**Executable Release: Database**

**Goal:**

Create an SQL database that can store and track all information relating to appointment scheduling, vaccination stocks, and user information

**Classes to be implemented:**

DatabaseConnection

**Previously implemented classes to use:**

none

**Use case to be implemented:**

Store user & campus data

Retrieve user & campus data

**Inputs:**

User Data(name, email, password, insurance)

Appointment Data (id, campus, date, time, brand)

Campus Data (isRegional, vaccinecount, revenue, brand, number to order, delivery date, orderPlaced )

**Outputs:**

User Data

Appointment Data

Campus Data

**Time for completion:**

To be completed by: 04/22/2021

**Executable Release: Save Campus Data**

**Goal:**

Store campus data in database

**Classes to be implemented:**

Campus, databaseConnection

**Previously implemented classes to use:**

Patient, Appointment, Vaccine, Insurance

**Use case to be implemented:**

Save values for the campus such as revenue, vaccines given, vaccine brand,etc

**Inputs:**

Values for revenue, vaccines given and on hand at campus, brand on hand, order info

**Outputs:**

none

**Time for completion:**

To be completed by: 04/27/2021

**Executable Release: Log In**

**Goal:**

Verify email/password string given by the user against values in database

**Classes to be implemented:**

LoginMenu, user

**Previously implemented classes to use:**

none

**Use case to be implemented:**

Login to system

**Inputs:**

Username

password

**Outputs:**

Verification from values in database

**Time for completion:**

To be completed by: 04/27/2021

**Executable Release: Schedule an Appointment**

**Goal:**

Create an appointment and save it to the database

**Classes to be implemented:**

Appointment, patient, appointmentMenu

**Previously implemented classes to use:**

databaseConnection, campus

**Use case to be implemented:**

Create appointment

**Inputs:**

Date, time, insurance, name

**Outputs:**

All inputs to database

**Time for completion:**

To be completed by: 04/27/2021

**Executable Release: Load Appointments from Database**

**Goal:**

Retrieve appointments from database

**Classes to be implemented:**

Campus, appointment

**Previously implemented classes to use:**

patient

**Use case to be implemented:**

Retrieve appointment data

**Inputs:**

none

**Outputs:**

Appointment data from database

**Time for completion:**

To be completed by: 04/27/2021

**Executable Release: Keep Log of Vaccines Given at Each Campus**

**Goal:**

Increment number of vaccines given at each campus, decrement vaccines on hand at each campus

**Classes to be implemented:**

campus

**Previously implemented classes to use:**

databaseConnection, appointment

**Use case to be implemented:**

Keep track of vaccines given at each campus and vaccines on hand

**Inputs:**

Appointment info in sql query

**Outputs:**

Save vaccines given and on hand to database

**Time for completion:**

To be completed by: 04/27/2021

**Executable Release: Cancel an Appointment**

**Goal:**

Allows a user to cancel an appointment

**Classes to be implemented:**

Appointment, viewapptsmenu

**Previously implemented classes to use:**

Patientmenu

**Use case to be implemented:**

Allow user to cancel an appointment

**Inputs:**

An appointment to be canceled

**Outputs:**

Removal of appointment from database

**Time for completion:**

To be completed by: 04/27/2021

**Executable Release: Load Campus Data from Database**

**Goal:**

Load all campus into class from database

**Classes to be implemented:**

campus

**Previously implemented classes to use:**

database

**Use case to be implemented:**

Retrieve data from database

**Inputs:**

**Sql query**

**Outputs:**

Data into class

**Time for completion:**

To be completed by: 04/27/2021

**Executable Release: Check Against Existing Appointments**

**Goal:**

Retrieve appointments from database and check current appointment against it

**Classes to be implemented:**

appointment

**Previously implemented classes to use:**

databaseConnection, campus

**Use case to be implemented:**

Keep track of existing appointments

**Inputs:**

Current appointment data

**Outputs:**

Boolean of match

**Time for completion:**

To be completed by: 04/27/2021

**Executable Release: Patient Menu Implementation/Structure**

**Goal:**

Create a menu to get patient input for different use scenarios

**Classes to be implemented:**

patientMenu

**Previously implemented classes to use:**

Appointment, patient

**Use case to be implemented:**

Select options

**Inputs:**

User selection of an option

**Outputs:**

Gui menu corresponding to user selection

**Time for completion:**

To be completed by: 04/29/2021

**Executable Release: Alert When New Shipment**

**Goal:**

Send an email alert when a new shipment arrives at a campus

**Classes to be implemented:**

alerts

**Previously implemented classes to use:**

Campus, databaseconnection

**Use case to be implemented:**

Send flash alert when campus has vaccines available

**Inputs:**

List of emails from database

Current date

Campus to send alert for

**Outputs:**

An email to every user in the database

**Time for completion:**

To be completed by: 04/29/2021

**Executable Release: Reschedule an Appointment**

**Goal:**

Allows the user to reschedule an appointment

**Classes to be implemented:**

Appointment, viewApptsMenu

**Previously implemented classes to use:**

Appointment, patient

**Use case to be implemented:**

Cancel an appointment

Schedule an appointment

**Inputs:**

Appointment to be rescheduled

New appointment Data

**Outputs:**

Removal of appointment from database and schedule a new one in the database

**Time for completion:**

To be completed by: 04/30/2021

**Executable Release: Calculate and display Revenue**

**Goal:**

Calculates revenue generated by each individual campus

**Classes to be implemented:**

campus

**Previously implemented classes to use:**

database

**Use case to be implemented:**

Display revenue earned at each campus

**Inputs:**

Patients with appointment at campus and their insurance values

**Outputs:**

Amount of money generated by each campus in form of bar graph

**Time for completion:**

To be completed by: 05/01/2021

**Executable Release: Order Vaccines**

**Goal:**

Order vaccines when vaccines on hand drops below 50 for regional campuses or 150 for the Kent campus

**Classes to be implemented:**

campus

**Previously implemented classes to use:**

database

**Use case to be implemented:**

Order vaccines when low on vaccines

**Inputs:**

Number of vaccines to order

**Outputs:**

A date for the order to be completed in the database, a boolean for an order being placed, and the number of vaccines

**Time for completion:**

To be completed by: 05/02/2021

**Executable Release: Visualize Data**

**Goal:**

Displays a bar graph for number of users vaccinated at each campus and another to display revenue earned at each campus

**Classes to be implemented:**

visualizationMenu

**Previously implemented classes to use:**

Database, campus

**Use case to be implemented:**

Visualize number of vaccinations at each campus

**Inputs:**

Number of users vaccinated at each campus

revenue from each campus

**Outputs:**

A bar graph of revenue from each campus

A bar graph of users vaccinated at each campus

**Time for completion:**

To be completed by: 05/03/2021