

Health Services



MyHealth_Services

Aaresh Bachana

Anish Churi

Harshil Sheth



Project Summary

- Java based stand-alone application which can be used in hospitals and clinics
- Application can be used by:
 1. Doctor
 2. Patient
 3. Clerk

Project Summary

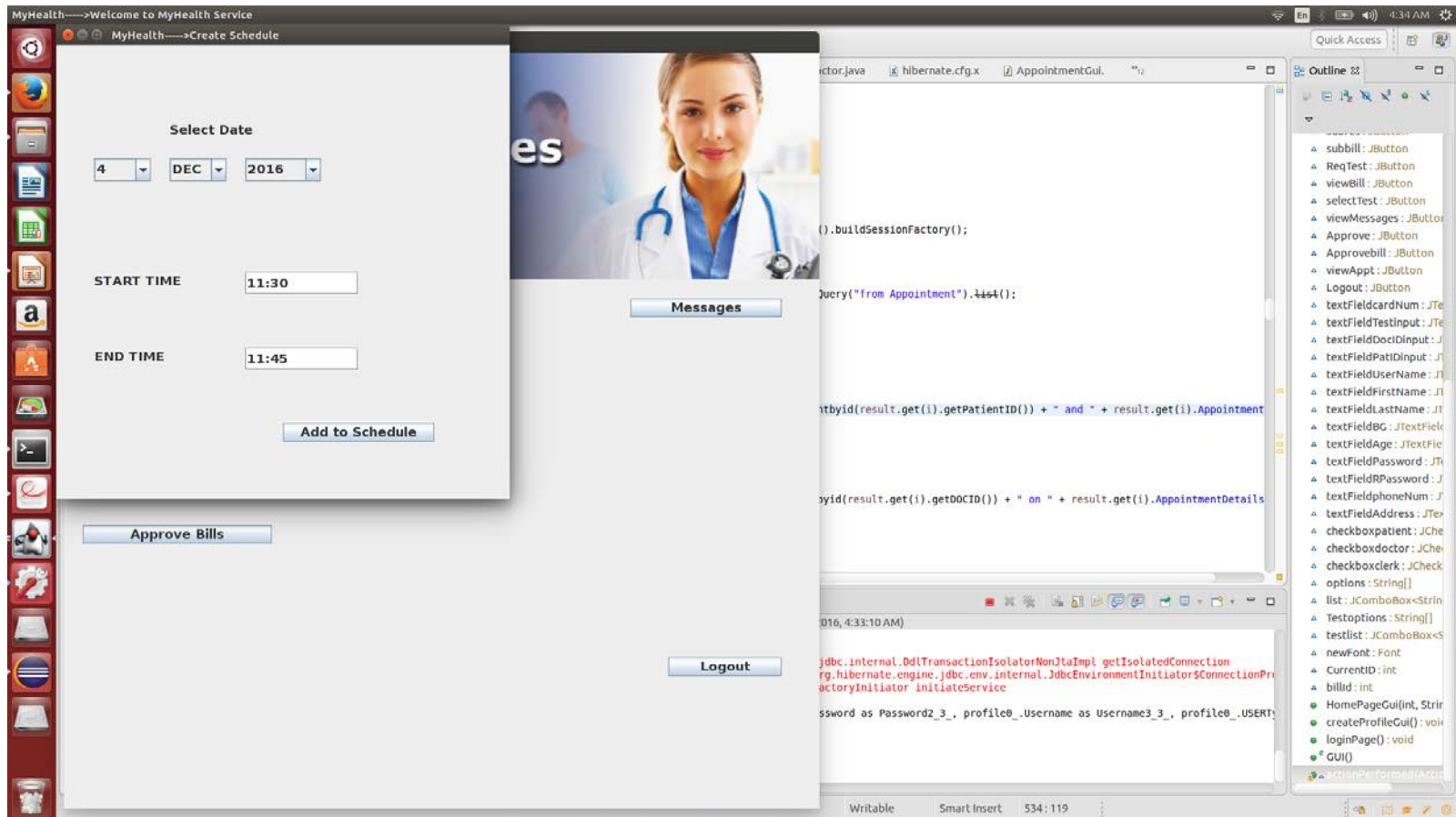
- Application can be used for
 1. Creating and Scheduling Appointments
 2. Request medical tests and getting notifications on completion of result computation
 3. Viewing and paying medical bills off when added to the patient's profile

