Executable Release: login/ register

Goal:

Verification and successful use of the registration and login system.

Classes to be implemented:

login, registration, user, db.

Use case to be implemented:

Users enter their username and password and the system allows access if they match.

Users can also create an account if one they don't have one yet.

Inputs:

Usernames and passwords

Outputs:

Outputs if login is successful or if it failed.

Time for completion:

April 2nd

Executable Release: create appointment

Goal:

Verification and successful use of the system to make appointments.

Classes to be implemented:

appointment, *campus*

Previously implemented classes to use:

db

Use case to be implemented:

Users create appointments.

Inputs:

dummy appointments.

Outputs:

Tells the user if their appointment was created or not

Time for completion:

April 16th

Executable Release: Order

Goal:

Verification and successful use of the system to order new vaccines when a campus is running low.

Classes to be implemented:

Vaccine, orderRequest

Previously implemented classes to use:

db, campus

Use case to be implemented:

Employees can order more vaccines when supplies are low.

Inputs:

A request to order.

Outputs:

Confirmation that or rejection of the order request

Time for completion:

April 23rd

Executable Release: Dashboard

Goal:

Display vaccine counts per campus to the user and give the mission statement.

Classes to be implemented:

Graphs

Previously implemented classes to use:

db, campus

Use case to be implemented:

Registered users can see completed appointment graphs

Inputs:

Input a campus

Outputs:

Graph showing how many vaccinations happened at a campus

Time for completion:

April 29th

Executable Release: Reschedule appointment

Goal:

Verification and successful use of the system to reschedule appointments.

Classes to be implemented:

appointment, *campus*

Previously implemented classes to use:

db,

Use case to be implemented:

Users reschedule an appointment.

Inputs:

dummy rescheduled appointments.

Outputs:

Informs the user of a rescheduled appointment

Time for completion:

April 29th

Executable Release: Cancel appointment

Goal:

Cancel already confirmed appointments

Classes to be implemented:

appointment, *campus*

Previously implemented classes to use:

db,

Use case to be implemented:

Users cancel an appointment.

Inputs:

dummy cancelled appointments.

Outputs:

Informs the user their appointment was cancelled

Time for completion:

April 29th

Executable Release: Availability

Goal:

Cancel already confirmed appointments

Classes to be implemented:

availability

Previously implemented classes to use:

Users, db, appointment

Use case to be implemented:

Users can view a daily schedule of our system

Inputs:

A day and month

Outputs:

A table listing out each campus name followed by appointments at specific times

Time for completion:

April 29th

Executable Release: Admin

Goal:

Show our systems working properly, showing usage of insurance and vaccine counts

Classes to be implemented:

admin

Previously implemented classes to use:

Db, users, campus,

Use case to be implemented:

Admins can view relevant data to the campuses

Inputs:

An admin user can input values on specific areas

Outputs:

Outputs range from what the user imputed… this ranges from dose counts per campus, total completions and revenue for a campus.

Time for completion:

April 29th

Executable Release: User Details

Goal:

Show the user their details and current appointments

Classes to be implemented:

user_details

Previously implemented classes to use:

Db, users

Use case to be implemented:

Users can view details related to themself.

Inputs:

Nothing

Outputs:

Shows the user relative data relating to themself.

Time for completion:

April 29th