****

**Dr. Help**

Version No.: 1.1

Date: 11/28/2017

Project Name: Dr. Help – Healthcare Management System

Presented by

Anantharaman Chandar – A20403439

Ashok Ramasami – A20441032

Gowrisankar Arumugam – A20400590

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Date | Prepared by / Modified by | Significant Changes |
| 1.0 | 11/03/2017 | Ananth  Ashok  Gowrisankar | Initial project proposal. |
| 1.1 | 11/28/2017 | Ananth  Ashok  Gowrisankar | Application Knowledge and Known Error Database. |

**WHO CAN REFER**

This document can be referred by the following persons as a part of ITMD510 Object Oriented Application Development

* James Papademas
* Shivam Narula
* Devan Chawla
* Anantharaman Chandar
* Ashok Ramasami
* Gowrisankar Arumugam

Table of Contents

[1. Introduction 6](#_Toc500093128)

[1.1 Cynosure of DR. HELP 6](#_Toc500093129)

[2. Technical Details 6](#_Toc500093130)

[2.1 Application Architecture 6](#_Toc500093131)

[2.2 Use case 8](#_Toc500093132)

[2.3 UML 8](#_Toc500093133)

[2.4 ERD 11](#_Toc500093134)

[2.5 Technology Used 11](#_Toc500093135)

[2.5.1 Front End: 11](#_Toc500093136)

[2.5.2 Back End: 11](#_Toc500093137)

[2.5.3 Technical Requirements: 11](#_Toc500093138)

[2.5.4 Domain Knowledge Requirements 11](#_Toc500093139)

[2.5.5 Tools Used: 12](#_Toc500093140)

[3. Login 12](#_Toc500093141)

[3.1 Login Screen: 12](#_Toc500093142)

[3.2 Screenshot 13](#_Toc500093143)

[3.2.1 Successful Login and Login time gets updated 13](#_Toc500093144)

[3.2.2 No user found for the registered domain: 14](#_Toc500093145)

[3.2.3 Invalid password for the registered user 14](#_Toc500093146)

[3.2.4 Label Description: 15](#_Toc500093147)

[3.2.5 Java Reference: 15](#_Toc500093148)

[3.2.6 Database Reference: 15](#_Toc500093149)

[3.2.7 Exception Handling 15](#_Toc500093150)

[4. Sign Up Screen 16](#_Toc500093151)

[4.1 User Registration Screen: 16](#_Toc500093152)

[4.1.1 Screenshot 16](#_Toc500093153)

[4.1.2 Label Description: 16](#_Toc500093154)

[4.1.3 Java Reference: 17](#_Toc500093155)

[4.1.4 Database Reference: 17](#_Toc500093156)

[4.1.5 Exception Handling 17](#_Toc500093157)

[5. User 18](#_Toc500093158)

[5.1 Issue Description screen 19](#_Toc500093159)

[5.1.1 Label Description: 19](#_Toc500093160)

[5.1.2 Java Reference: 20](#_Toc500093161)

[5.1.3 Database Reference: 20](#_Toc500093162)

[5.1.4 Exception Handling: 20](#_Toc500093163)

[5.2 Appointment Schedule Screen: 20](#_Toc500093164)

[5.2.1 Label Description: 21](#_Toc500093165)

[5.2.2 Java Reference: 21](#_Toc500093166)

[5.2.3 Database Reference: 21](#_Toc500093167)

[5.2.4 Exception Handling: 21](#_Toc500093168)

[5.3 Hospital List Screen: 22](#_Toc500093169)

[5.3.1 Label Description: 22](#_Toc500093170)

[5.3.2 Java Reference: 23](#_Toc500093171)

[5.3.3 Database Reference: 23](#_Toc500093172)

[5.4 Appointment Summary Screen: 23](#_Toc500093173)

[5.4.1 Label Description: 24](#_Toc500093174)

[5.4.2 Java Reference: 24](#_Toc500093175)

[5.4.3 Database Reference: 24](#_Toc500093176)

[6. Doctor 24](#_Toc500093177)

[6.1 Appointment Screen: 25](#_Toc500093178)

[6.1.1 Screenshot 25](#_Toc500093179)

[6.1.2 Label Description: 25](#_Toc500093180)

[6.1.3 Java Reference: 26](#_Toc500093181)

[6.1.4 Database Reference: 26](#_Toc500093182)

[6.1.5 Exception Handling 26](#_Toc500093183)

[6.2 Appointments details Screen: 26](#_Toc500093184)

[6.2.1 Screenshot 27](#_Toc500093185)

[6.2.2 Label Description: 27](#_Toc500093186)

[6.2.3 Java Reference: 28](#_Toc500093187)

[6.2.4 Database Reference: 28](#_Toc500093188)

[6.2.5 Exception Handling 28](#_Toc500093189)

[7. Manager 28](#_Toc500093190)

[7.1.1 Screenshot 29](#_Toc500093191)

[7.1.2 After a successful Login 30](#_Toc500093192)

[7.1.3 Enter Start Date and End Date to fetch reports 31](#_Toc500093193)

[7.1.4 Populate list of appointments for the users 31](#_Toc500093194)

[7.1.5 Update Logout time when Logout button is clicked 32](#_Toc500093195)

[7.1.6 Label Description: 32](#_Toc500093196)

[7.1.7 Java Reference: 33](#_Toc500093197)

[7.1.8 Database Reference: 33](#_Toc500093198)

[7.1.9 Exception Handling 34](#_Toc500093199)

[8. Admin 34](#_Toc500093200)

[8.1 Screenshot 35](#_Toc500093201)

[8.1.1 After a successful Login 36](#_Toc500093202)

[8.1.2 If Admin wants to Perform a DML Activity 37](#_Toc500093203)

[8.1.3 Basic Syntax for every DBA Activity populates automatically 37](#_Toc500093204)

[8.1.4 Invalid Ticket Number or Ticket is not in Open Status 38](#_Toc500093205)

[8.1.5 SQL Syntax Error if Query is wrong 38](#_Toc500093206)

[8.1.6 If SQL Query is valid 39](#_Toc500093207)

[8.1.7 Label Description: 39](#_Toc500093208)

[8.1.8 Java Reference: 39](#_Toc500093209)

[8.1.9 Database Reference: 40](#_Toc500093210)

[8.1.10 Exception Handling 40](#_Toc500093211)

[9. Raise A Request 40](#_Toc500093212)

[9.1.1 Standard Template Followed 41](#_Toc500093213)

[9.2 Screenshot 41](#_Toc500093214)

[9.2.1 Label Description: 41](#_Toc500093215)

[9.2.2 Java Reference: 42](#_Toc500093216)

[9.2.3 Database References 42](#_Toc500093217)

[9.2.4 Exception handling 42](#_Toc500093218)

[10. Exception Handling 42](#_Toc500093219)

[10.1.1 Login Error Handler: 42](#_Toc500093220)

[10.1.2 Empty Field Error Handler: 43](#_Toc500093221)

[10.1.3 SQL Error Handler 43](#_Toc500093222)

[10.2 Screenshot 44](#_Toc500093223)

[10.2.1 Login Error Handler 44](#_Toc500093224)

[10.2.2 Empty Fields Error Handler 45](#_Toc500093225)

[10.2.3 SQL Error Handler 45](#_Toc500093226)

[10.2.4 Java Reference 46](#_Toc500093227)

[10.2.5 Database Reference 46](#_Toc500093228)

[11. References 46](#_Toc500093229)

[11.1 Java 46](#_Toc500093230)

[11.2 Database 47](#_Toc500093231)

[11.2.1 List of tables 47](#_Toc500093232)

[11.2.2 Column Details 48](#_Toc500093233)

[12. Future Scope 52](#_Toc500093234)

# Introduction

**Dr. Help** is a stand-alone application which is can be accessed by Users to book, cancel medical consultations. Doctors can view, update medical records of the User. Higher Management can view the Hospitals metrics, the performance of the Doctors, Users consultation report and provide feedback to the Developers. Admins to perform DBA activities and Ad-hoc request raised by Higher Management.

## Cynosure of DR. HELP

* Login made easy by selecting your email domain from the drop-down box
* MD5 Password authentication is provided for making the password secure
* Handled maximum exceptional handling and recorded all the exceptions in the database table to improve this application.
* Choosing medical issue made easy by classifying your medical issue into three (3) categories
* Doctors can view, update user’s medical records
* Provided in-built report generation to some of the most important reports which will be fetched from the database and listed in the screen
* Admins are provided with the syntax for their IT Operations.
* Maintain Login and Logout time for each individual.

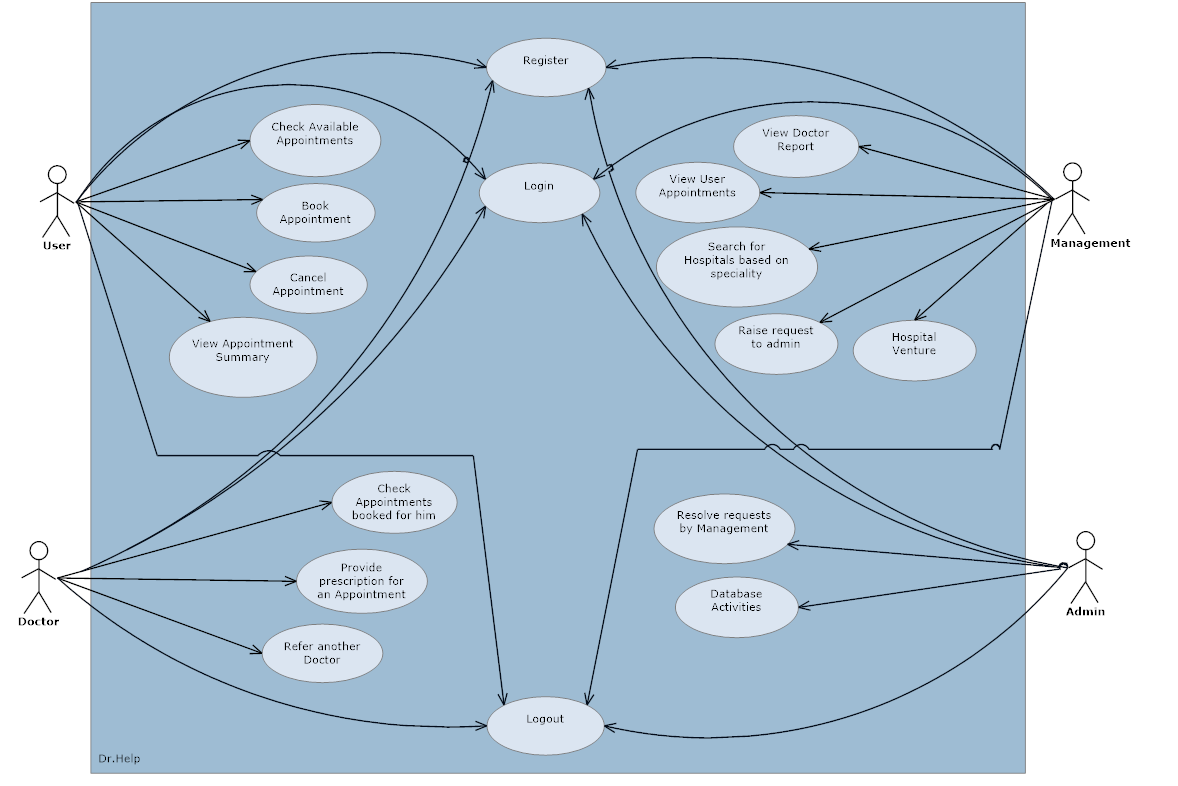
# Technical Details

## Application Architecture

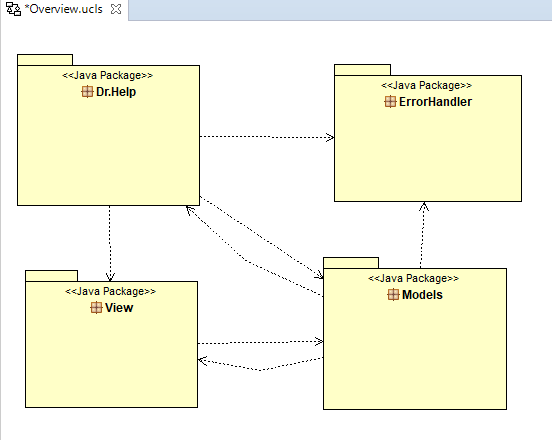
Below is the overview of the application and the navigation inside the application depending upon the category

