

**Course Title:** Data Structure and Algorithm

**Course Code:** DSA-241 **Section:** D

**Submitted to:** Athar Ikhlaq

**Project -HealthHub**

**Team Members:**

Shareen Faisal **(Team Leader)** 221400013 (4.0)

Eman Faisal 221400020 (4.0)

Dua zaidi 221400026 (3.09)

Ayesha Saeed 221400054 (2.5)

Sajal saleem 221400017 (2.4)

**DSA-Project**

**Summary**

Many customers and doctors need online healthcare services, but they face various challenges, such as lack of access, high cost, poor quality, and low satisfaction.

HealthHub system aims to provide a way to access and deliver healthcare services online, by allowing customers to order medicines (searching, selecting, adding to cart, and paying), book lab tests (view categories, search), book online doctor consultation, and manage their health records, and allowing doctors to register and view their booked appointments and payments. Customer will be able to manage their blood pressure records and the system will generate a warning if the blood pressure is outside the normal range. Both customers and doctors will be able to log in and register.

**Project Scope:**

**Admin:**

He controls Medicines, Lab Tests and Doctor Consultation.

He can view, add, and remove medicines.

He can view, add, remove lab tests.

He can view, add, and remove doctors and view their appointments.

**Doctor Consultation:**

Customer can view doctors in various categories. He can filter and select a doctor to for an appointment/consultation. He views the schedule of the doctor and selects the date and time for the online appointment. He pays for the booked appointment.

The doctor can log in. The doctor can view his appointments for the day.

**Lab Tests:**

Customer can view different categories of lab tests. He can search any lab tests. He then selects a lab test and selects date and time for its home sampling. He adds the lab tests to cart and can view the cart. He proceeds to checkout to pay for the lab test. The price of the lab test is deducted from the customer’s account.

**Blood Pressure Records:**

Customers can add, update, remove, view high warning records, and analyze their blood pressure records. The customers can set their advised systolic and diastolic blood pressure readings and can also edit them. The system will check for a warning each time a record is added or updated. The warnings will be according to the set advised readings, if not set no warnings will be displayed. The system will generate 3 types of warnings:

* If both systolic and diastolic readings are within ±10 of the advised ranges, it's considered a "LOW-LEVEL WARNING".
* If both systolic and diastolic readings are within ±20 of the advised ranges but not within the ±10 range, it's considered a "MEDIUM LEVEL WARNING".
* If the readings are outside the ±20 range of the advised ranges, it's considered a "HIGH-LEVEL WARNING".

The customer can also get average systolic and average diastolic readings for a specific date.

The record of each customer is stored in a file named the ID of the customer.

**Medicine Ordering:**

The customer can order medicines by selecting them from displayed categories or by searching them. The customer can add them to cart and pay for them using their account. The customer can add balance to their account or remove items from cart if their balance is insufficient. The customer can also cancel order.

**Functional Requirements:**

1. **Admin:**

**Medicines:**

* The admin shall be able to add a medicine if it does not already exist.
* The admin shall be able to remove a medicine.
* The admin shall be able to view all medicine.
* The admin shall be able to analyze medicines (view the most frequently sold medicines).

**Lab Tests:**

* The admin shall be able to add a lab test if it does not already exist.
* The admin shall be able to remove a lab test.
* The admin shall be able to view all lab tests.

**Doctor Consultation:**

* The admin shall be able to add a doctor if it does not already exist.
* The admin shall be able to remove a doctor.
* The admin shall be able to view all doctors.
* The admin shall be able to view the appointments of every doctor.

1. **Login:**

**Doctor:**

* The doctor is able to login by entering name, id, password.
* The system shall be able to validate the id and password.

**Customer:**

* The customer is able log in with username and password.
* The system shall be able to validate the id and password.

1. **Sign In(Customer):**

The customer is able to sign in with a username and password, phone number, address, gender, account Number, and balance.

The system sets these attributes and checks if this user name exists before.