Use Cases Implemented

- User can login or create profile
- Doctor creates schedule
- Patient searches for doctor by name
- Patient can book an appointment
- Patient can request medical test
- Doctor can approve medical test for the respective patient
- Clerk notifies patient and doctor when results computed
- Clerk can upload bills to patient profile
- Patient can pay bills using authorization via database

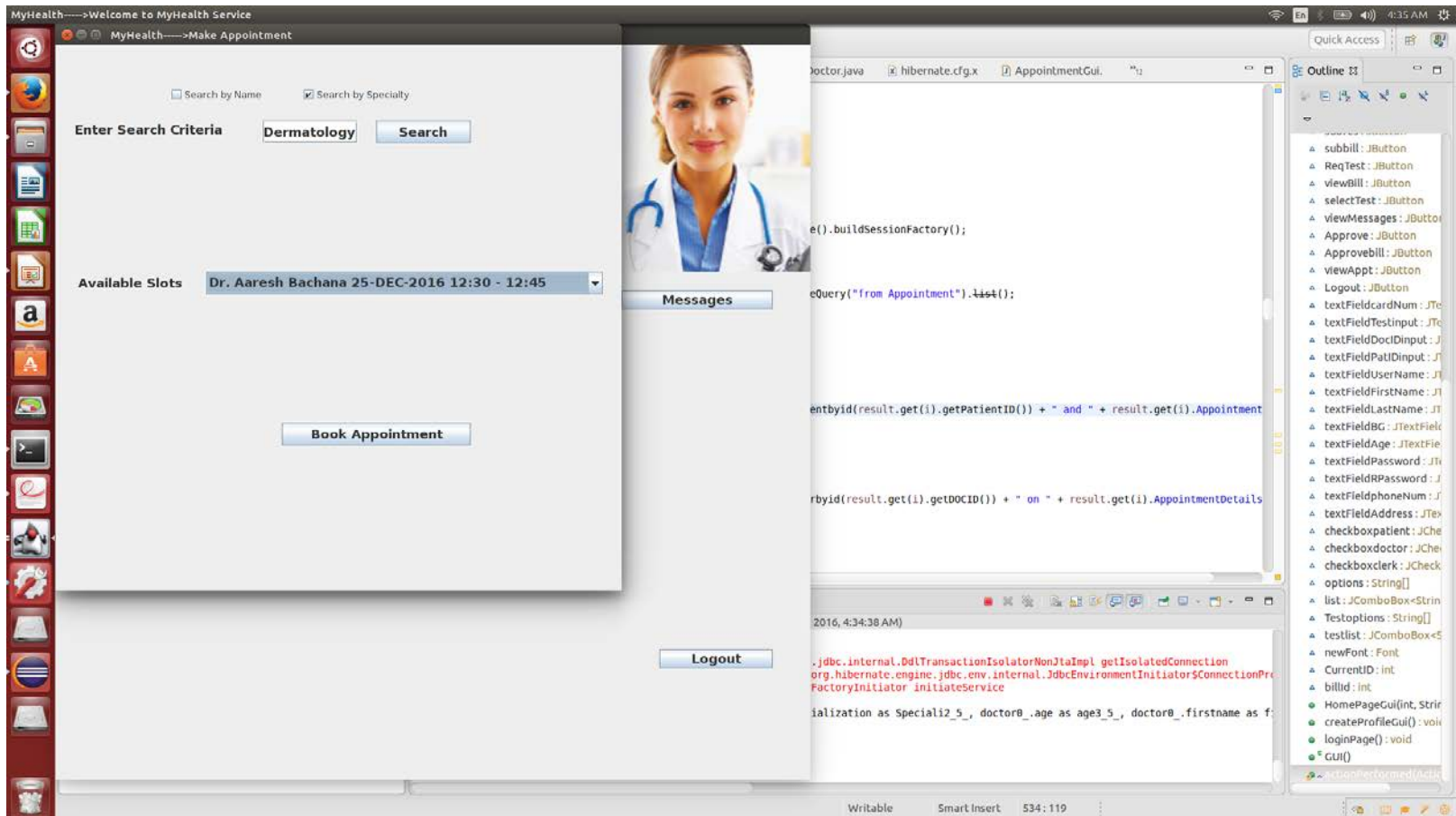
Use Cases