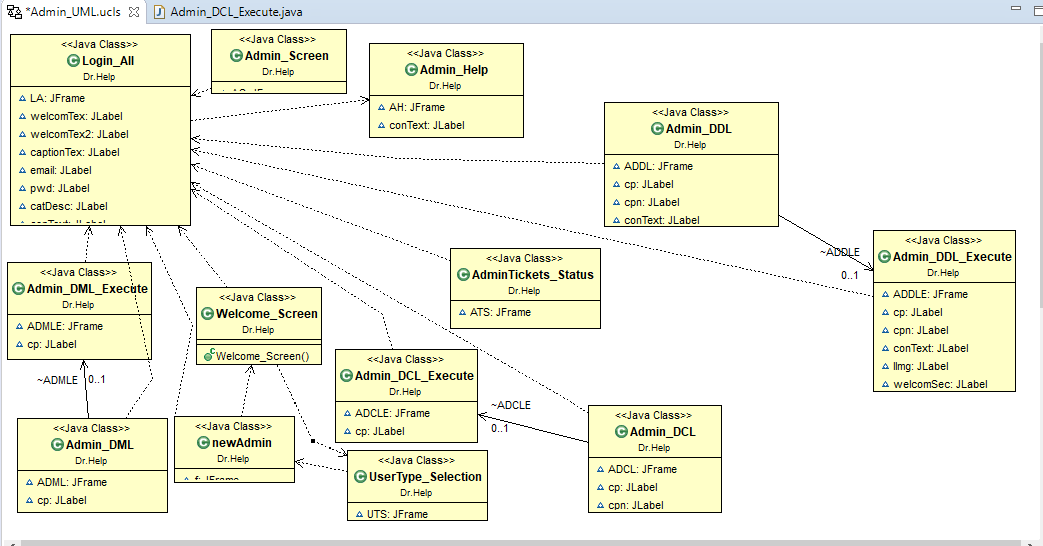


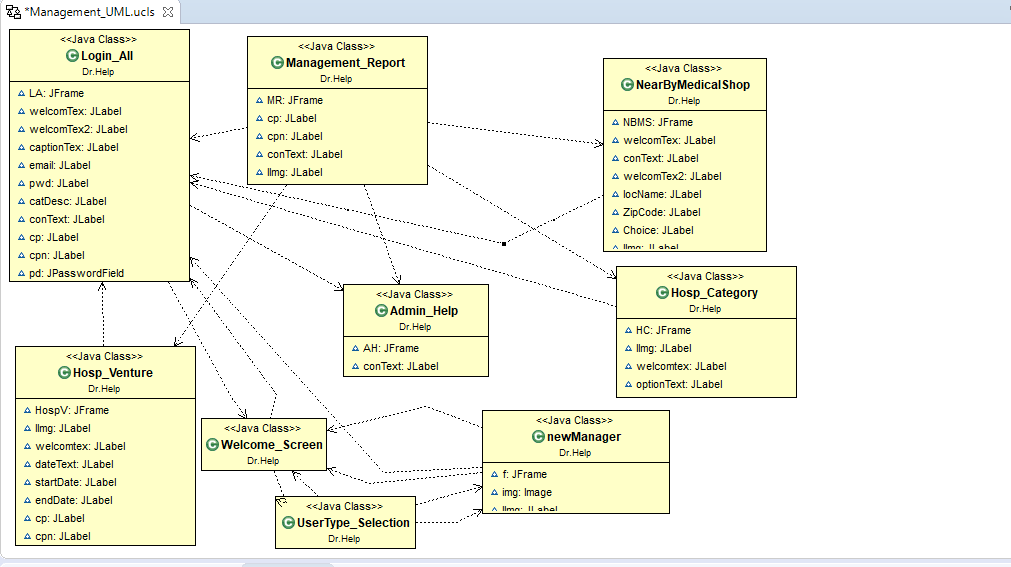
## Use case

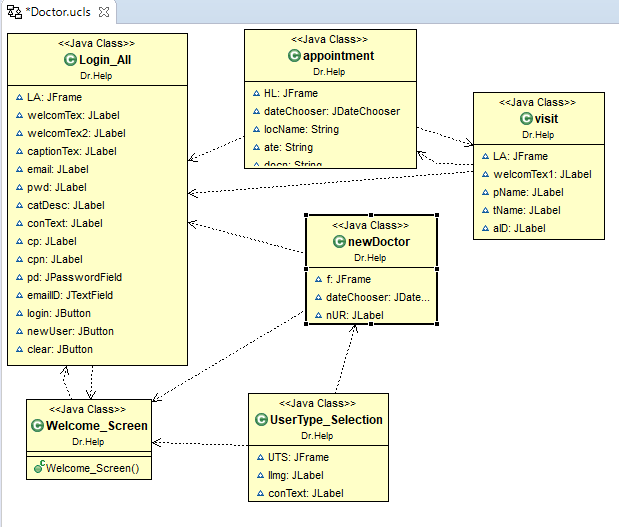


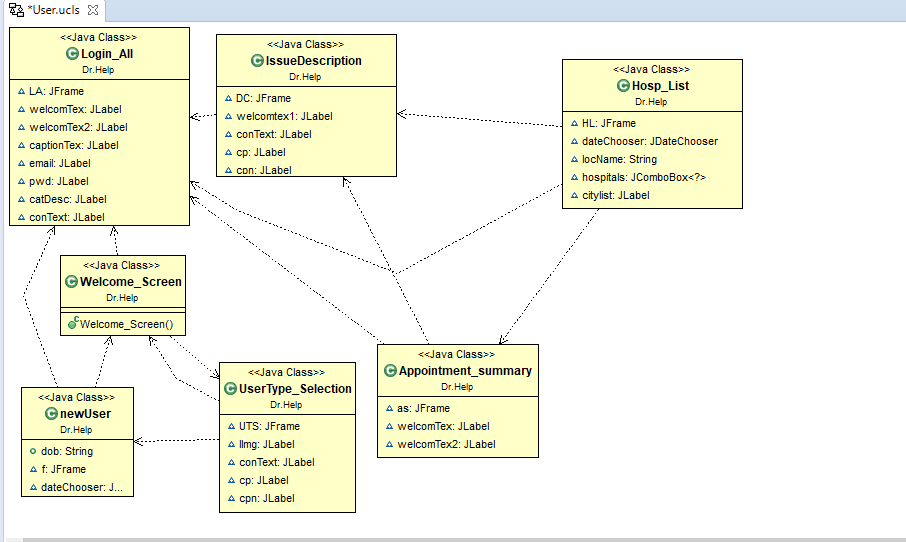
## UML



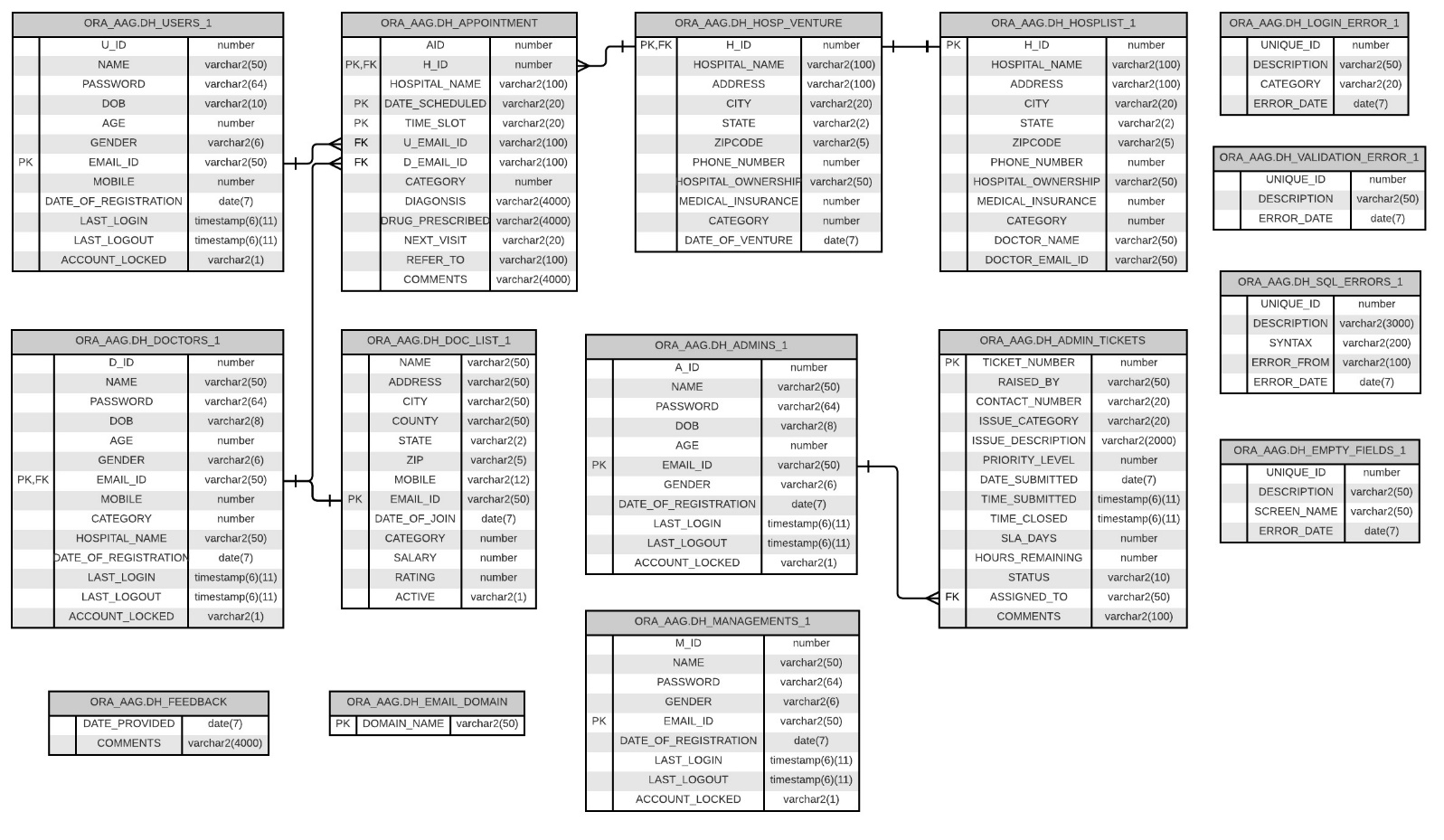








## ERD



## Technology Used

### Front End:

JavaFX: Software platform for creating and delivering desktop applications.

### Back End:

Oracle SQL Developer: Integrated development environment for working with SQL in Oracle databases

### Technical Requirements:

Below is the technology that needed to build this project

Java

JavaFX

Oracle Database or MySQL

### Domain Knowledge Requirements

Health Care System or Medical Consultation system or Hospital appointment consultation

### Tools Used:

Java Eclipse EE Oxygen

Oracle SQL Developer

# Login

## Login Screen:

The login screen will be displayed when the application is started and will be a common screen for every individual. Depending upon the Category the Login screen will redirect to the respective screens.

When an individual login he/she must enter their Email-Id in the Email-Id Textbox and select the domain from the drop-down. If there are any discrepancies such as domain name in the Email-Id text box, use @ symbol in the Email-Id textbox, not entering any values in the Email-Id text box all these are covered as a part of exception handling.

If the password value is not entered and left as blank then an error message pops up saying Password Field is empty.

The category defines what type of individual you are either USER or DOCTOR or MANAGEMENT, ADMIN. If the category is not selected then an error message pops up saying category must be selected.

Once all the required fields are filled the following validations will be executed to determine whether the entered details are valid or invalid.

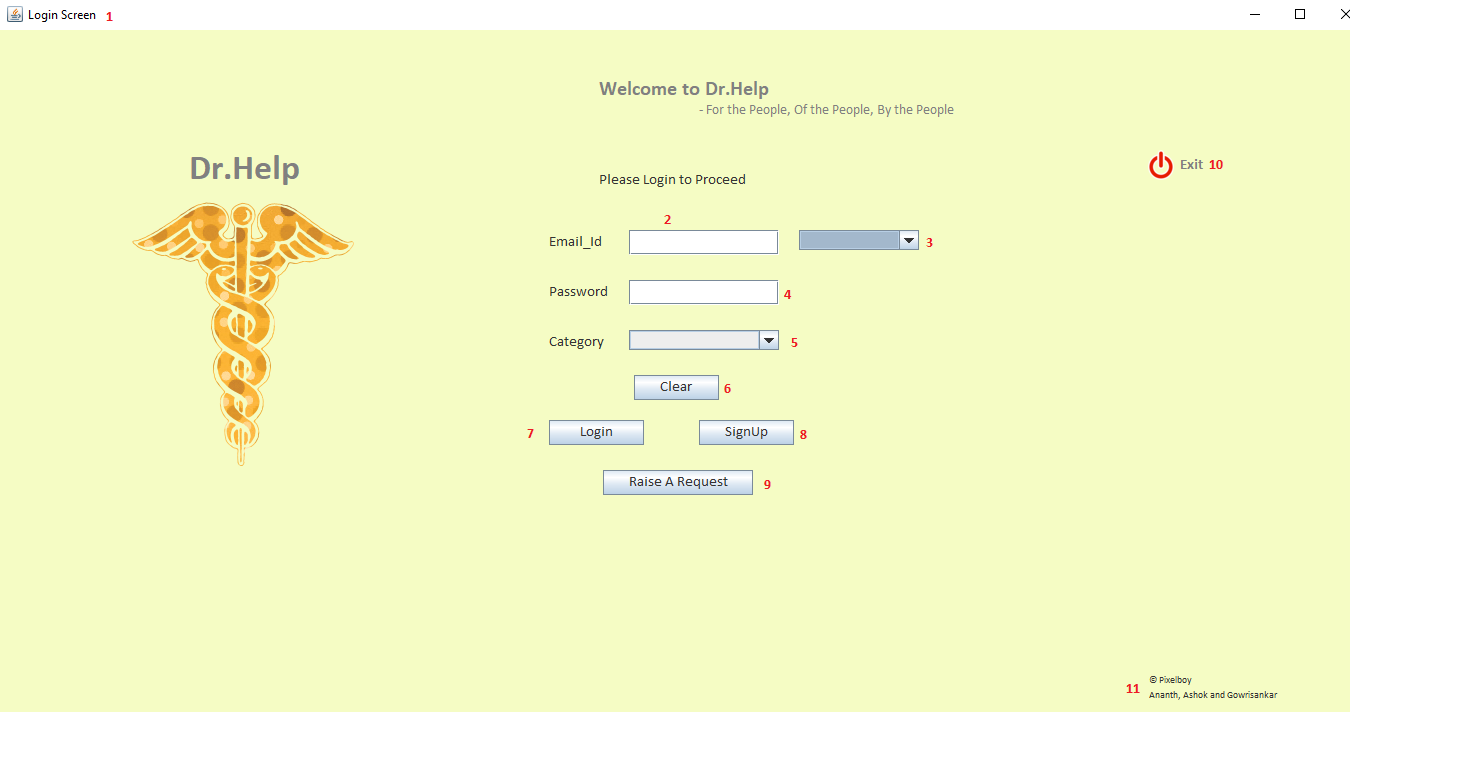
Depending upon the category selected the application will check whether there is an Email-ID available for the entered value. If the Email-Id is not available then an error message pops up saying no user found in the selected category.

If the Email-Id is available then the entered password is converted to a Hash value format and validates against the existing password in the database which is also stored in a Hash value format. If the Hash value is not matched then an error message pops up saying Incorrect Password for the entered user. If password matches then Login time is captured and it will be updated to the logged in user.

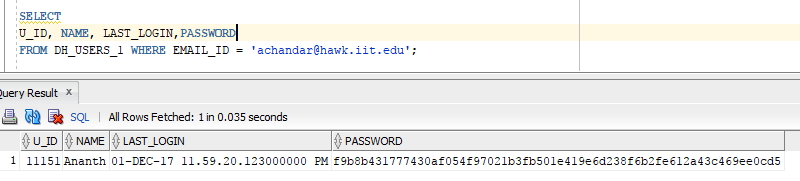
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No** | **Email-ID** | **Domain** | **Password** | **Category** | **Description** |
| 1 | achandar | @hawk.iit.edu | [Ananth@1234](mailto:Ananth@1234) | User | Successful Login |
| 2 | achandar | @gmail.com | [Ananth@1234](mailto:Ananth@1234) | User | No user found for this registered category |
| 3 | achandar | @hawk.iit.edu | [Ananth@1000](mailto:Ananth@1000) | User | Invalid password for the registered user |

Similarly, the same validations are implemented if an individual chooses the category DOCTOR or MANAGEMENT, ADMIN.

