

Facility Finder

Significance of Our Project:

A hospice is an organization which is responsible for finding appropriate facilities for individuals receiving palliative end-of-life. Patients will often spend the rest of their lives in a facility.

A facility is wherever the patient is going to spend their life, which is any place that will have the resources they need.

Right now, there is a contract system where hospices will get contracts signed with facilities which establishes pathways for hospice patients to be transferred to facilities they have contracts with based on availability. Contract really just sets up a relationship between hospice and facility, lets them put patients into specific facilities. But even with contracts, it's a messy system that is inefficient and slow. Hospices will need to call facilities whenever you need to replace a patient and call hundreds of facilities because they are often full. People have a hard time figuring out where patients can go. Thus, this database exists to allow administrators to reserve rooms in facilities in order to make the reserving process much more efficient.

Instructions for running our project:

- Run the createTables.sql file to create Users, Facilities, Rooms, and Patients tables in MySQL
- Run the insertData.sql file to insert data into the three tables
- The maven project is provided in the project.zip folder
- Put application.properties and the jar file onto your Desktop
- Open Command Prompt/Terminal and go to your Desktop, then open the jar file to turn on the server
- The landing page is localhost/register.html

Using the Website

- Users who have used the site before are stored in the Users table and should skip the register and go to the login page
- New users must register before logging in
- Once logged in, the user can search a facility by typing a facility name in the searchbar
- Empty rooms in the specific facility should be visible in a new table on the page
- To reserve, user enters a patient id (a large number), patient name, illness, additional notes, the facility id, and the room number (of one of the empty rooms they see in the table)

- The page automatically reloads, and the reservation should be reflected: The room is not shown anymore when searching for it because it is not empty, and the number of available rooms in the facility decreases by 1
- This patient's information on their booking has been added to the Patients table. User can view patients by clicking on button "Access Patients Database"
- User can cancel a reservation by typing in facility id and room number in the provided form.
- On reloading page, this cancellation should be reflected: The room is shown when when searching for it because it empty again, and the number of available rooms in the facility increases by 1
- Note: If user is not logged in, then data on patient page or database page will not load. (Due to token validation). If user logs in successfully, then the data on these pages will load

CALL COMPLETION CHECKLIST: (copied from our original proposal)

API Calls to be created on Spring Server:

/registerUser - Creates a user with username and password and stores information in database

Created a web page that asks for a username and password. The username and password is then stored into MySQL database, in the Users table. There is also a button that allows users to log in if they already have login credentials.

/login - Verifies user login credentials and return a token for the client to use other API calls

Web page asks for users to input their username and password. If the credentials are correct, then they receive a token and are directed to the homepage that has the facilities database. If the credentials are not correct, then they receive an error message. Those who have not logged in are unable to see any of the database information even if they go to the link.

/facilitySearch - Searches for facilities in database which match entered facility name

In the home page, there is a search bar for the user to put in a facility name in order to look at their information such as number of rooms available, their phone number, etc. The facility name must fully be written, either in lower or uppercase.

/reserve - User reserves a room, which makes the room not available anymore in the database

In order to reserve a room, users must input some information such as the patient's name, their illness, the desired room and facility, etc. Once the information is put in, then the facility, room, and patient database are all updated. The facility database decrements one room from the rooms available column for that particular facility, the room database changes the availability of

that particular room, and that patient is inserted into the patient database. If a user tries to reserve a room that is already reserved, then an error message will pop up.

/cancel - User cancels a reservation, makes the room available in the database

In order to cancel a room, users must input the facility ID and room number they wish to cancel. The facility database increments one for the rooms available column for that particular facility, and that room changes its availability status to available. If a user tries to cancel an already available room, there is an error.

HTML for the front-end of the website

We used HTML.

MySQL database stores hospice, room, patient, and booking information.

We also used MySQL.

BONUS: WE DID PATIENTS DATABASE TOO! (this was not in our proposal)

So, we decided to do some extra work so that users will also be able to see all of the patients that have reserved rooms and their information. Woot!