**PATIENT101**

**Logo, company name

Description automatically generated**

**Sowmya Talasila, Noah Kenton, Aasri Inuganti**

**Special Topics: Data and Information**

**Dmitriy Babichenko**

**December 5th, 2022**

**Table of Contents**

Introduction . . . Page # 3

ER Model Page # 4

Business Rules . . . Page # 5 to Page # 6

Class Diagram Page # 7

Conclusion . . . Page # 8

**Introduction:**

For the project we chose to create a database about COVID-19. This is because it is one of the most important problems we have faces in the past and we thought finding out more information about it would be helpful. Ever since the pandemic started, we have found out lots of information about COVID-19, but no one knows for sure about what to expect when you get exposed to the disease and what factors might make the symptoms better or worse. We created this database as an attempt to dive deep, find out the patters related to this disease, and share it with other people. This will also give the doctors a little more information about what to look out for when a patient sees them regarding the disease.

The target audience for this database are the patients that tested positive and health care officials as well since they will be looking after patients with severe symptoms. We chose this particular group of audience because we thought they would find this information the most helpful and it is also a good source to keep track of how the disease is affecting each patient and how it is progressing.

**Some of the benefits of this database can include:**

* Lesser anxiety in patients as they have answers to a few questions.
* More knowledge for health care officials about how the disease is affecting individuals.
* Knowledge to the general public about the severity of the disease itself.

**UML-COMPLIANT ER MODEL**

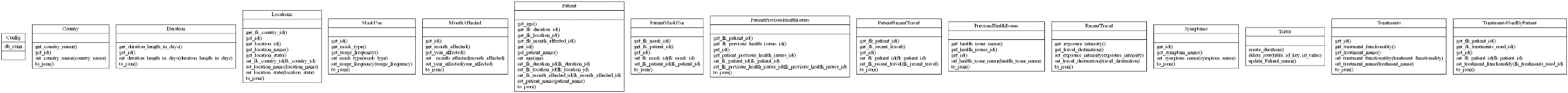
**Timeline

Description automatically generated**

**Business Rules:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity 1** | **Entity 2** | **Cardinality on Entity 1 side** | **Cardinality on Entity 2 side** | **Business Rule(s)** |
| patient | symptom | 1..\* | 1..\* | A patient can have many symptoms. A symptom can be experienced by many people. |
| patient | locationz | 1..\* | 1 | A patient can only be on one location at a time. A location could have many patients at one time. |
| locationz | country | 1..\* | 1 | A specific location can only be in one country. One Country can have many locations. |
| patient | duration | 1..\* | 1 | A patient can only have one duration for recovery. One specific duration can be experienced by many people. |
| patient | month\_affected | 1..\* | 1 | One patient can only be affected once in a month. A month could have many patients that were affected. |
| patient | previous\_health\_issues | 1..\* | 1..\* | A patient could have many health issues. A health issue can be experienced by multiple people. |
| patient | treatments | 1..\* | 1..\* | A patient could use multiple treatments when affected by the disease. A treatment can be used by multiple people while recovering. |
| patient | Masks\_use | 1..\* | 1..\* | A patient can use multiple mask types and one mask type can be used my multiple patients |
| patient | Recent\_travel | 1..\* | 1..\* | A patient could have travelled to multiple locations and one location could have multiple patients visited. |

**Class Diagram**

****

**(A clear picture is in the github repository)**

**Conclusion:**

Overall, this was a very interesting project. It gave us an opportunity to really think about how we can apply what we have learned throughout the semester in a real-world scenario. we had trouble in the beginning as we did not know where to start. But after a lot of thinking and doing a rough sketch about what we wanted our database to look like, we were able to get started. Even though this project was very long and time consuming, the outcome of it was really worth is. One of the most difficult things we had to do was implementing views and transactions. It might seem simple, but we never had experience with it and we had to do some research to figure out how to implement them.