- Doctor can create schedule



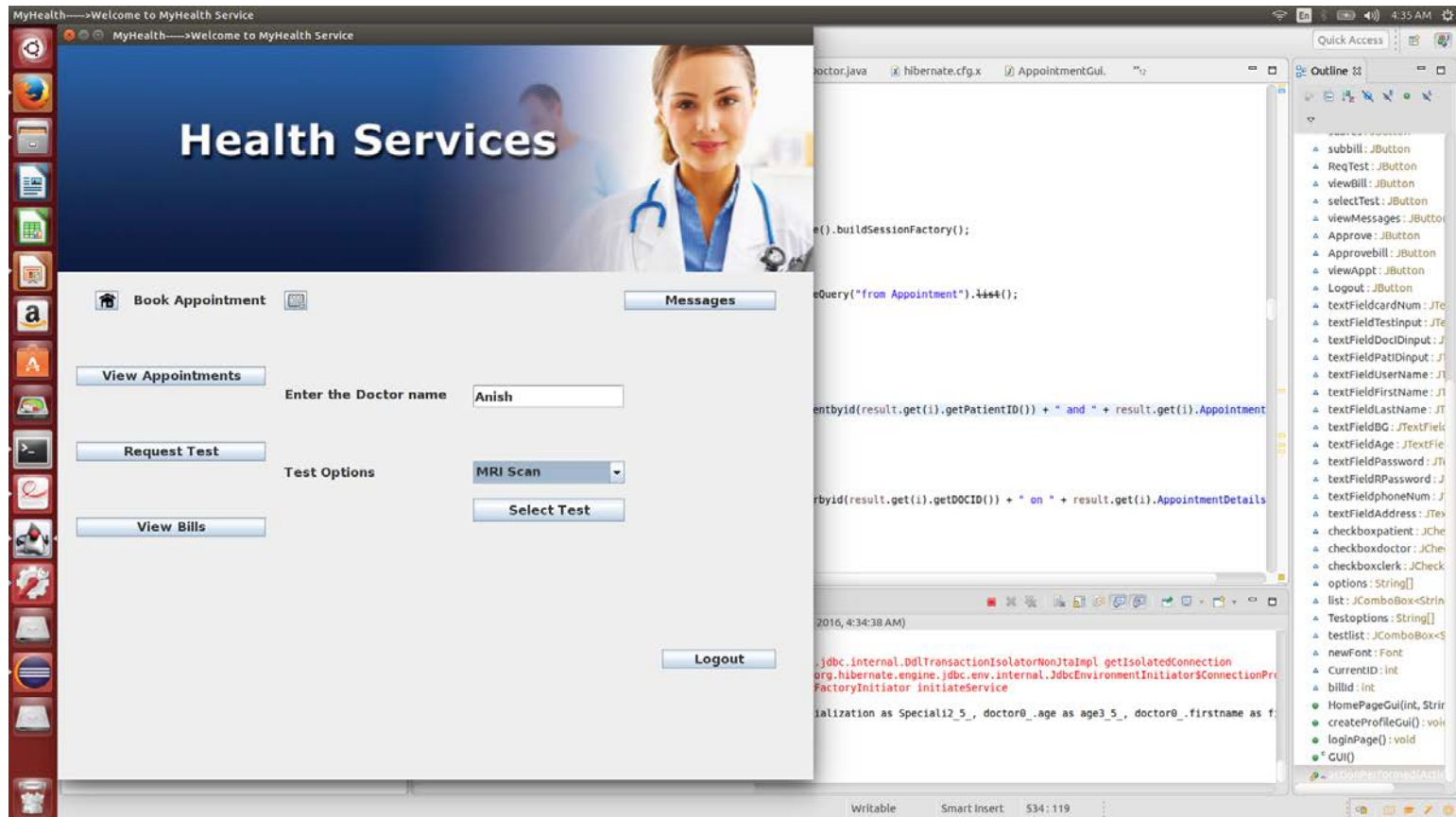
Use Cases

- Patient can search for a certain doctor



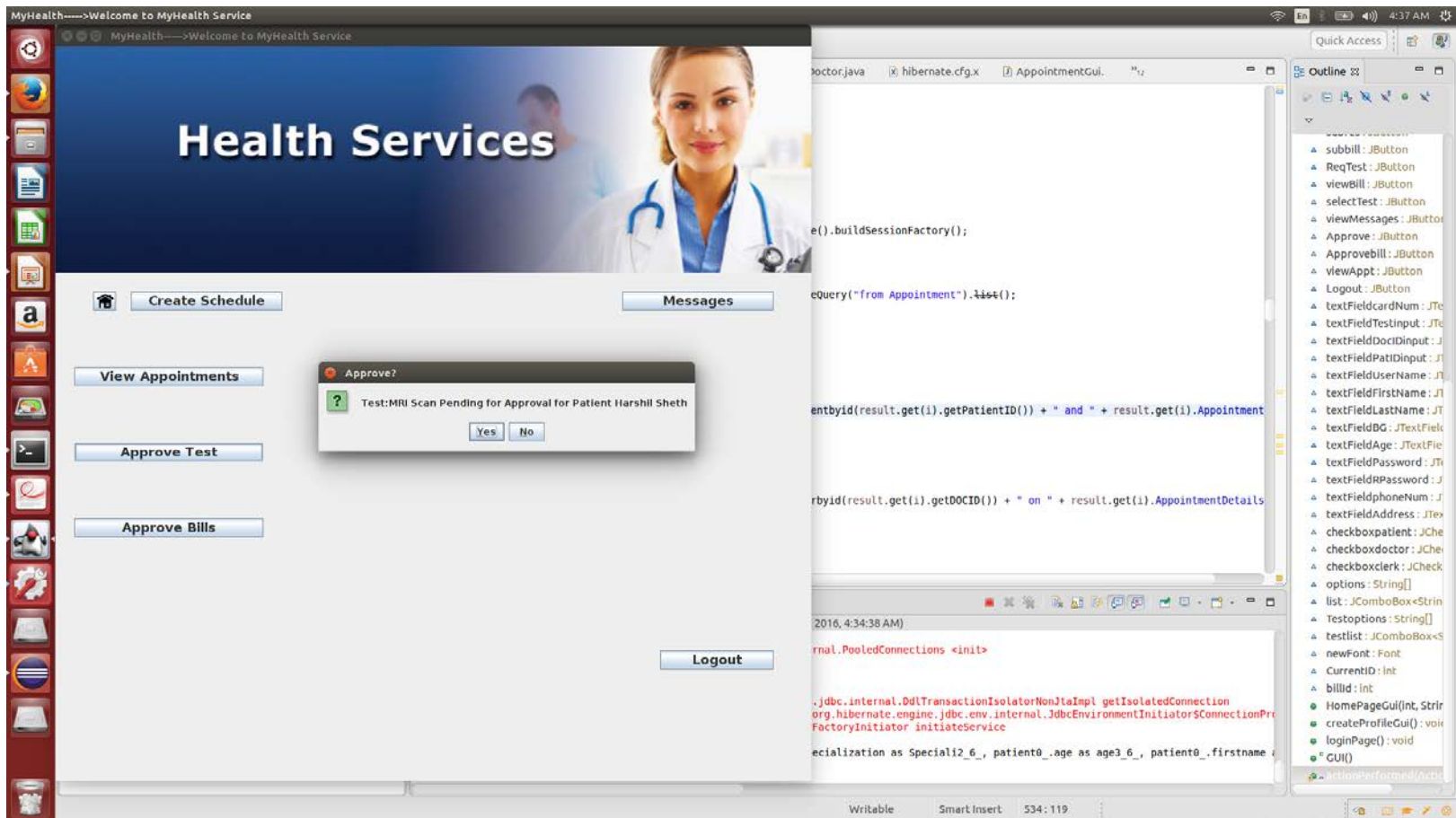
Use Cases

- Patient can request medical test