1. **Ordering Medicines:**

* The system allows the customer to either view categories or search a medicine at a time.
* The system allows customer to view categories of medicines, such as Eye, Nose, Ear Medicines, Gastrointestinal Tract Medicines, Cardiovascular System Medicines, and Central Nervous System Medicines.
* Customer can choose a category, and the system displays medicines belonging to that category.
* Customer can search for medicines based on partial or full names.
* The system provides search results, including names and prices of matching medicines.
* The system checks whether a medicine with a given name already exists before adding it. If it exists, the system informs the customer.
* After displaying search results, the system prompts users to either exit or search again.
* Customer can choose to exit the program or perform another search.
* The system shall be able to remove a customer choosen item from the cart if he has insufficient balance to pay.
* The system displays the updated cart after the removal of the item/labtest.
* The system shall be able to store orders.
* The system shall be able to calculate the total bill of the lab tests booked.
* The system shall be able to generate the bill of the lab tests ordered.
* The system shall be able to apply discount of 10% on the bill if it is above Rs. 5000.
* The system shall be able to apply tax of 5% on the total bill.

**Data Structures Implemented:**

* Doubly Linked List

1. **Doctor Consultation:**

* The system shall be able to store doctors. (An ArrayList is implemented. There is further an ArrayList on each index of the ArrayList).
* The customer shall be able to view various categories.
* The customer shall be able to view various doctors in a single category.
* The customer shall able to filter the doctors in one category on the basis of ratings.
* The customer shall be able to select a doctor and view his/her appointment schedule
* The customer shall be able to enter the date for the appointment.
* The customer shall be able to enter the time for the appointment.
* The customer shall be able to add balance in his account if he has less balance than the doctor’s consultation fee.
* The customer shall be able to pay for the appointment.
* The system shall be able to display the appointment summary.
* The system shall be able to store the appointments. (in priority Queue)
* The system shall be able to remove appointment.

**Data Structures implemented:**

* ArrayList with further an ArrayList on each index. (the first arraylist contains an arraylist storing doctors of a specific category).
* PriorityQueue (for storing appointments)
* 2-D Array (for schedule of appointmnets of a doctor)

1. **Lab Test Booking:**

* The system shall allow the addition of new lab tests alphabetically with details such as name, sample requirements, price, turnaround time, sample type, category, and quantity.
* The customer shall be able to view categories of lab tests.
* The customer shall be able to search a specific lab test based on name.
* The customer shall be able to select a specific lab test and add it to cart.
* The customer shall be able to enter date for the lab test booking.
* The customer shall be able to enter time for the lab test booking.
* The customer shall be able to view the cart details.
* The system shall be able to remove a customer chosen item from the cart he he has insufficient balance to pay.
* The system displays the updated cart after the removal of the item/labtest.
* The system shall be able to store orders.
* The customer shall be able to pay for the lab test.
* The customer shall be able to add balance in his account if he has less balance than the lab test price.
* The customer shall be able to book a lab test.
* The customer shall be able to cancel his order.
* The system shall be able to calculate the total bill of the lab tests booked.
* The system shall be able to generate the bill of the lab tests ordered.
* The system shall be able to apply discount of 10% on the bill if it is above Rs. 5000.
* The system shall be able to apply tax of 5% on the total bill.

**Data Structure Implemented:**

* Sorted ArrayList
* Algorithm : Binary Search

1. **Health Records (Blood Pressure Only):**

* The customer shall be able to enter the doctor-advised systolic and diastolic reading.
* The customer shall be able to add a new record containing systolic and diastolic readings.
* The customer shall be able to remove a reading.
* The customer shall be able to update a record by changing systolic and diastolic values.
* The customer shall be able to analyze his/her records. (View the average systolic and diastolic readings for a given date)
* The system shall be able to generate a warning if the systolic and diastolic reading deviates more from the doctor-advised reading.
* The customer shall be able to view all the BP records having a high warning.

**Data Structures Implemented:**

* Files
* Algorithm: Merge Sort and Binary Search
* Arraylist

1. **Health plan Suggestions:**

* The system shall be able to suggest health plans based on category of the medicines ordered by the customer.

**Data Structure Implemented:**

* Binary Tree