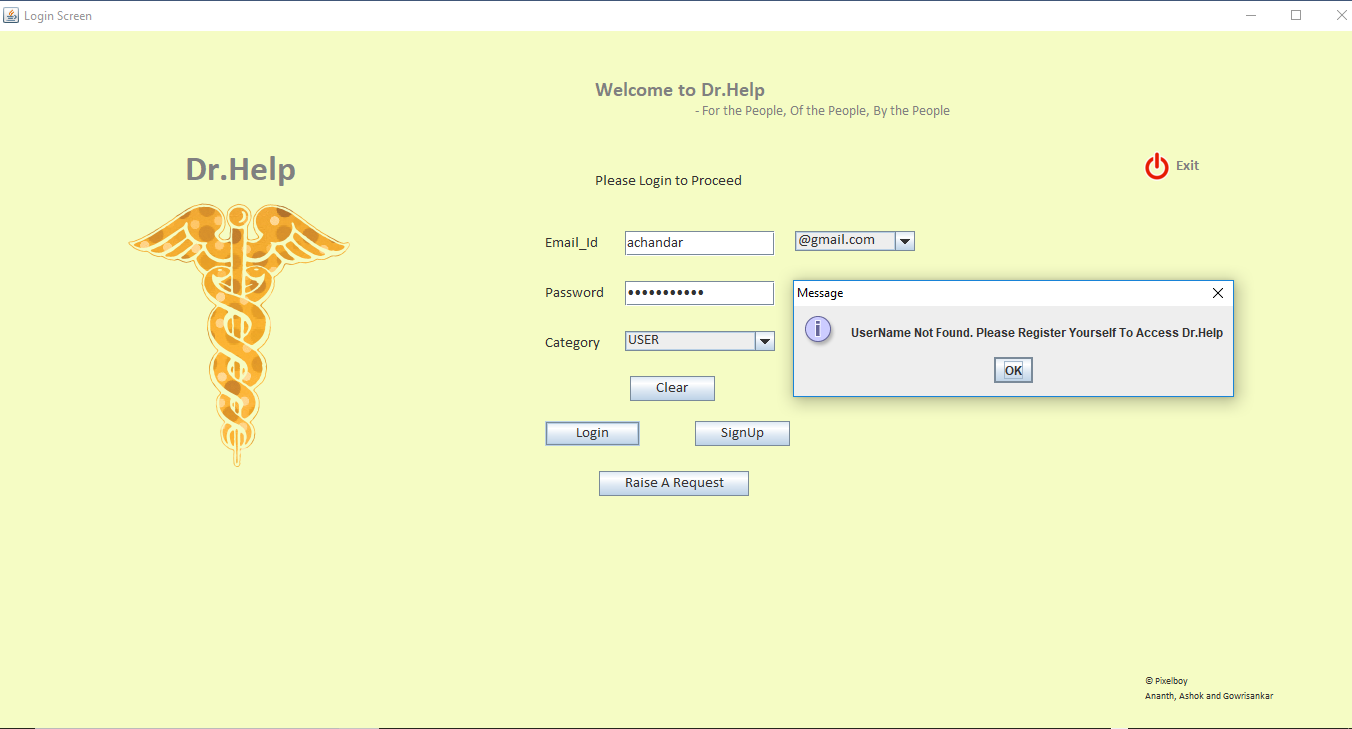
## Screenshot



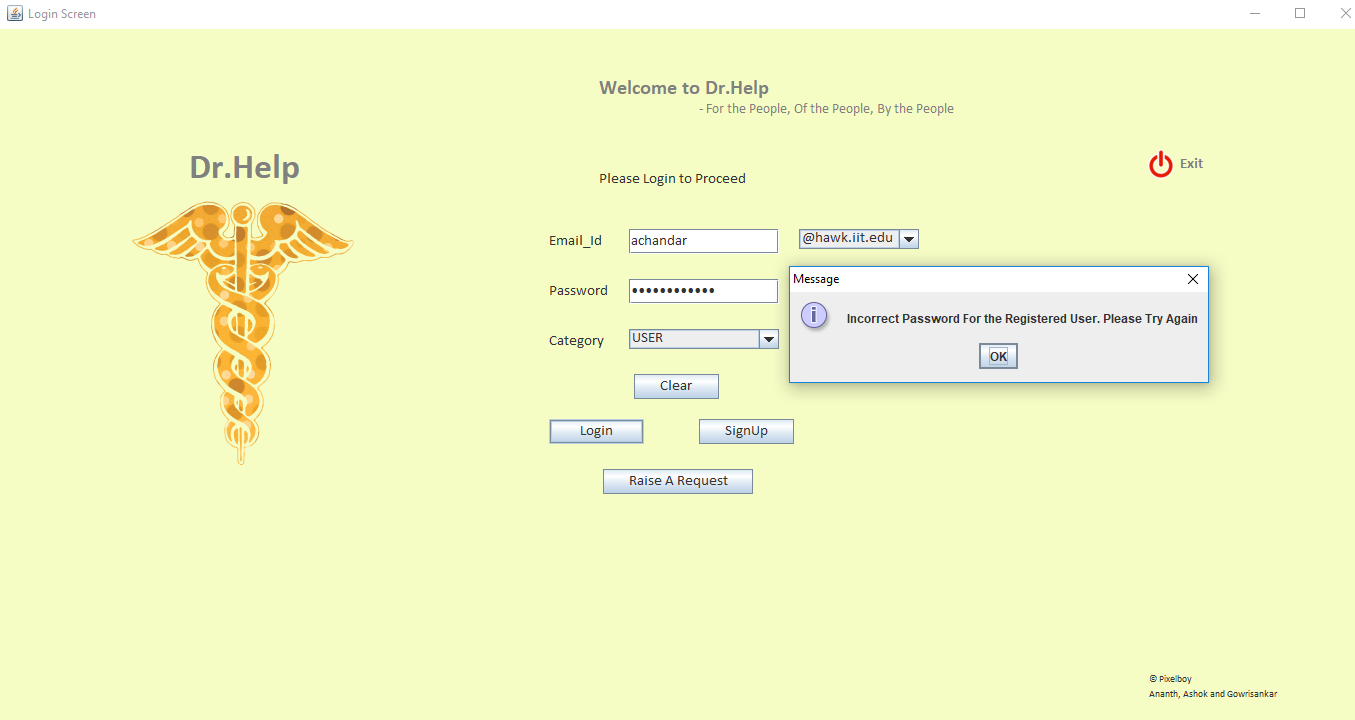
### Successful Login and Login time gets updated



### No user found for the registered domain:



### Invalid password for the registered user



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Login Screen | JFrame Name |
| 2 | Email\_Id | To enter Email\_Id |
| 3 | Dropdown | Select email domain |
| 4 | Password | To enter Password for the user |
| 5 | Category | Identify yourself whether User or Doctor or Management or Admin |
| 6 | Clear | Clear all the details on the Login Screen |
| 7 | Login | Validates and route the individual to corresponding screens |
| 8 | Sign up | New Registration for Dr. Help |
| 9 | Raise A Request | Raise a ticket for Admin Support team |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | Login\_All | ResultSetFetch.getLoginDetails | - | - |
|  |  | ResultSetFetch.getEmailCount |  |  |
|  |  | ResultSetFetch.getAccountStatus |  |  |
|  |  | ResultSetFetch.updateLoginTime |  |  |
|  |  | ResultSetFetch.get\_SecurePassword |  |  |

### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | DH\_USERS\_1 | Registered USERS details |
| 2. | DH\_DOCTORS\_1 | Registered DOCTORS details |
| 3. | DH\_MANAGEMENTS\_1 | Registered higher MANAGEMENT details |
| 4. | DH\_ADMINS\_1 | Registered ADMINS details |
| 5. | DH\_EMPTY\_FIELDS | If any fields are filled empty |
| 6. | DH\_LOGIN\_ERROR\_1 | Login Validation failure |

### Exception Handling

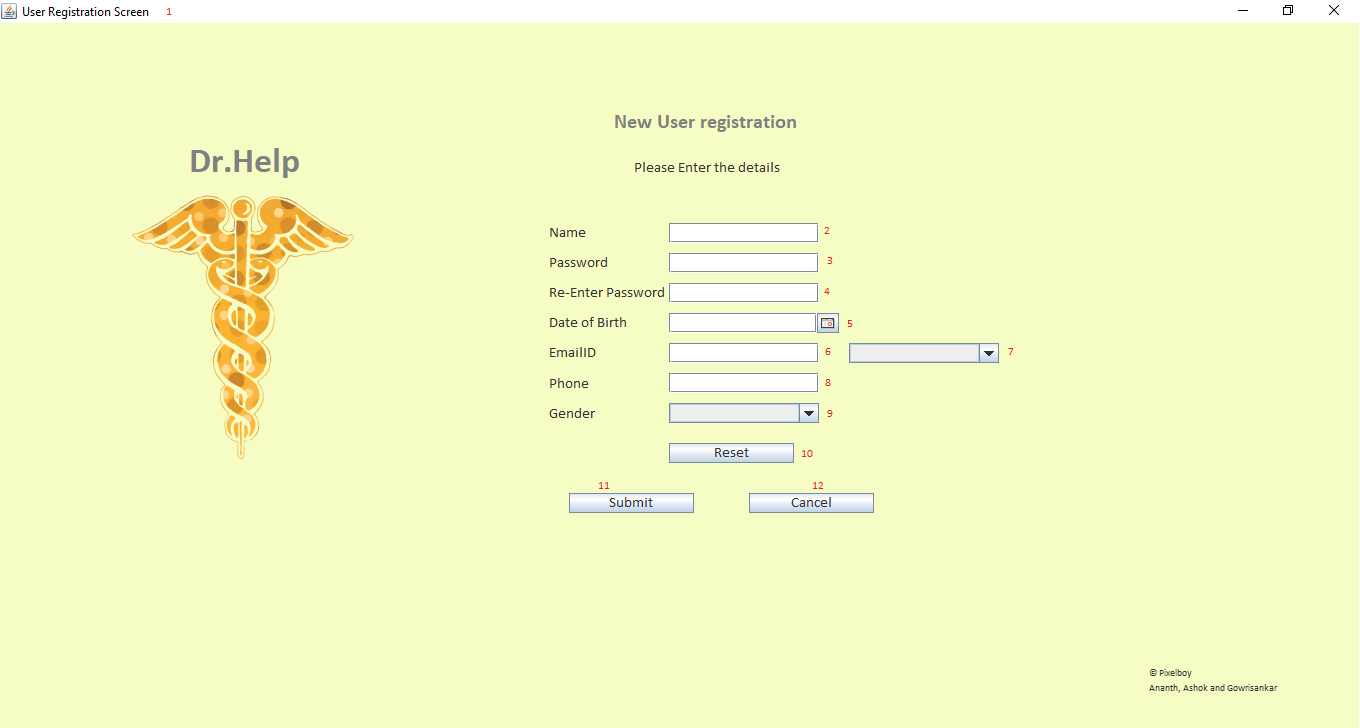
|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1. | Email\_Id | Empty | DH\_EMPTY\_FIELDS\_1 |
| 2. | Dropdown for Domain | Empty | DH\_EMPTY\_FIELDS\_1 |
| 3. | Password | Empty | DH\_EMPTY\_FIELDS\_1 |
| 4. | CATEGORY | Empty | DH\_EMPTY\_FIELDS\_1 |
| 5. | Incorrect Password | Password Mismatch | DH\_LOGIN\_ERROR\_1 |
| 6. | Incorrect Email-Id | No User Found | DH\_LOGIN\_ERROR\_1 |

# Sign Up Screen

The Sign-up screen is the screen through which new registration can be done, the individual can be a User, Doctor, Management team member, Admin. Here the basic details of the individual are needed like name, password, date of birth, email-id, phone number, and gender. There is separate screen for registration of a User, Doctor etc, though the details required for registration is common for all kinds of users.

## User Registration Screen:

### Screenshot



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | User Registration Screen | JFrame name |
| 2 | Name | Text field to enter the name of the individual (mandatory field) |
| 3 | Password | Password field for the individual's account |
| 4 | Re-Enter Password | Password field to confirm that the individual doesn’t enter random password |
| 5 | Date of Birth | Date picker to get the date of birth of the individual |
| 6 | Email-Id text box | Text field for the individual to enter his/her email id so that already registered mail id cannot be used |
| 7 | Mail domain | Combo box with email domains available for registration |
| 8 | Phone number | Text field where the individual can input his/her personal contact number |
| 9 | Gender | Combo box to choose the individual’s gender |
| 10 | Reset | JButton to reset the inputs in all the fields in the screen |
| 11 | Submit | JButton to submit the account details and create a new account |
| 12 | Cancel | JButton to cancel the registration process. |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | Welcome\_Screen | - | - | Login\_All |
| 2. | UserType\_Selection | - | - | Welcome\_Screen |
| 3. | newUser, newAdmin, newDoctor,  newManager | InsertInToDB.insertNewUsers  InsertInToDB.insertNewDoctor  InsertInToDB.insertNewManager  InsertInToDB.insertNewAdmin | - | UserType\_Selection |

### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred for** |
| 1. | DH\_USERS\_1 | Inserting the account details into the database for new User registration |
| 2. | DH\_DOCTORS\_1 | Inserting the account details into the database for new Doctor registration |
| 3. | DH\_MANAGEMENTS\_1 | Inserting the account details into the database for new Manager registration |
| 4. | DH\_ADMINS \_1 | Inserting the account details into the database for new Admin registration |
| 5. | DH\_EMPTY\_FIELDS\_1 | When the individuals didn’t fill in any mandatory fields during registration |

### Exception Handling

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1. | Name | Not filled | DH\_EMPTY\_FIELDS\_1 |
| 2. | Password | Not filled | DH\_EMPTY\_FIELDS\_1 |
| 3. | Re-Enter Password | Not filled | DH\_EMPTY\_FIELDS\_1 |
| 4. | Password Mismatch | If fields Passwords and Re-Enter password field inputs mismatch | DH\_EMPTY\_FIELDS\_1 |
| 5. | Date of Birth | Not Selected | DH\_EMPTY\_FIELDS\_1 |
| 6. | Email-Id | Not Filled,  Contains special character “@” | DH\_EMPTY\_FIELDS\_1 |
| 7. | Email domain | Not Selected | DH\_EMPTY\_FIELDS\_1 |
| 8 | Phone | Not Filled,  Exceeds 10 digits | DH\_EMPTY\_FIELDS\_1 |
| 9. | Gender | Not Selected | DH\_EMPTY\_FIELDS\_1 |

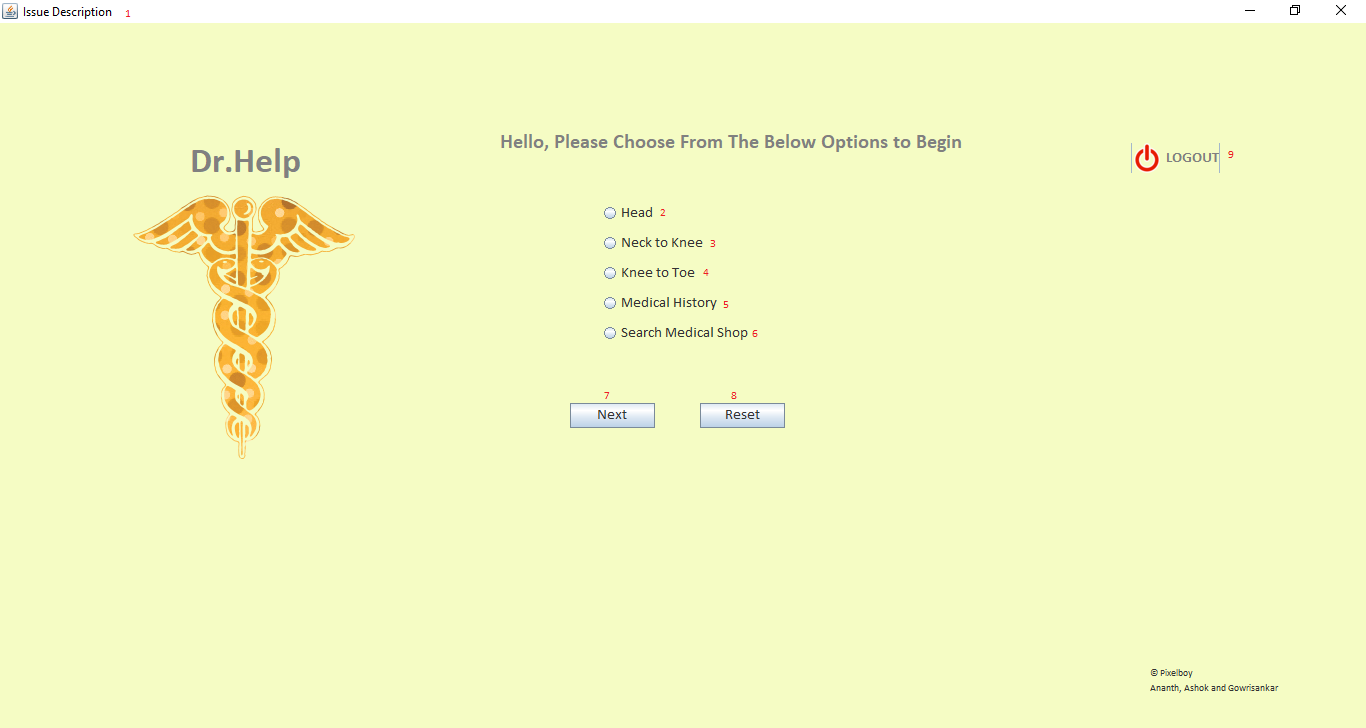
# User

Once the individual logs in as a User he has access to four screens namely Issue Description, Appointment Schedule, Hospital List, Appointment Summary screen. The operations possible for User Selecting a category for which he needs to visit the doctor, search for pharmacy available in the needed location. Choosing the city in which he wants to search for a hospital. Choose between “Having Medical Insurance/No Medical Insurance” to list the hospitals with cheaper fare. Select a hospital from the list based on previous responses and check the available time slots on a day of his choice. Book an appointment two day from now until 30 days on his own choice provided the slot is available. Cancel the appointment if he doesn’t need it. Finally, he can log out of the application which will lead to the start screen of the application (“Login All screen”).

