# **Block Diagram - CoviDisclosure**

# Group VB 08

# Vignati Yalamanchili: 20

# Andrew Wilken: 30

# Siddharth Rana: 35

# Shagun Bansal: % 15

**Block Diagram**

Graphical user interface

Description automatically generated

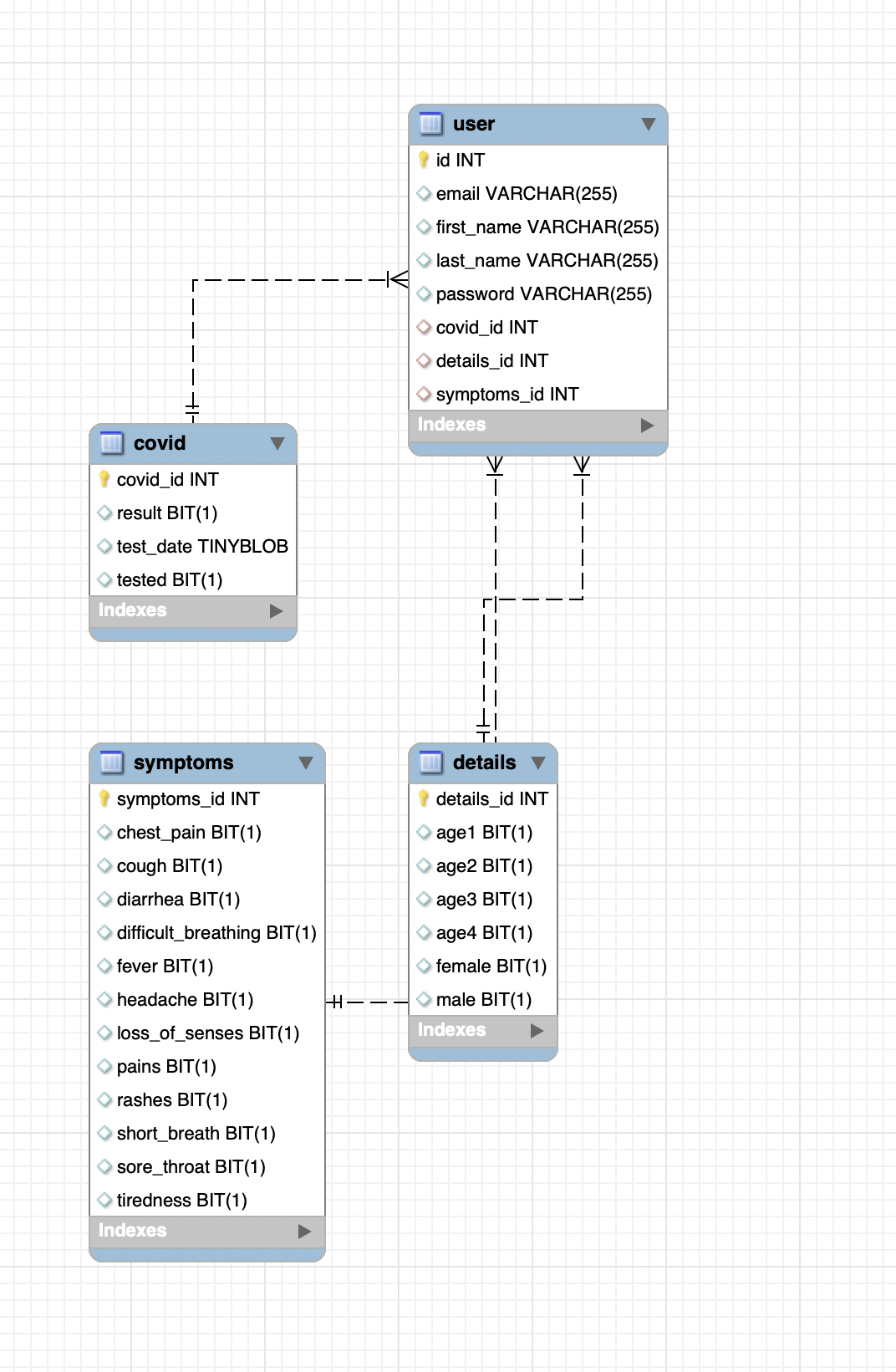
**Design Description**

Our UI consists of multiple activities, that run independently or connect with an api. We begin by Login and Registration. That information will be passed between the front and backend. Once a user is created, when logging in, the combination will be checked with the database and receive a unique identifier. A user may then be allowed to view statistics or submit their own. Their description is submitted to the backend via url JSON POST request. A user may also find hospitals near them through the use of a google maps api or talk to a chatbot that can answer simple questions for the user. When viewing statistics they will be met with statistics for them. Helper methods, receive user description, and compare it with all users stats that match that description with JSON GET request. At any point, a user may decide to update their information. They can reset their password, email, update their test status, and duration of symptoms. All of this will be posted to the database and stored.

Most of the true calculations occur in the backend, however we do calculate statistics for individuals in the front end. We would like to move this to the backend, but unless the database got quite large there would be minimal impact. Other than that, we use POST and GET requests to send json objects to the database. Along with the UI design and activities the only thing not mentioned is the use of a unique identifier for the user who is currently logged in. This is received when credentials are deemed to be accurate so that we can track a user and update their information throughout the duration of their use.

Used for the backend. The controllers run all the java classes we have for our project. All the work like, getting JSON data from the Rest APIs, or getting or setting data from our MySQL databases. Apart from getting JSON data, using that data to analyze the Covid conditions in different states, like the total number of recorded positive and negative cases, recovered cases, or deaths. And sending to the frontend. Checking, if the states are under lockdown or what zone they’re in (Green, Orange, Red). Apart from the data around the world, a User’s personal information is also managed here. When a user enters his information, that data is managed and stored in the databases. If a user enters his symptoms or his Covid status apart from keeping track of his symptoms, a quarantine timer keeps track of how many days of quarantine are left for him. These symptoms can also be updated by the user, and that information is updated in the database too. The age of the user is used to put them in one of the 4 age groups, because different age groups have varying effects of CoronaVirus. And the basic work like, the login credentials and new users sign up information is also handled here. The login ID and password are matched from the database when a user tries to login. And new information is stored in the database when a new user registers.

**Table Relationships Diagram**

****

**MySQL**

# User Entity has a One to One unidirectional relationship with the Symptoms, Covid and Details Entity.

1. Details Entity relates to the age group and gender of the user
2. Covid Entity relates to the covid test results and dates of the user
3. Symptoms Entity relates to the symptoms a user may have.