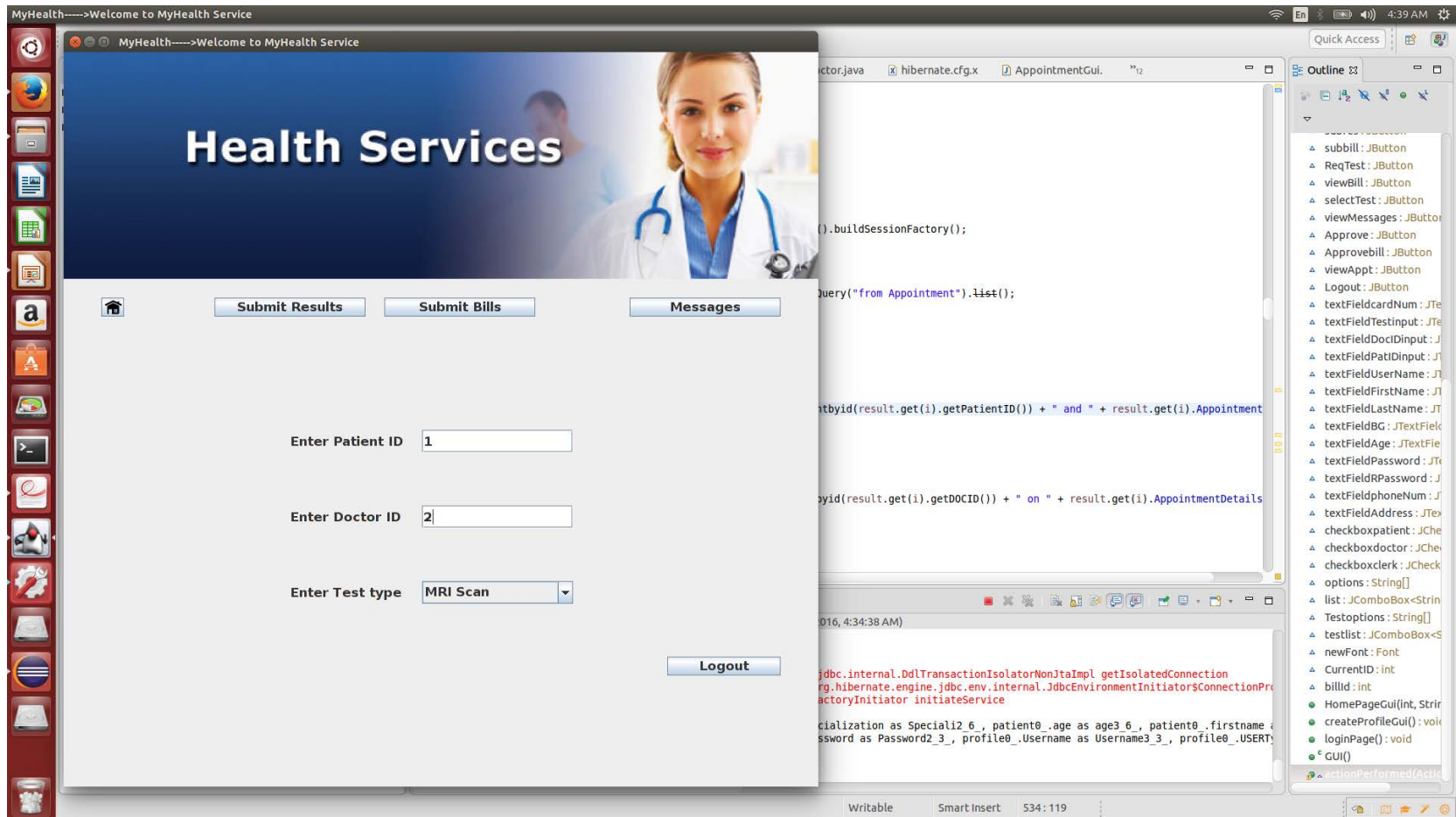
Use Cases

- Doctor can approve medical test



Use Cases

- Clerk can upload bills



Use of Hibernate

- Used MySQL database for storage of persistent data
- Used Hibernate for mapping classes to the database