Below are the Login ID and Password for logging in to User Section and book an appointment. Once a successful login is made the login time is updated for the individual. Any Invalid Email-Id and Password will display error message explained in the [Login Section](#_Login_Screen:)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No** | **Email-ID** | **Domain** | **Password** | **Category** | **Description** |
| 1 | achandar | @hawk.iit.edu | [Ananth@1234](mailto:Ananth@1234) | User | Successful Login |

## Issue Description screen



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Issue Description | JFrame Name |
| 2 | Head | Radio Button for letting the user book appointments in the hospitals that specialize treatment related to organs in Head |
| 3 | Neck To Knee | Radio Button for letting the user book appointments in the hospitals that specialize treatment related to organs in between Neck and the Knees. |
| 4 | Knee to Toe | Radio Button for letting the user book appointments in the hospitals that specialize treatment related to organs in between Knee and the Toes. |
| 5 | Medical History | The radio button to be selected if the user needs to check his previous appointments and medications. |
| 6 | Search Medical Shop | Radio Button to be selected if the user needs to search for nearby available medical shops. |
| 7 | Next | JButton to move to the next screen along with the selection of the above radio button |
| 8 | Reset | JButton to clear the selection made by the user. |
| 9 | Logout | JButton to log out of the application. |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | IssueDescription | - | - | Login\_All |

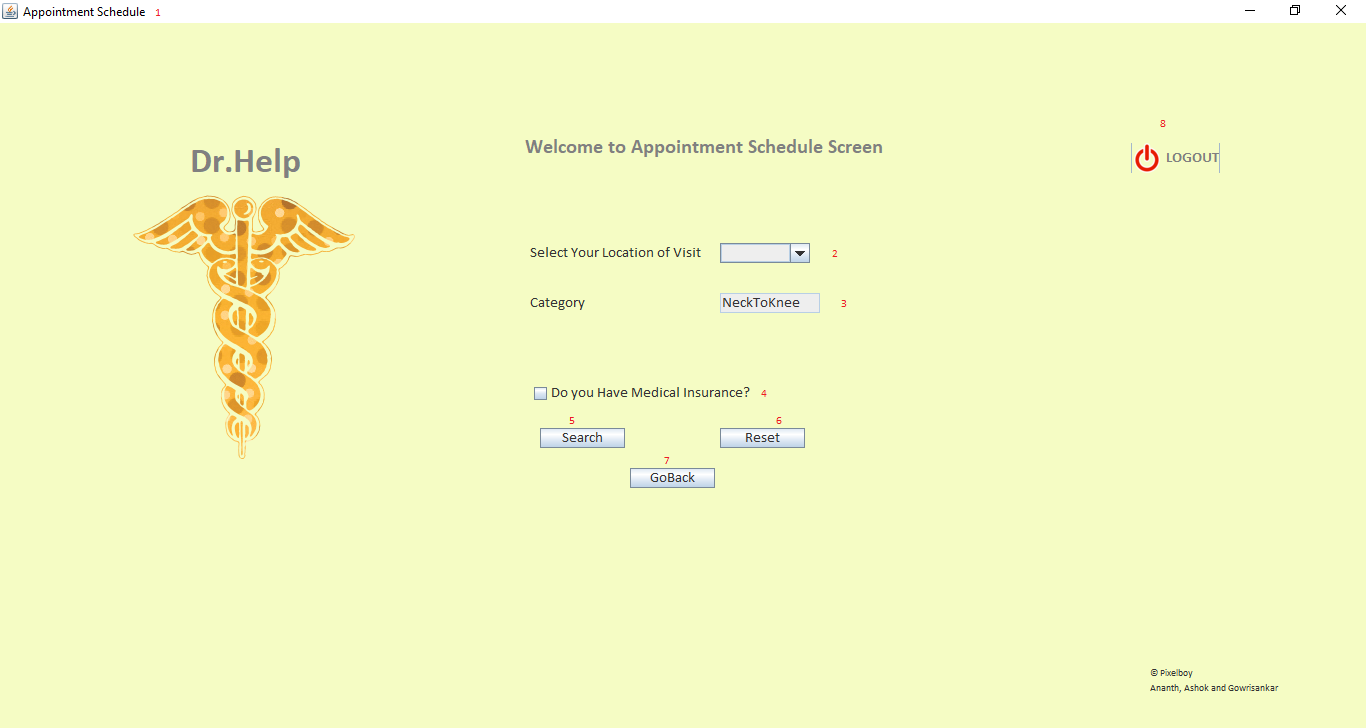
### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred for** |
| 1. | DH\_EMPTY\_FIELDS\_1 | When there is no selection made by the user. |

### Exception Handling:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1. | Radio Buttons (Head, Neck To Knee, Knee to Toe, Medical history, Search Medical Shop) | No Selection | DH\_EMPTY\_FIELDS\_1 |

## Appointment Schedule Screen:



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Appointment Schedule | JFrame Name |
| 2 | Location | Combo box lists the cities where hospitals have been registered with Dr. Help |
| 3 | Category | Text field prefilled with the user choice of category from the Issue Description screen (Non\_Editable) |
| 4 | Insurance | Checkbox whether to know if the user has Medical insurance or not to suggest hospitals with cheap fare for users with no insurance |
| 5 | Search | JButton to search for hospitals based on the city selection by the User |
| 6 | Reset | JButton to reset all the choices on the screen made by the User |
| 7 | GoBack | JButton to redirect the User to the previous screen (“Issue Description”) |
| 8 | Logout | JButton to log out of the application. |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | UserAppConfirmation | - | - | IssueDescription |

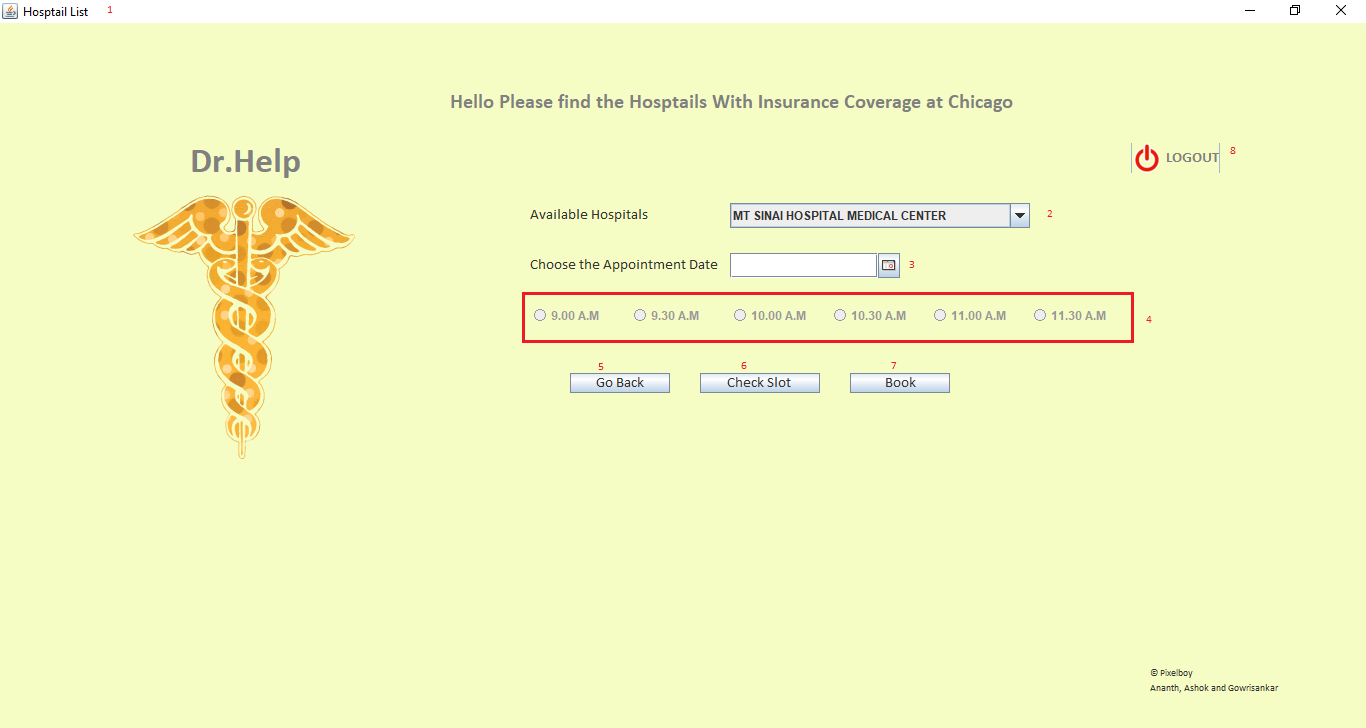
### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred for** |
| 1. | DH\_EMPTY\_FIELDS\_1 | when there is no city selection made by the user. |

### Exception Handling:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1. | Location | No Selection | DH\_EMPTY\_FIELDS\_1 |

## Hospital List Screen:



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Hospital List | JFrame Name |
| 2 | Available Hospitals | Combo box lists the available hospitals based on the user selection from the previous screen |
| 3 | Appointment date | Date picker for selecting the date of user choice for booking an appointment |
| 4 | Time Slot | Group of Radio Buttons that are enabled based on the time available of the hospital selected on the day chose by the user |
| 5 | GoBack | JButton to redirect the User to the previous screen (“Issue Description”) |
| 6 | Check Slot | JButton to check the available time slots from the database for the date and hospital selection, enable only radio buttons with available time slots. |
| 7 | Book | JButton to Book the appointment for the User selection and insert the details into the database. |
| 8 | Logout | JButton to log out of the application. |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | Hosp\_List | ResultSetFetch.hosplist | - | UserAppConfirmation |
|  |  | ResultSetFetch.checkdate |  |  |
|  |  | ResultSetFetch.getdoc |  |  |
|  |  | InsertInToDB.bookappointment |  |  |

### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred for** |
| 1. | DH\_HOSPLIST\_1 | Getting Hospitals and Doctor associated with the Hospital based on the city selected by the User |
| 2. | DH\_APPOINTMENT | Finding Available time slots in the hospital, inserting the booking details selected by user |

## Appointment Summary Screen:



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Summary Screen | JFrame Name |
| 2 | Appointment Number | JLabel displays the appoint number booked by the User |
| 3 | Hospital Name | JLabel displays the User Selected Hospital name from previous screen and city from Appointment screen |
| 4 | Visiting Category | JLabel displays the User selected category from Issue Description screen. |
| 5 | Appointment Schedule | JLabel displays the Doctor name associated with the hospital selected by the user. |
| 6 | Date of Visit | JLabel displays the date User chose in the previous screen. |
| 7 | Time Scheduled | JLabel displays the time slot chosen by the User for the appointment. |
| 8 | Cancel | JButton to cancel the appointment the user just made. |
| 9 | Book New | JButton to redirect the User to Issue Description screen, to begin with the booking of a new appointment. |
| 10 | Logout | JButton to log out of the application. |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | Appointment\_summary | ResultSetFetch.getsummary | - | Hosp\_List |
|  |  | InsertInToDB.cancelappointment |  |  |

### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred for** |
| 1. | DH\_APPOINTMENT | Getting Summary details of the User booked appointment and delete the record if User cancel the appointment |

# Doctor

The appointments scheduled by the users are then accessed by doctors through their login. When a doctor logs in to the application, we will be taken to a screen where he can check the appointments. Once he selects an appointment and clicks on the “Details” button, he will be redirected to a page which displays the information about appointment which was selected. Each doctor is linked to a hospital based on the category. Sample login details for doctor login.

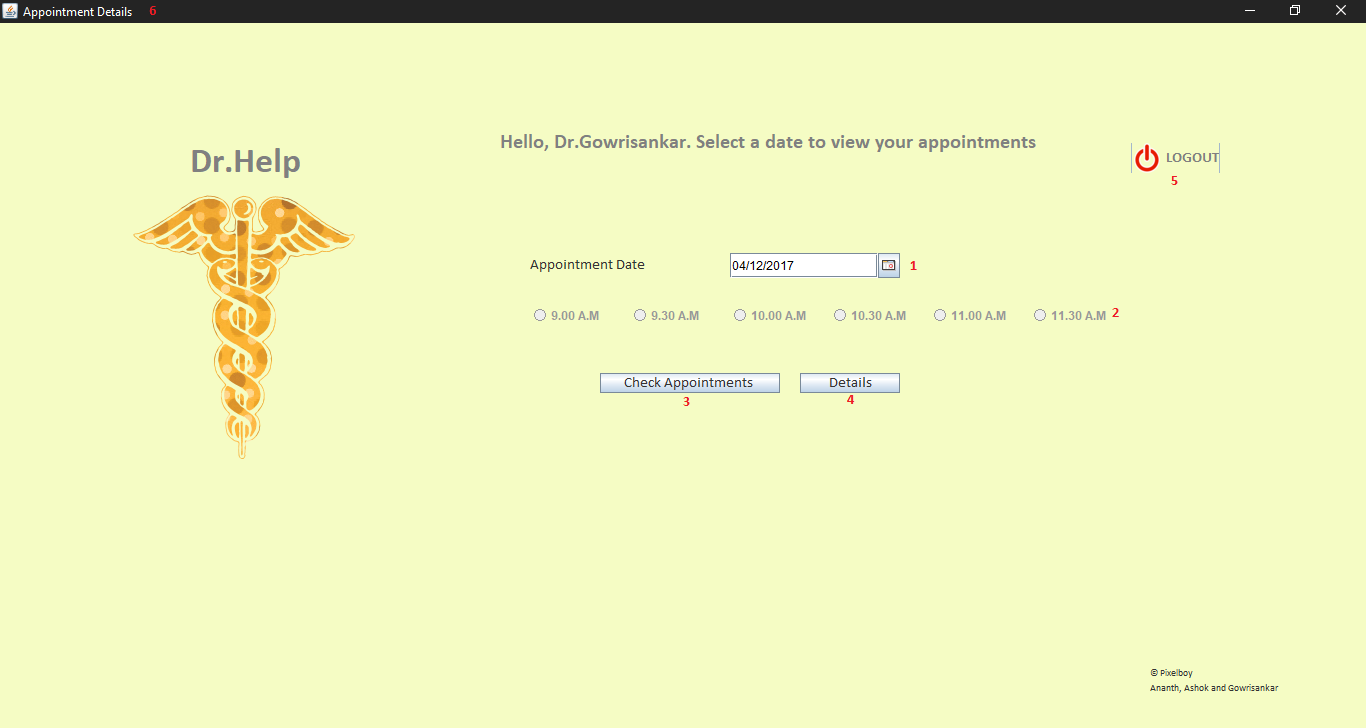
Below are the Login ID and Password for logging in to Doctor Report Section and medical records. Once a successful login is made the login time is updated for the individual. Any Invalid Email-Id and Password will display error message explained in the [Login Section](#_Login_Screen:)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No** | **Email-ID** | **Domain** | **Password** | **Category** | **Description** |
| 1 | garumugam | @hawk.iit.edu | [itmd510](mailto:Ananth@1234) | Doctor | Successful Login |

## Appointment Screen:

The doctor is greeted with an appointment availability check screen where he can select a date which is 30 days from today. On clicking on the “Check appointments” button doctor will be able to select only the time slots for which appointments are booked. All the time slots which are not booked will be disabled and the doctor will not be able to select them. After the doctor has selected a particular time slot he can click on “Details” button to make any changes to the appointment.

