Muhammad Daniyal Mateen

HotelKey | 2022

Hospital Appointment System

**Project Workflow**

**Project Introduction**

*Hospital Appointment System* is an AWS based appointment system which is using SQS for its internal operations.

Hospital Appointment System contains three doctors with different Doctor Codes assigned to them and each doctor in the system has its separate queue of patients waiting for their turns for their checkups.

**Doctor**

Each doctor is able is to login to their account, which is protected with One-Way Hashed Password, and can see the queue of patients waiting for him. The status (FREE/BUSY) of doctor is also shown there. “FREE” represents that there is no patient in his clinic whereas “BUSY” represents that someone’s checkup is in progress.

*Next Patient Button* simulates that the Checkup of current patient has ended (if any) and new patient is now called in the clinic for checkup. Also, an email is sent to the patient as a reminder informing him about his turn.

Reload button just simply reloads the list of patients waiting for their checkups and updates the list if any patient is added.

Main Menu button simply takes the control to the main login area.

**Patient**

Each Patient can see its data using its portal by entering its patient Id. He can see it name, reason, estimated waiting time, estimated checkup time.

Next to his record, there is a main menu button that can take the control to the index page.

**Workflow**

* First the Patient Details are added by the receptionist
* The Patient details are converted into Gson() of details.class
* Then the details object are passed to the Receptionist Class
* From that details object, a patient and appointment object is created after generating ids
* That patient and appointment objects are added in the Database by Receptionist.
* The appointment is added in the Queue by the Receptionist
* This stores the data in Database and the Queue

**Doctor Servlet**

* The doctor logs in to the portal using its Doc Code and password which is One-Way Hashed.
* The Doctor Code is set in Attribute and the request is forwarded to doctorList Servlet.
* The docCode variable is initialized by request parameters.
* Then using doctor code we get the Status of Doctor (representing that if the doctor is Busy or Free) which is later printed using servlet.
* Later, using the same docCode we print all the messages present in the Queue.
* Below, there is a buzzer button that reads the top most appointment from the queue based on following conditions:
  + If the doctor is FREE and there are appointments in the queue
    - Then the status of the doctor is changed to the receipt handle of the top most message (which will be further used to delete the message) and then the message is read from the queue, making it go in the visibility timeout. And a mail is sent to the patient as a reminder.
  + If the doctor is BUSY and there are appointments in the queue
    - The status of that doctor in the DB is replaced by the receipt handle of the top most appointment placed in the queue. The mail is also sent as the reminder
  + Else the doctor status is set to free
* Then the control is forwarded to doctorList servlet.
* Below there is main menu button that takes you back to the index page.

**Patient Servlet**

* The patient logs in to the portal using its provided Patient ID.
* The request is then forwarded to the ShowPatientDetails Servlet after setting the patient ID in the request
* Using that patient ID the details of the patient are displayed on the screen using ShowPatientDetails Servlet along with estimated time and Estimated checkup time.