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
  <session-factory>
    <!-- Database connection settings -->
    <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="connection.url">jdbc:mysql://localhost:3306/MyHealth</property>
    <property name="connection.username">root</property>
    <property name="connection.password">Kagebushin123$</property>

    <!-- JDBC connection pool (use the built-in) -->
    <property name="connection.pool_size">10</property>

    <!-- SQL dialect -->
    <property name="dialect">org.hibernate.dialect.MySQLDialect</property>

    <!-- Disable the second-level cache -->
    <property name="cache.provider_class">org.hibernate.cache.NoCacheProvider</property>

    <!-- Echo all executed SQL to stdout -->
    <property name="show_sql">true</property>

    <!-- Drop the existing tables and create new one -->
    <property name="hbm2ddl.auto">update</property>

    <!-- Mention here all the model classes along with their package name -->
    <mapping class="Patient"/>
    <mapping class="Doctor"/>
    <mapping class="Clerk"/>
    <mapping class="Profile"/>
    <mapping class="Schedule"/>
    <mapping class="Slots"/>
    <mapping class="Appointment"/>
    <mapping class="Records"/>
    <mapping class="Bills"/>
    <mapping class="CreditCards"/>

  </session-factory>
</hibernate-configuration>
```

Hibernate Annotations

- @Entity- Used for the class to be persisted.
- @Id- Used to generate a unique id depending on the Strategy used in @GeneratedValue.
- @GeneratedValue- Used to determine the strategy for the Id.(eg- GenerationType.IDENTITY)
- @Column- Used to create different attribute columns in the database tables.
- @MappedSuperclass- Used to not persist the abstract superclass.

MySQL Database

```
anish@anish-HP-ENVY-x360-Convertible: ~
mysql>
mysql>
mysql> select *from LoginInfo;
+-----+-----+-----+-----+
| ProfileID | Password | Username | USERType |
+-----+-----+-----+-----+
| 1 | csc15448 | anich123 | Doctor |
| 2 | csc15448 | aarbac123 | Doctor |
| 3 | csc15448 | harsh123 | Patient |
| 4 | csc15448 | abma123 | Clerk |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select *from MyHealth_Patient;
+-----+-----+-----+-----+-----+-----+-----+-----+
| PersonID | Specialization | age | firstname | gender | lastname | location | phoneNo | securityAnswer |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | NULL | 23 | Harshil | Male | Sheth | Glenwood Drive | 3368742667234 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select *from MyHealth_Doctor;
+-----+-----+-----+-----+-----+-----+-----+-----+
| PersonID | Specialization | age | firstname | gender | lastname | location | phoneNo | securityAnswer |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Cardiology | 24 | Anish | Male | Churi | Folsom Street | 87234878347 | NULL |
| 2 | Dermatology | 23 | Aaresh | Male | Bachana | Marine Street | 723854825432 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select *from MyHealth_Clerk;
+-----+-----+-----+-----+-----+-----+-----+-----+
| PersonID | Specialization | age | firstname | gender | lastname | location | phoneNo | securityAnswer |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | NULL | 26 | Abhilash | Male | Manjunath | Arapahoe Village | 823767348482 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