### Screenshot



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Appointment Date | Date chooser where the doctor can select a date for which he can check the appointments for. |
| 2 | Time Slot | Displays the available time slots |
| 3 | Check Appointments | Clicking on this button will refresh the time slots for the doctor to select the time slots which are booked for a particular date. |
| 4 | Details | Clicking on this button will take the Doctor to the next page to edit the details of the appointment. |
| 5 | Logout | Logout from the current session. |
| 6 | Appointment details | Screen name |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1 | Appointment.java | ResultSetFetch.hosplist  ResultSetFetch.checkdate | - | Login\_All |

### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred for** |
| 1 | DH\_DOCTORS\_1 | To retrieve the Name of the doctor who has logged in |
| 2 | DH\_APPOINTMENT | To display the time slot booked for a particular day. |

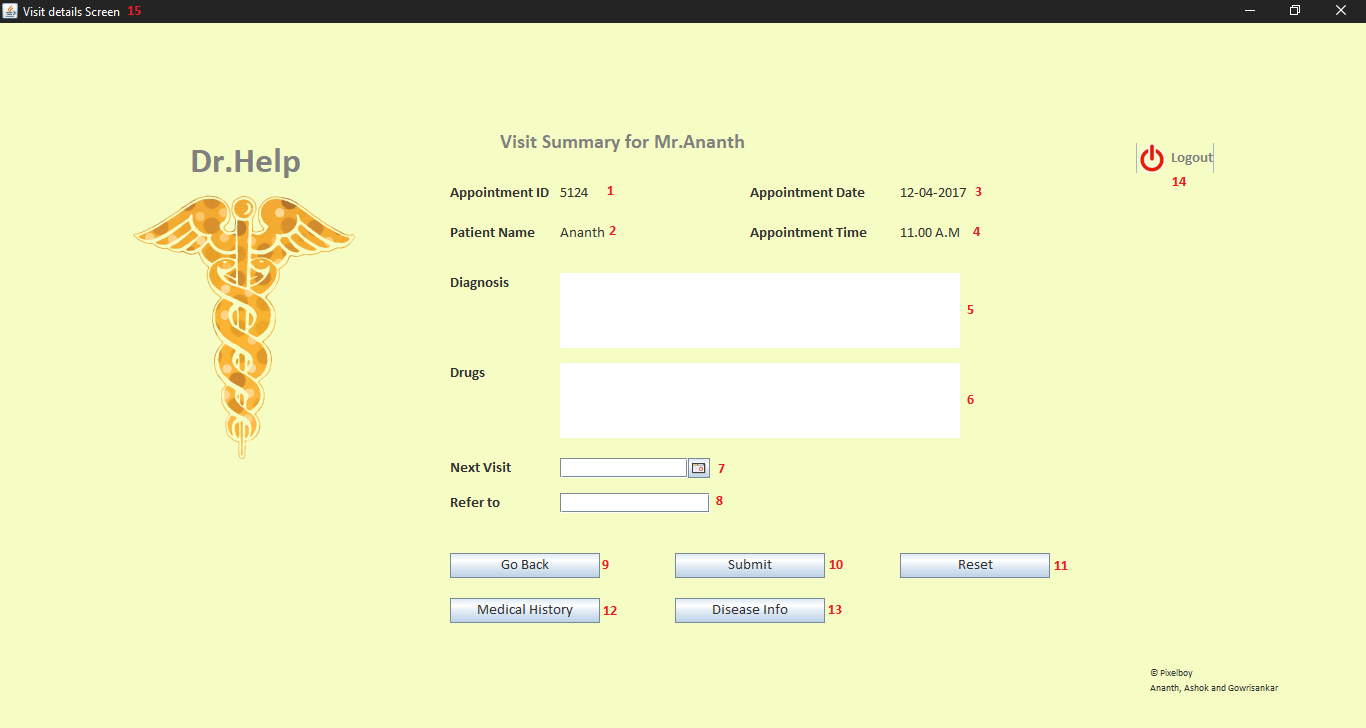
### Exception Handling

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1 | Appointment date | Any date which is lesser than today or greater than 30 days from now is selected. | - |

## Appointments details Screen:

The appointments details screen will be the preceding screen on the Check Appointments screen. The screen will be used by the Doctor to update the Diagnosis, Drugs prescribed, date of the next visit and refer to any other doctor if he needs to refer a particular patient to another doctor. There are also options for the doctor to see the medical history of a patient and look up details of diseases. These two options are not yet incorporated in the application and will be added in the future.

### Screenshot



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Appointment ID | Displays the appointment ID for the appointment selected by the doctor in the previous screen. |
| 2 | Patient Name | Displays the Patient Name for the appointment selected by the doctor in the previous screen. |
| 3 | Appointment Date | Displays the Appointment date for the appointment selected by the doctor in the previous screen. |
| 4 | Appointment Time | Displays the Appointment time for the appointment selected by the doctor in the previous screen. |
| 5 | Diagnosis | Text box for the doctor to input the diagnosis for the patient. |
| 6 | Drugs | Text box for the doctor to input the drugs for the patient. |
| 7 | Next Visit | A date chooser box for the doctor to select a date for the next visit. |
| 8 | Refer to | Text box for the doctor to refer the patient to another doctor. |
| 9 | Go Back | Takes the Doctor to the previous screen. |
| 10 | Submit | Submits the data entered to the database. |
| 11 | Reset | Resets the input fields in the screen. |
| 12 | Medical History | Future scope – Displays the appointment history of the patient |
| 13 | Disease Info | Future scope – Displays the details of the diseases. |
| 14 | Logout | Log out the doctor from the current session. |
| 15 | Visit Details screen | Screen Name |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1 | visit | ResultSetFetch.selapp  ResultSetFetch.selusr  ResultSetFetch.updateApp  ResultSetFetch.updateApp1 | - | appointment |

### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred for** |
| 1 | DH\_USERS\_1 | To retrieve the name of the patient for that appointment. |
| 2 | DH\_APPOINTMENT | To display the appointment details, update the details entered by the doctor. |

### Exception Handling

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1 | Next visit date | Next visit date is not from tomorrow and 30 days from tomorrow. |  |

# Manager

Dr. Help is developed in a way that Higher Management can view the performance metrics which can be used for the application development, provide feedback and create an ad-hoc request for customized reports via Raise A Request option. Dr. Help provides 5 customized reports which are available on the go also a feedback section to provide comments and suggestions.

Following are the functionalities of a Management which is available in this application. The manager can Login or Register for accessing Dr. Help, Access reports such as Doctor Report, User Report, Hospital Report, Hospital Venture Report, Hospital Category list. Create a customized report, add a new record, drop a table, grant privileges, revoke privileges by creating an ad-hoc request from Raise A Request option. Management can also give suggestions and their comments to improve the application by clicking Provide Feedback To Developers.

Below are the Login ID and Password for logging in to Management Report Section and access reports. Once a successful login is made the login time is updated for the individual. Any Invalid Email-Id and Password will display error message explained in the [Login Section](#_Login_Screen:)

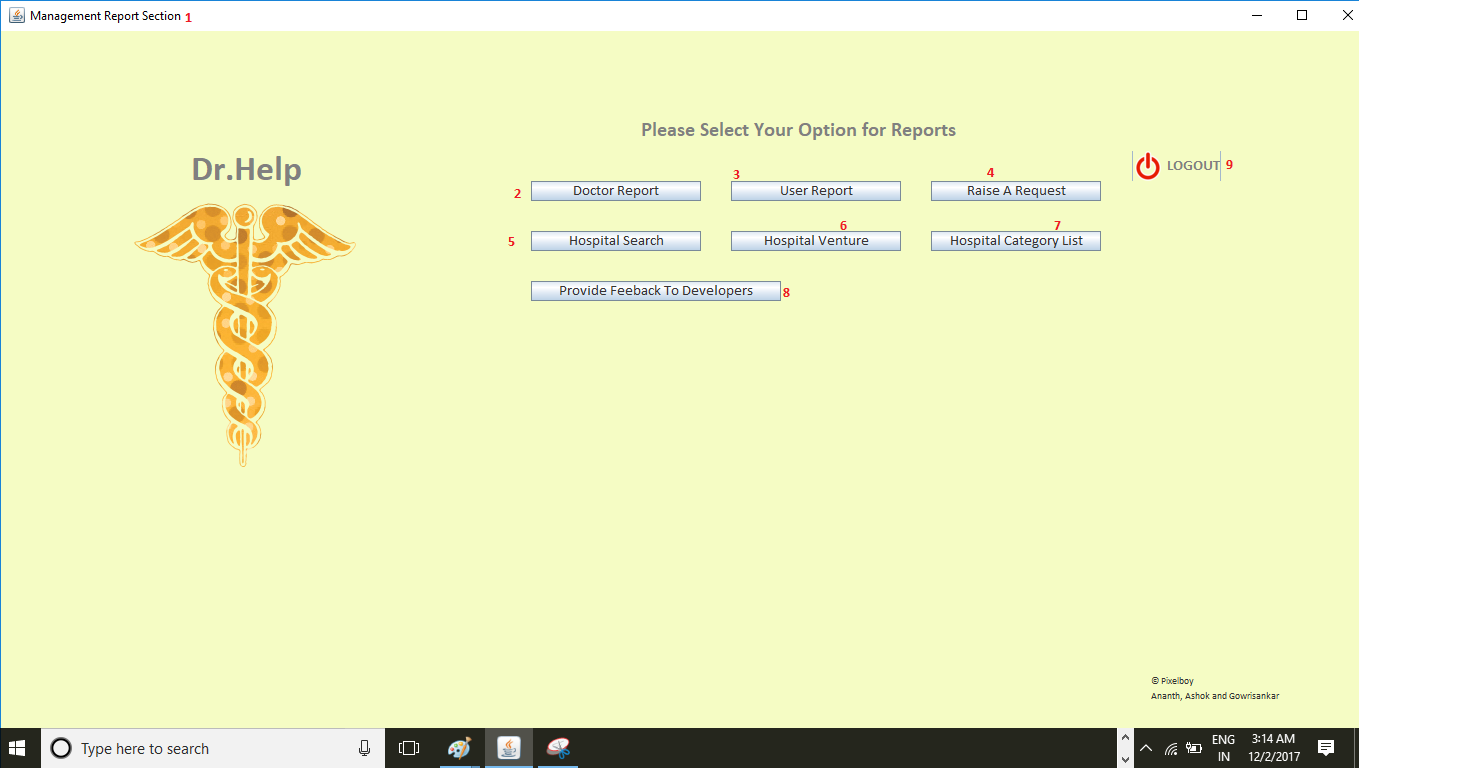
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No** | **Email-ID** | **Domain** | **Password** | **Category** | **Description** |
| 1 | cananthc | @yahoo.com | [Ananth@1234](mailto:Ananth@1234) | MANAGEMENT | Successful Login |

After a successful login below screen will be displayed and the reports are readily available to view. All these results are fetched from the database and populated in a new JFrame.

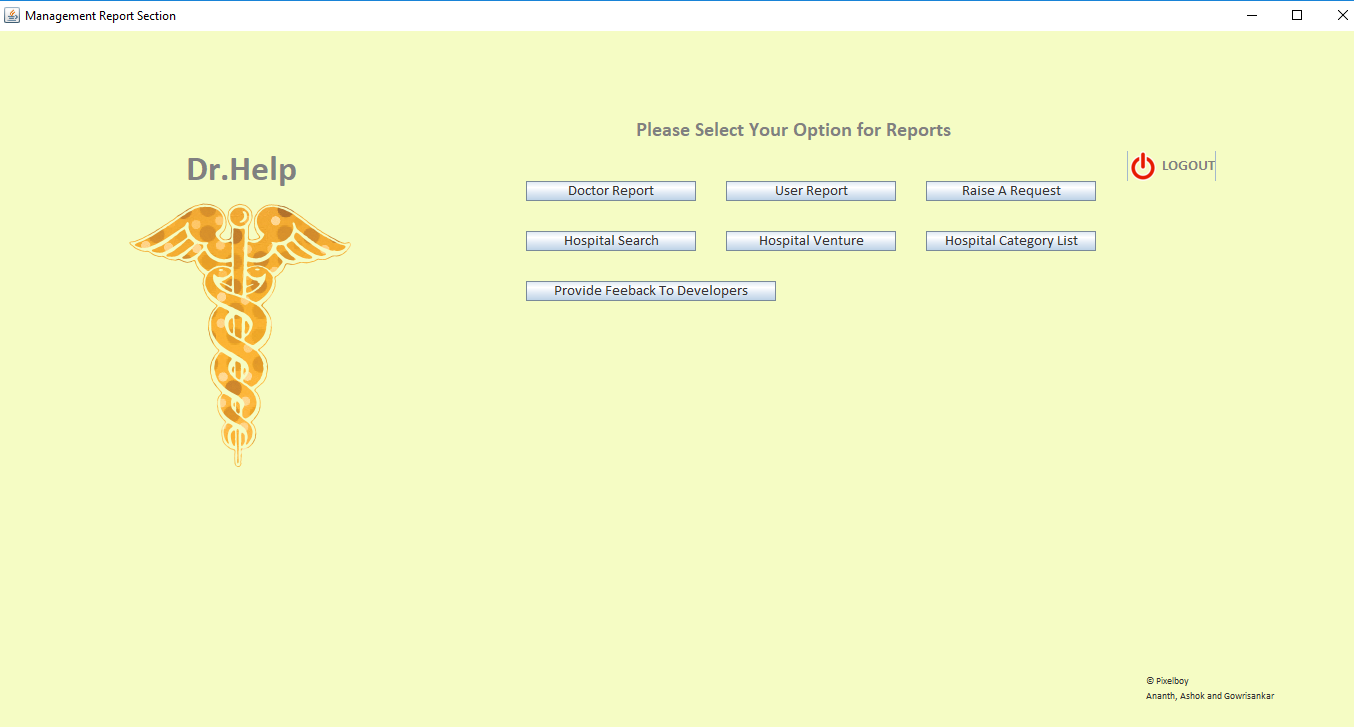
Exceptional handling is also implemented if the Start Date and End Date is not filled. A pop-up message comes stating Start Date or End Date cannot be empty. Once all the necessary fields have filled the reports are displayed in a new JFrame.

