request Processing |
| Description | A user request must be parsed. Any information with a user-defined scope must be retrieved, processed and/or visualized, and returned in a user-defined format. |
| Priority | 0 |
| Precondition(s) | User made a request for COVID information. |
| Basic Flow | Parameters associated with the user request define scope and format.  The scope parameter may include range of dates over which information is required, location of information, and type of information (media/news or data/characteristic).  The information within the given scope is retrieved.  The format parameter may denote returned information as an article link & summary for media, a table for numerical data, or a graph for data series.  The retrieved information is processed and visualized in the format requested. |
| Postconditions(s) | Information in the requested scope and format is returned to the user. |
| Use Case Diagram | 3.1. User Request Processing |

### COVID Information Retrieval

|  |  |
| --- | --- |
| ID | FR2 |
| Title | COVID Information Retrieval |
| Description | Once a user request is made, relevant COVID information must be gathered from reliable sources for the user. |
| Priority | 0 |
| Precondition(s) | User made a request for COVID information within a particular scope. |
| Basic Flow | The most recent available COVID information is gathered and stored using the newsfetch and Pandas packages if it has not been already.  The information within the scope of the user's request is retrieved. |
| Postconditions(s) | Information within the user-provided scope is made available for processing. |
| Use Case Diagram | 3.1. User Request Processing |

### Data Processing and Visualization

|  |  |
| --- | --- |
| ID | FR3 |
| Title | Data Processing and Visualization |
| Description | COVID information returned after a user request is made must be processed so it can be effectively visualized as a message, graph, or table. |
| Priority | 0 |
| Precondition(s) | User made a request for COVID information within a particular scope and format.  COVID data matching the scope have been retrieved and are available for processing. |
| Basic Flow | COVID information is processed based on the provided format.  Information from news articles is returned as a list of dictionaries, with each dictionary holding a summary of the article, the title of the article, and a link to the article.  Information from JHU COVID data can be returned multiple ways. It can be returned as a Pandas DataFrame containing the requested data, or a visualization of data can be created with Matplotlib and returned. |
| Postconditions(s) | Information in the user-defined format is returned. |
| Use Case Diagram | 3.1. User Request Processing |

## Nonfunctional Requirements

### Request Scope

|  |  |
| --- | --- |
| ID | NFR1 |
| Title | Request Scope |
| Description | A data structure, object, or class will be designed such that a user can properly define the scope of their request. |
| Priority | 1 |
| Applicable FRs | FR1, FR2 |

### Request Format

|  |  |
| --- | --- |
| ID | NFR2 |
| Title | Request Format |
| Description | A data structure, object, or class will be designed such that a user can properly define their desired response format/medium. |
| Priority | 1 |
| Applicable FRs | FR1, FR3 |

# Constraints

## Tool Constraints

### Required Python Packages

|  |  |
| --- | --- |
| Title | Required Python Packages |
| Description | The Pandas, Matplotlib, and newsfetch packages must be installed for Coronapyrus to function properly. |
| Priority | 0 |

### Data Availability

|  |  |
| --- | --- |
| Title | Data Availability |
| Description | Any numerical data used will be retrieved from publicly available John Hopkins University databases ([https://github.com/CSSEGISandData/COVID-19/tree/master/csse\_covid\_19\_data](https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data" \t "_blank)). If this data becomes unavailable, an alternative source of COVID data will need to be used. |
| Priority | 0 |

## Language Constraints

### Python Constraint

|  |  |
| --- | --- |
| Title | Python Constraint |
| Description | The only supported language for the Coronapyrus package at this time is Python for both development and use. |
| Priority | 0 |

## Platform Constraints

### Python Package Management Platform

|  |  |
| --- | --- |
| Title | Python Package Management Platform |
| Description | Independent of operating system, a Python package management strategy or system (such as pip) is required. |
| Priority | 0 |

## Network Constraints

### Request COVID Information

|  |  |
| --- | --- |
| Title | Request COVID Information |
| Description | A proper connection to the network is required to download recent COVID information. |
| Priority | 0 |

## Deployment Constraints

### Python Environment

|  |  |
| --- | --- |
| Title | Python Environment |
| Description | The Coronapyrus package will be retrievable from the pip package manager and from Github. It will be deployable in any Python development environment. A Python distribution such as Anaconda is required to create applications or scripts using Coronapyrus. |
| Priority | 0 |

## Budget and Schedule Constraints

### Time Constraint

|  |  |
| --- | --- |
| Title | Time Constraint |
| Description | This project must be completed by the end of the Spring 2021 University of Miami school semester. |
| Priority | 0 |

### Funding Constraint

|  |  |
| --- | --- |
| Title | Funding Constraint |
| Description | There is no funding for this project being sourced from a client at this time; as such, this will be open source. |
| Priority | 5 |

# Requirements Modeling

## User Request Processing

|  |  |
| --- | --- |
| Name | User Request Processing Use Case |
| Description | This is the primary use case for the Coronapyrus package. |
| Actors | The User. |
| Trigger | This use case is initiated when the User requests information about COVID-19. |
| Precondition(s) | None. |
| Basic Flow | 1. The User’s request is processed by the Coronapyrus package and any needed data or media is retrieved based on the User’s defined scope. 2. The data or media retrieved are processed into responses or visualized then converted into responses based on the User’s defined format. 3. A response is returned to the User containing the information within the scope they defined, in the format they requested it in. |
| Exceptions | The Coronapyrus package will raise errors if the requested data cannot be found or cannot be retrieved from the network. This includes requests for data that does not exist (such as COVID statistics from future dates) or a lack of internet connection. |
| Postcondition(s) | The User has received a properly formatted response containing the information they requested. |

Figure 1 – User Request Processing Use Case Diagram (Rough Draft)

Diagram

Description automatically generated

## Class Diagram

Figure 2 – Class Diagram (Rough Draft)

Diagram

Description automatically generated