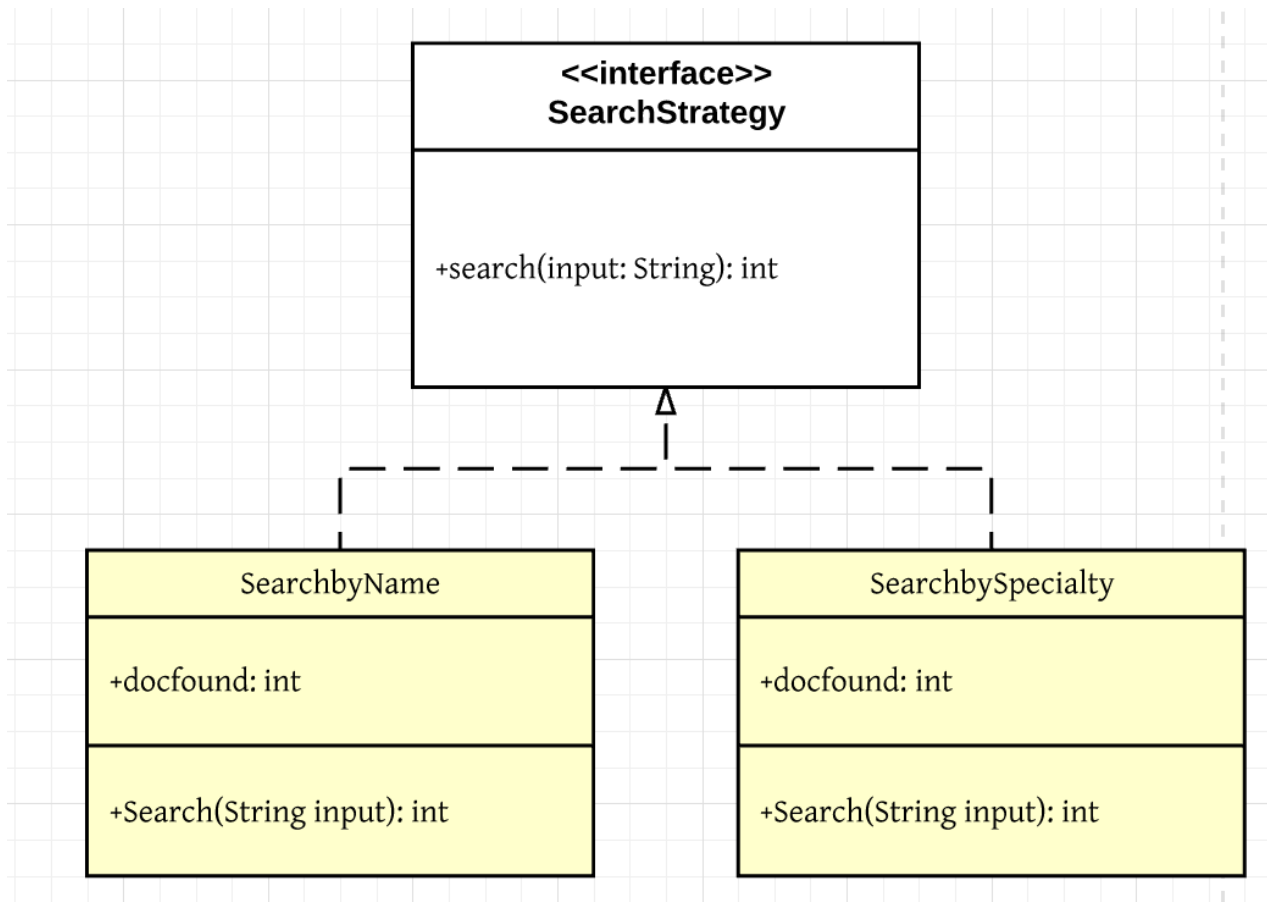
mysql> select *from AppointmentInfo;
+-----+-----+-----+-----+-----+-----+
| appointmentID | AppointmentDetails | DOCID | patientID | slotnum |
+-----+-----+-----+-----+-----+-----+
| 1 | Dr. Anish Churi 7-DEC-2016 11:30 - 11:45 | 1 | 3 | 2 |
| 2 | Dr. Aaresh Bachana 26-DEC-2016 10:30 - 10:45 | 2 | 3 | 5 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select *from RecordsInfo;
+-----+-----+-----+-----+-----+-----+-----+
| RecordID | RApproval | RDocID | RPatID | RType | Rtestname | Status |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | NA | 1 | 3 | Results | Ultrasound | Results Uploaded |
| 2 | No | 1 | 3 | Test | MRI Scan | Pending for Approval |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