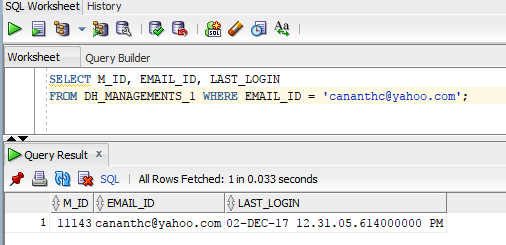
Similarly, all other reports work the same way and exception handling is implemented to all the fields and these exceptions are stored in the database which can be later used to improve the application. Once the Logout option is clicked the application routes back to the Login page and updates the Logout timings for the respective Manager.

### Screenshot

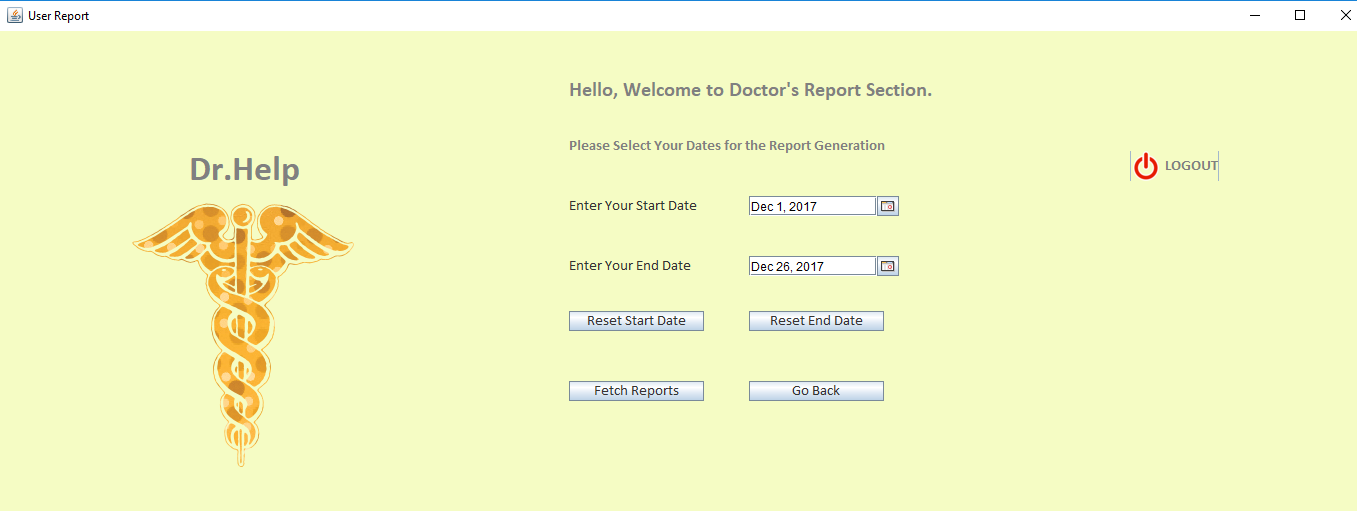


### After a successful Login

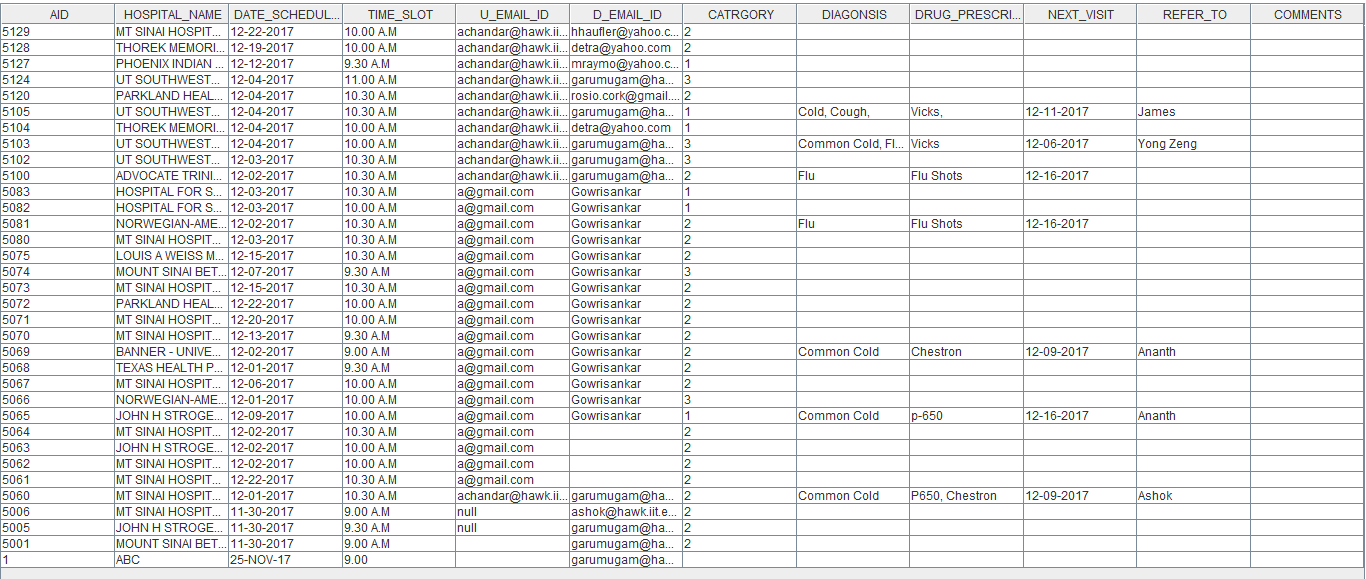




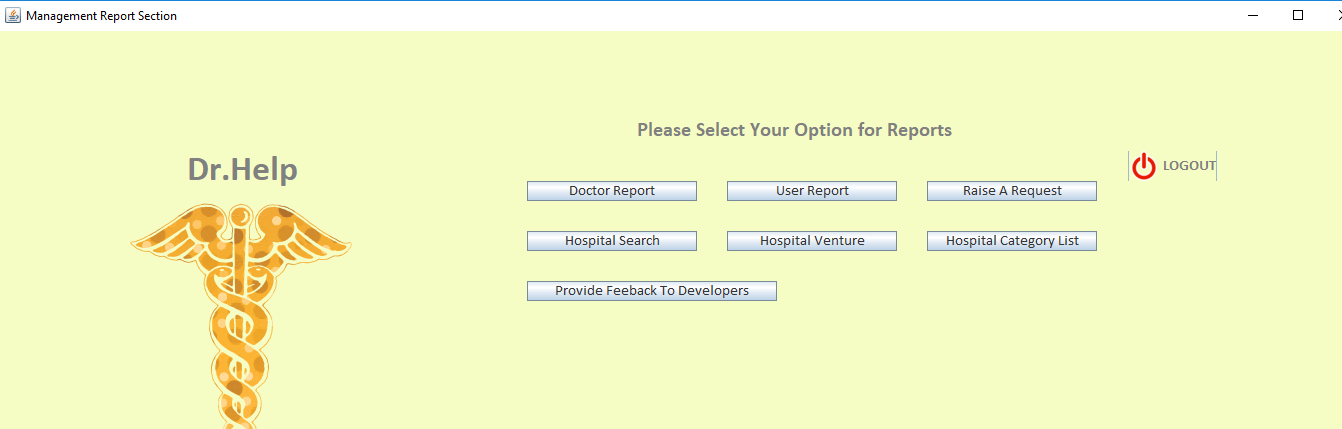
### Enter Start Date and End Date to fetch reports

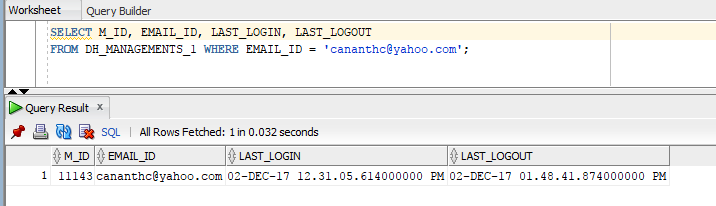


### Populate list of appointments for the users



### Update Logout time when Logout button is clicked





### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Management Report Section | JFrame Name |
| 2 | Doctor Report | View list of Doctors who joined over a period of time |
| 3 | User Report | View User appointments for over a period of time |
| 4 | Raise A Request | Create ad-hoc request, or add, delete, update, insert any new records into the database |
| 5 | Hospital Search | View Hospitals names by entering location or zip code |
| 6 | Hospital Venture | View list of hospitals which was agreed to be a part of Dr. Help |
| 7 | Hospital Category List | View list of hospitals category wise |
| 8 | Provide Feedback | Provide comments and suggestions to Developer |
| 9 | Logout | Takes back to Login page and updates Logout time |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | Login\_All | ResultSetFetch.getLoginDetails | - |  |
| 2. | Login\_All | ResultSetFetch.getEmailCount |  |  |
| 3. | Login\_All | ResultSetFetch.getAccountStatus |  |  |
| 4. | Login\_All | ResultSetFetch.updateLoginTime |  |  |
| 5. | Login\_All | ResultSetFetch.get\_SecurePassword |  |  |
| 6. | newManager | InsertIntoDB.insertNewManager |  | Login\_All |
| 7. | Management\_Report |  |  | Login\_All |
| 8. | ManagementReport  (Doctor Report) | ResultSetFetch.getDocList | Report\_View |  |
| 9. | ManagementReport  (User Report) | ResultSetFetch. getUserAppointment | User\_Report |  |
| 10. | Admin\_Help  (Raise A Request) | InsertIntoDB.insertRequestTick |  |  |
| 11. | NearByMedicalShop  (Hospital Search) | ResultSetFetch.getMedRecordsByLoc  ResultSetFetch.getMedRecordsBy.  getMedRecordsByZip | MedViewByLoc  MedView |  |
| 12. | Hospital\_Venture | ResultSetFetch.getHospVenture | HospVenture\_View |  |
| 13. | Hosp\_Category | ResultSetFetch.getHospCategory | Hosp\_Category |  |
| 14. | Management\_Report  (feedbackReport) | InsertInToDB.feedbackComments |  |  |

### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | DH\_MANAGEMENTS\_1 | Registered higher MANAGEMENT details |
| 2. | DH\_DOC\_LIST\_1 | Doctor Report |
| 3. | DH\_APPOINTMENT | User Report |
| 4. | DH\_ADMIN\_TICKETS | Raise ad-hoc request |
| 5. | DH\_HOSPLIST\_1 | Hospital Search |
| 6. | DH\_HOSP\_VENTURE | Hospital Venture |
| 7. | DH\_HOSPLIST\_1 | Hospital Category |
| 8. | DH\_FEEDBACK | Feedback Report |
| 8. | DH\_EMPTY\_FIELDS | If any fields are filled empty |
| 9. | DH\_SQL\_ERRORS\_1 | SQL Error Identifier |
| 9. | DH\_LOGIN\_ERROR\_1 | Login Validation failure |

### Exception Handling

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1. | Email\_Id | Empty | DH\_EMPTY\_FIELDS\_1 |
| 2. | Dropdown for Domain | Empty | DH\_EMPTY\_FIELDS\_1 |
| 3. | Password | Empty | DH\_EMPTY\_FIELDS\_1 |
| 4. | CATEGORY | Empty | DH\_EMPTY\_FIELDS\_1 |
| 5. | Incorrect Password | Password Mismatch | DH\_LOGIN\_ERROR\_1 |
| 6. | Incorrect Email-Id | No User Found | DH\_LOGIN\_ERROR\_1 |
| 7. | Start Date | Empty | DH\_EMPTY\_FIELDS\_1 |
| 8. | End Date | Empty | DH\_EMPTY\_FIELDS\_1 |
| 9. | Name | Empty | DH\_EMPTY\_FIELDS\_1 |
| 10. | Contact Number | Empty | DH\_EMPTY\_FIELDS\_1 |
| 11. | Issue Type | Empty | DH\_EMPTY\_FIELDS\_1 |
| 12. | Issue Description | Empty | DH\_EMPTY\_FIELDS\_1 |
| 13. | Priority Level | Empty | DH\_EMPTY\_FIELDS\_1 |

# Admin

Admin plays the role of IT Ops in Dr. Help. Admins functionalities are described as to manage the overall application, handle issues and concerns raised in a form of a ticket. Admins will also be able to perform DBA activities such as Data definition language (DDL), Data manipulation language (DML), Data control language (DCL).

Below are the operations Admin can perform in Dr.Help. Admin can Login to access the Admin portal or can signup with Dr.Help and requests for an Admin account. After a successful Login admin can perform activities such as View list of Open/Closed/In-Progress tickets depending upon the priority for a period of time.

Dr. Help also helps admin to perform DBA activities such as Data definition language (DDL) operations such as (SELECT, INSERT, UPDATE), Data manipulation language (DML) such as (CREATE, ALTER, DROP), Data control language(DCL) such as (GRANT, REVOKE). Dr. Help also provides basic syntax for the above DBA operations to help the Admins.

To perform the above activities there are certain validations that should be satisfied to perform DBA activities.

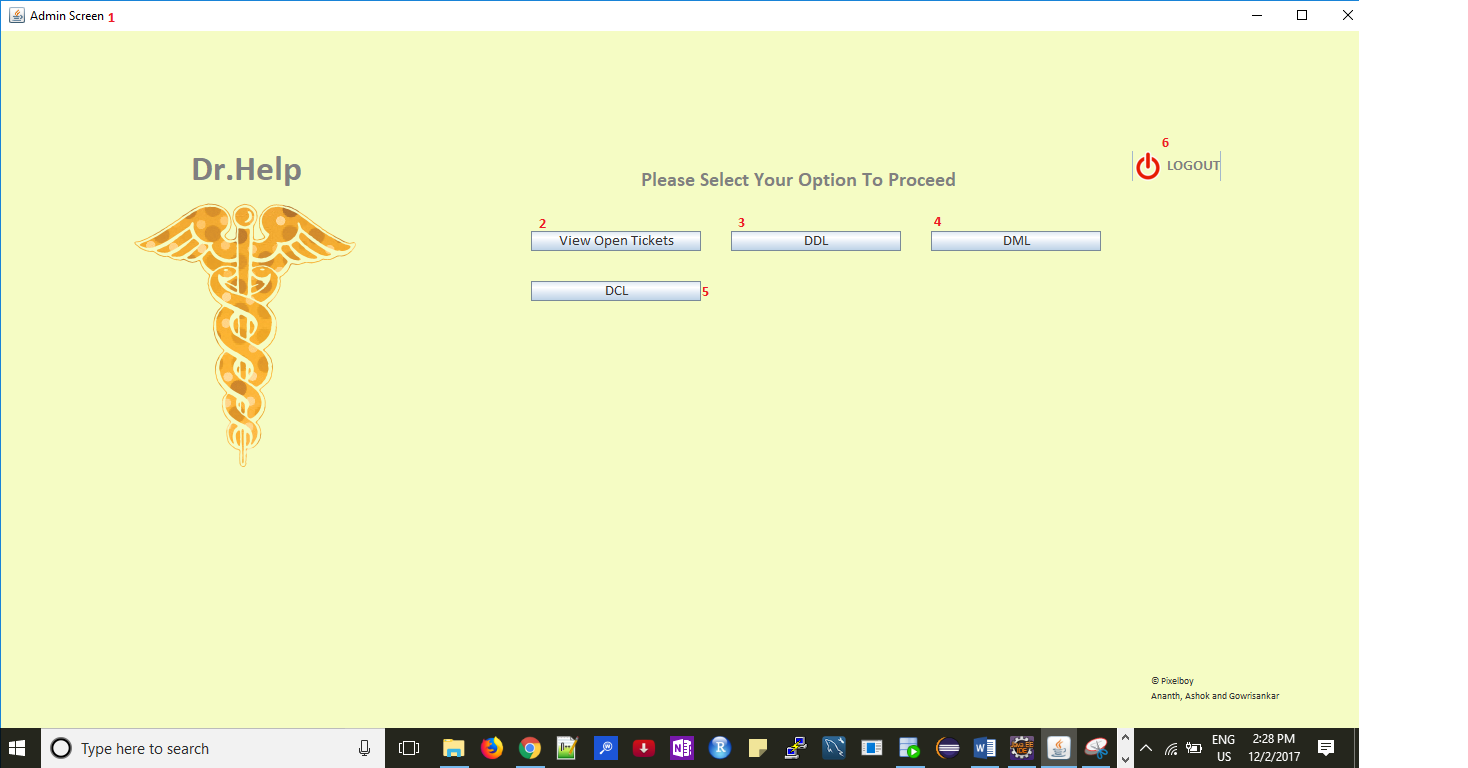
Must have a valid Ticket Number registered for the operation else a pop us saying Invalid Ticket Reference Number or Ticket is not in Open Status will be displayed. Syntax will be auto-populated depending on the DBA activities. Admin must enter a valid syntax before executing the operations. All the entered syntax will be validated from the Validate SQL Query Button before executing against the database.

Special care has been taken for DROP and DELETE statements as it will result in data loss. Hence when an Admin performs DROP or DELETE a confirmation dialogue will be popped up for confirmation.

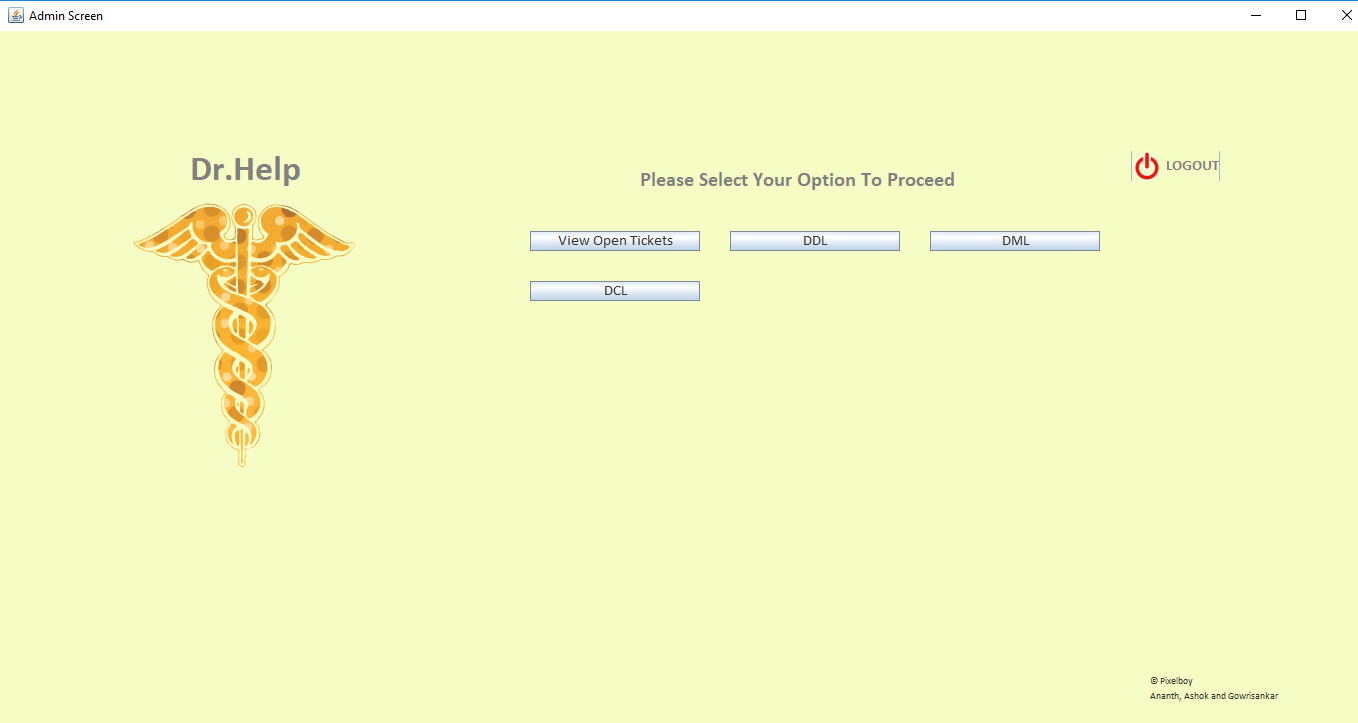
Below are the Login ID and Password for logging in to Admin Portal Section. Once a successful login is made the login time is updated for the individual. Any Invalid Email-Id and Password will display error message explained in the [Login Section](#_Login_Screen:).

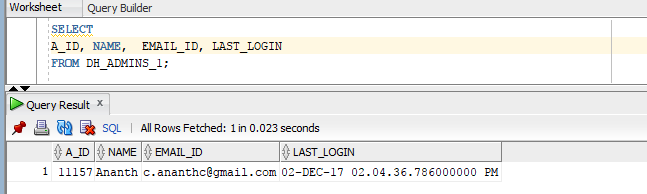
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No** | **Email-ID** | **Domain** | **Password** | **Category** | **Description** |
| 1 | c.ananthc | @gmail.com | [Ananth@1234](mailto:Ananth@1234) | ADMIN | Successful Login |

## Screenshot

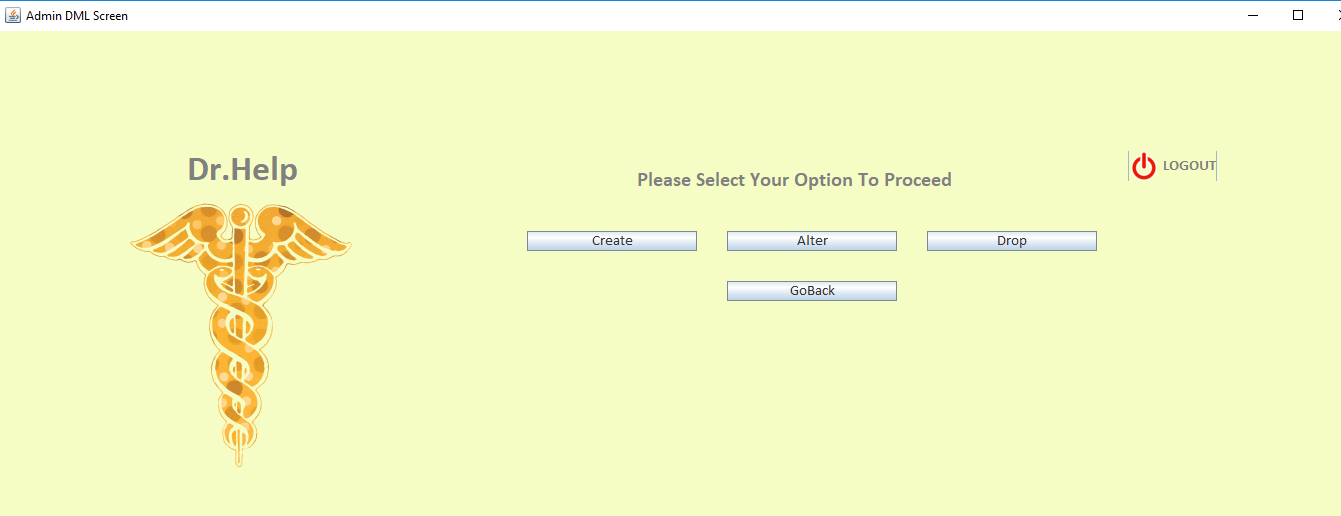


### After a successful Login

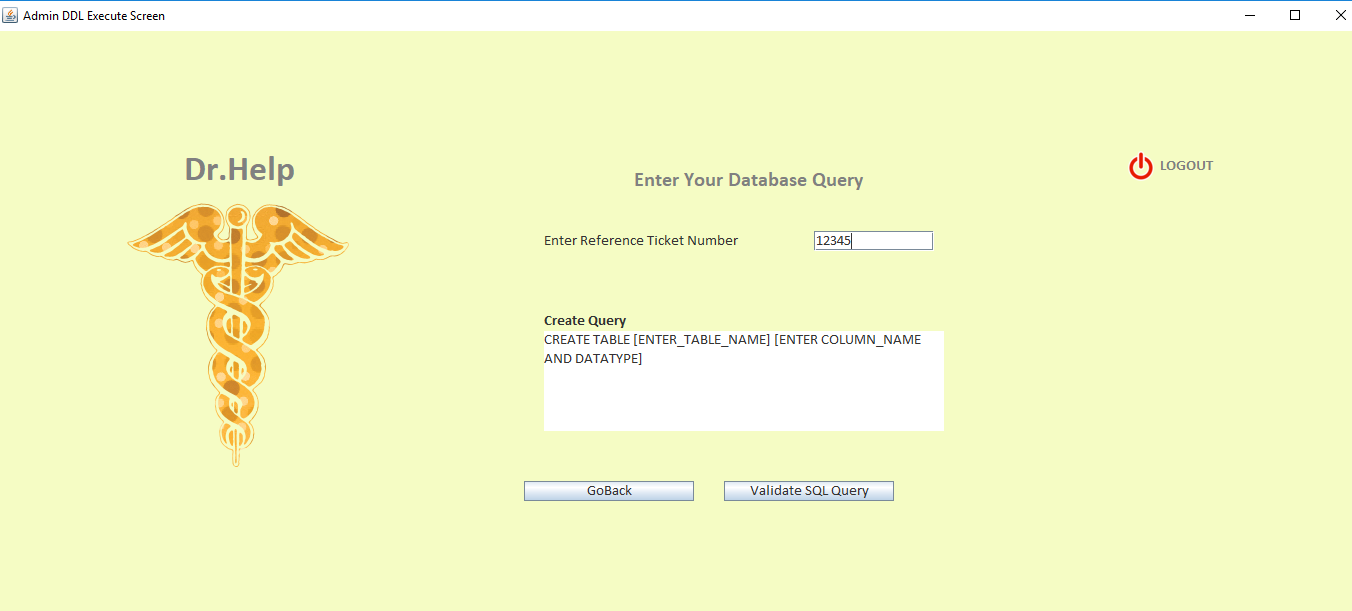




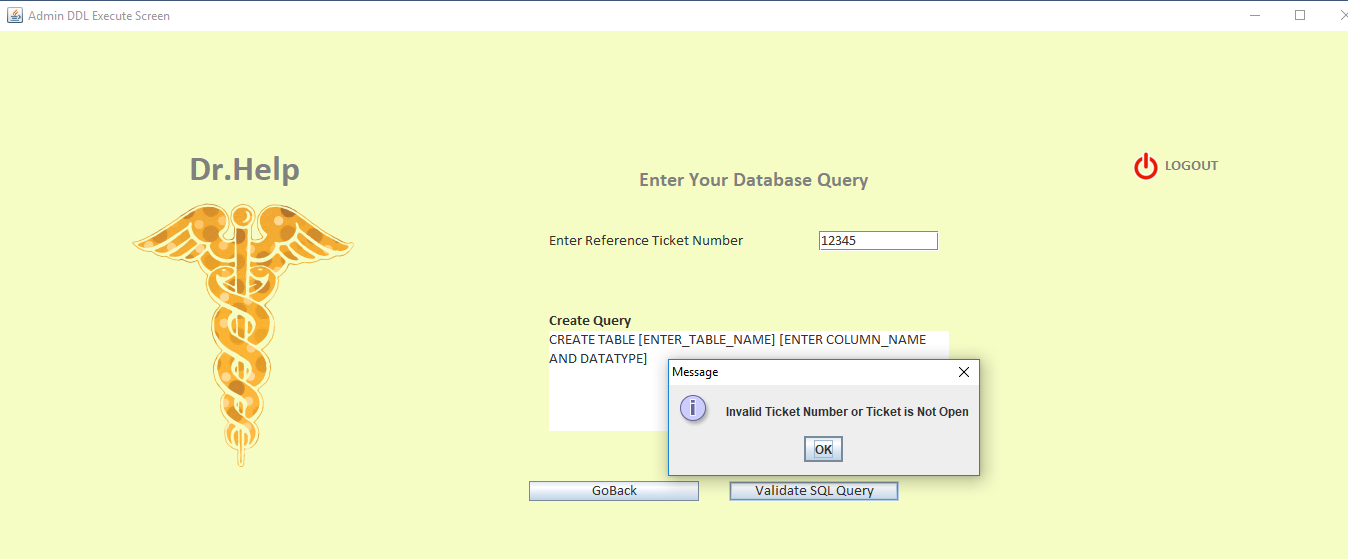
### If Admin wants to Perform a DML Activity



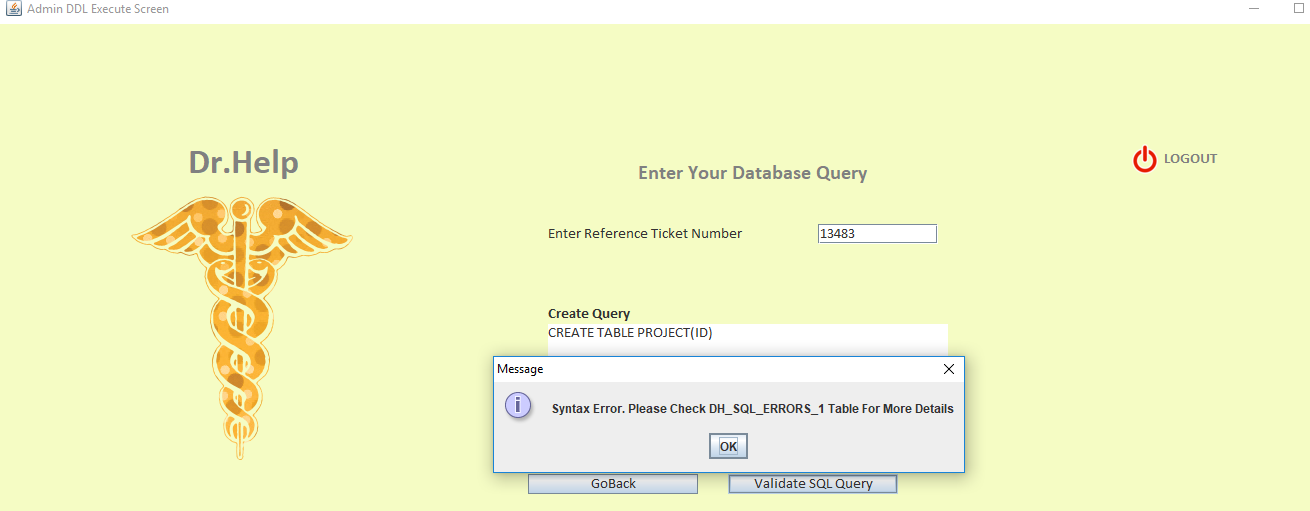
### Basic Syntax for every DBA Activity populates automatically