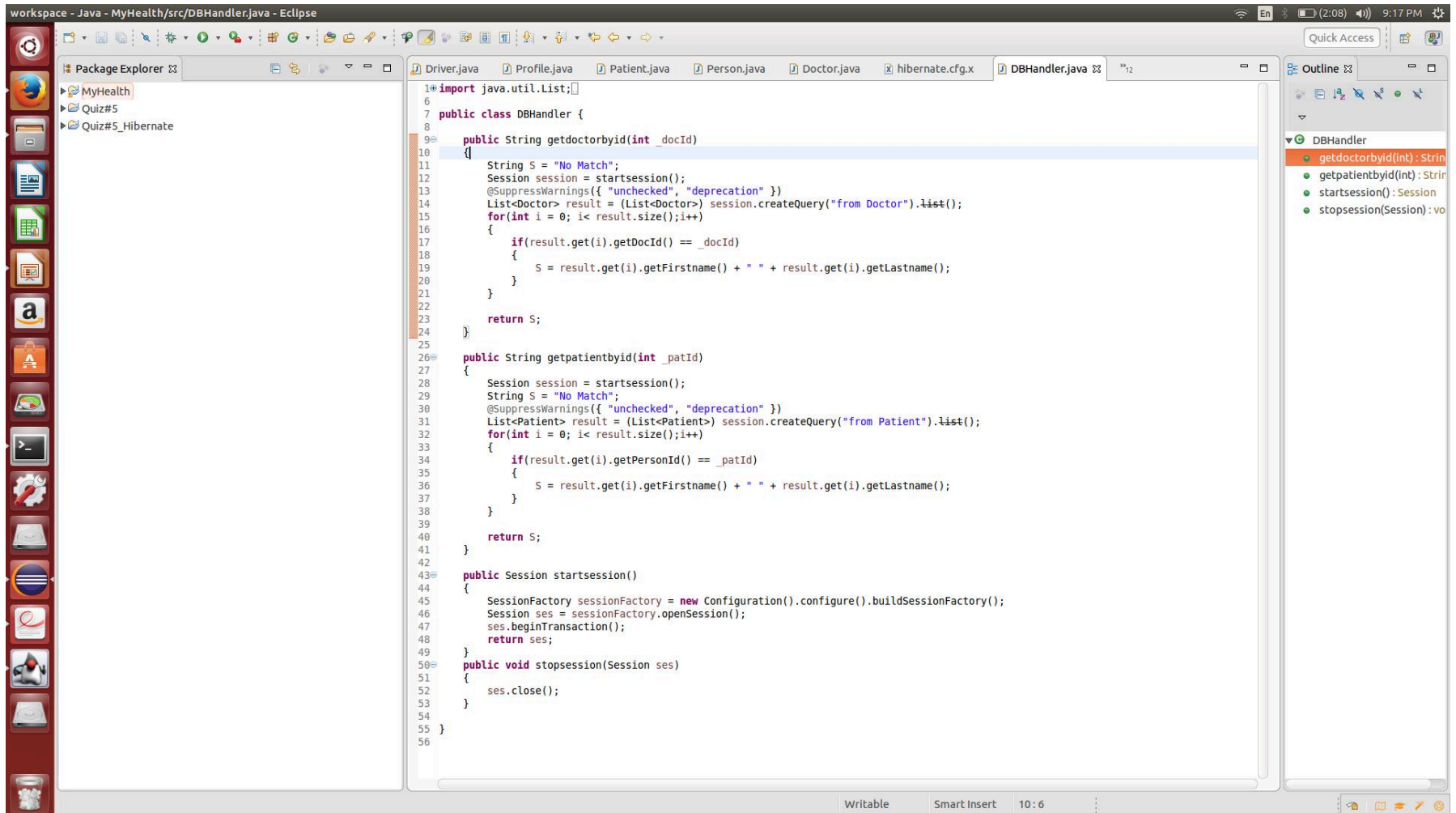
mysql>
```

Design Patterns Implemented

- Strategy Design Pattern



Refactoring



Future Development

- Refactoring the code
- Iterator Design Pattern
- Observer Design Pattern
- Strategy Design Pattern

Project Demo

- [Project demo link](#)