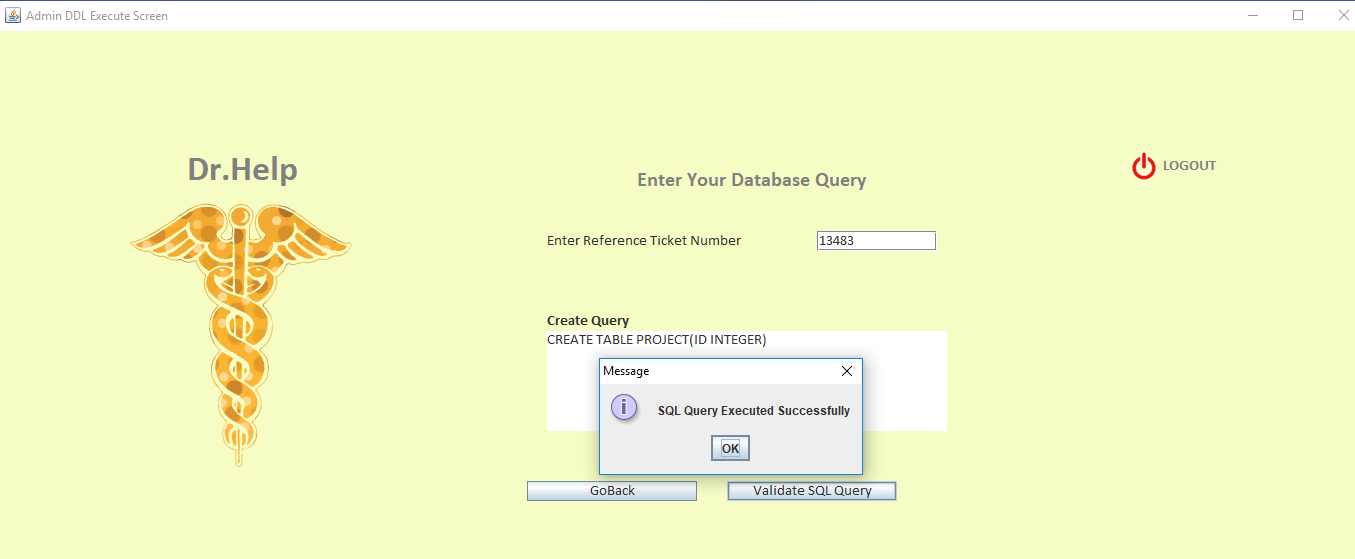
### Invalid Ticket Number or Ticket is not in Open Status



### SQL Syntax Error if Query is wrong



### If SQL Query is valid



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Admin Screen | JFrame Name |
| 2 | View Open Tickets | View list of tickets which is Open/Closed/In-Progress depending upon priority |
| 3 | DDL | Perform DDL Operations |
| 4 | DML | Perform DML Operations |
| 5 | DCL | Perform DCL Operations |
| 6 | Logout | Takes back to Login page and updates Logout time |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | Login\_All | ResultSetFetch.getLoginDetails | - |  |
| 2. | Login\_All | ResultSetFetch.getEmailCount |  |  |
| 3. | Login\_All | ResultSetFetch.getAccountStatus |  |  |
| 4. | Login\_All | ResultSetFetch.updateLoginTime |  |  |
| 5. | Login\_All | ResultSetFetch.get\_SecurePassword |  |  |
| 6. | newManager | InsertIntoDB.insertNewManager |  | Login\_All |
| 7. | Admin\_Screen |  |  | Login\_All |
| 8. | Admin\_Ticket\_status |  | Admin\_Tickets\_View |  |
| 9. | Admin\_DDL\_Execute | ResultSetFetch.validateSQLQueryDDL  ResultSetFetch.validateSQLQuerySelect  ResultSetFetch.updateClosedTicketTime | Admin\_Select\_DDL |  |
| 10. | Admin\_DML\_Execute | ResultSetFetch.validateSQLQueryDML  ResultSetFetch.updateClosedTicketTime |  |  |
| 11. | Admin\_DCL\_Execute | ResultSetFetch.validateSQLQueryDCL  ResultSetFetch.updateClosedTicketTime |  |  |

### Database Reference:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | DH\_ADMINS\_1 | Registered Admin details |
| 2. | DH\_ADMIN\_TICKETS | View Ad-hoc request tickets |
| 3. | DH\_LOGIN\_ERROR\_1 | Login failure validations |
| 4. | DH\_EMPTY\_EMPTY\_FIELDS | If any fields are filled empty |
| 5. | DH\_SQL\_ERRORS\_1 | Any SQL Error while performing DBA activities |

### Exception Handling

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1. | Email\_Id | Empty | DH\_EMPTY\_FIELDS\_1 |
| 2. | Dropdown for Domain | Empty | DH\_EMPTY\_FIELDS\_1 |
| 3. | Password | Empty | DH\_EMPTY\_FIELDS\_1 |
| 4. | CATEGORY | Empty | DH\_EMPTY\_FIELDS\_1 |
| 5. | Incorrect Password | Password Mismatch | DH\_LOGIN\_ERROR\_1 |
| 6. | Incorrect Email-Id | No User Found | DH\_LOGIN\_ERROR\_1 |
| 7. | SQL Syntax | Syntax Error | DH\_SQL\_ERRORS\_1 |
| 8. | SQL Syntax | Empty | DH\_EMPTY\_FIELDS |
| 9. | Ticket Reference Number | Invalid or Ticket Reference number not Open | DH\_EMPTY\_FIELDS\_1 |
| 10. | Name | Empty | DH\_EMPTY\_FIELDS\_1 |
| 11. | Contact Number | Empty | DH\_EMPTY\_FIELDS\_1 |
| 12. | Issue Type | Empty | DH\_EMPTY\_FIELDS\_1 |
| 13. | Issue Description | Empty | DH\_EMPTY\_FIELDS\_1 |
| 14. | Priority Level | Empty | DH\_EMPTY\_FIELDS\_1 |

# Raise A Request

Dr. Help has a ticket request tool which can be used to raise a request and Admin will take over the ticket to resolve the issue.

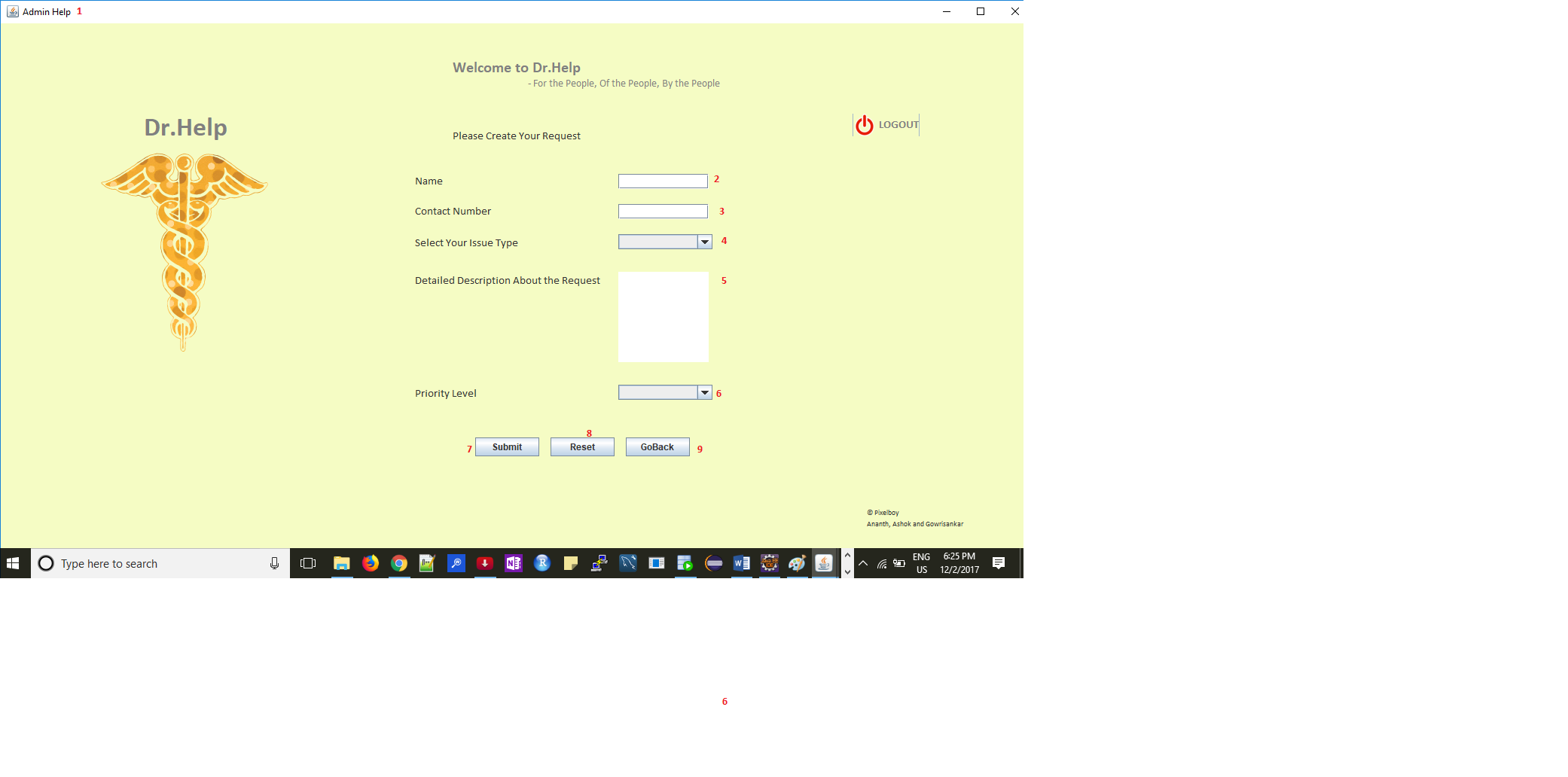
Raise A Request is an online form where you can use this to Create any reports, tables etc, delete any records, reports, drop tables, grant access, remove access, create customized reports, unlock Account.

An automated TicketNumber will be generated for every new ticket and will be updated in the database when it is raised. There is an option called priority which should be selected to action your request as per urgency. With this priority, the SLA Days, Hours remaining will be calculated automatically as per the industry standards. All these are populated into DH\_ADMIN\_TICKETS table

### Standard Template Followed

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Priority** | **SLA\_DAYS** | **Hours Remaining** |
| 1 | 1 | 1 | 2 |
| 2 | 2 | 1 | 4 |
| 3 | 2 | 2 | 8 |
| 4 | 3 | 2 | 16 |

## Screenshot



### Label Description:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Name** | **Description** |
| 1 | Admin\_Help | JFrame Name |
| 2 | Name | Name of the Person requesting |
| 3 | Contact Number | Phone Number of the person requesting |
| 4 | Select Issue Type | Select Your Issue Category type |
| 5 | Detailed Description Of the Request | Provide details about the request |
| 6 | Priority Level | Select your Priority of the Request |

### Java Reference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Class Name** | **Model Dependency** | **View Dependency** | **Controller Dependency** |
| 1. | Admin\_Help | InsertInToDB.insertRequestTick | - | Login\_All |

### Database References

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | DH\_ADMIN\_TICKETS | Ticket details are stored |

### Exception handling

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Reference** | **Error If** | **Reference Table** |
| 1. | Name | Empty | DH\_EMPTY\_FIELDS\_1 |
| 2. | Contact Number | Empty | DH\_EMPTY\_FIELDS\_1 |
| 3. | Issue Type | Empty | DH\_EMPTY\_FIELDS\_1 |
| 4. | Issue Details | Empty | DH\_EMPTY\_FIELDS\_1 |
| 5. | Priority | Empty | DH\_LOGIN\_ERROR\_1 |

# Exception Handling

One of the most important aspects is error trapping and error handling which is incorporated in Dr. Help. Maximum Exception Handling has been covered and also stored in the database depending upon the classification of errors.

In this application exception handling is covered for each field which will be filled or left blank. Starting from Login page to Logout page every interaction made with this application is recorded in the database which will be used for further development of Dr. Help

### Login Error Handler:

If all the required fields are not filled or if there no Email-Id for the registered category, bad username and password all these are captured under one table with actions that are performed. Since these are related to Login issues, these are captured in a table called DH\_LOGIN\_ERROR\_1. Each column located under this table has its own explanation.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| UNIQUE\_ID | Generated unique ID whenever an error occurs |
| DESCRIPTION | Describes what type of Login Error |
| CATEGORY | From which Category Login |
| ERROR\_DATE | Date of Error |

### Empty Field Error Handler:

If a field is not filled and left empty and the USER/DOCTOR/MANAGEMENT/ADMIN want to go to the next screen it will not be possible because it may be a mandatory field. Then these are captured in a table called DH\_EMPTY\_FIELDS\_1.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| UNIQUE\_ID | Generated unique ID whenever an error occurs |
| DESCRIPTION | Describes what type of Login Error |
| SCREEN\_NAME | From which Screen error came |
| ERROR\_DATE | Date of Error |

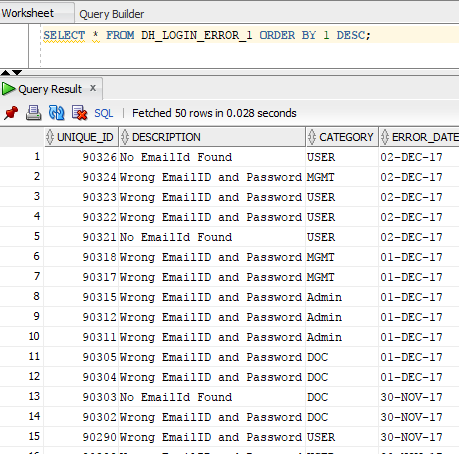
### SQL Error Handler

When Admin performs any DBA activities and if any syntax is wrong these are captured along with the actual error and admin will be informed to check the table for more details. Table name for SQL Error handler is DH\_SQL\_ERRORS where the error is captured along with SQL Error Code.

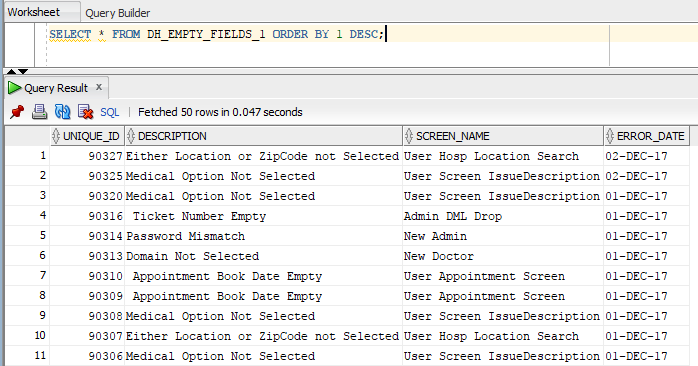
|  |  |
| --- | --- |
| **Column Name** | **Description** |
| UNIQUE\_ID | Generated unique ID whenever an error occurs |
| DESCRIPTION | Describes what type of Login Error |
| SYNTAX | What is the Syntax entered by Admin |
| ERROR\_FROM | Which Screen this SQL ERROR with SQL Error Code |
| ERROR\_DATE | Date of Error |

## Screenshot

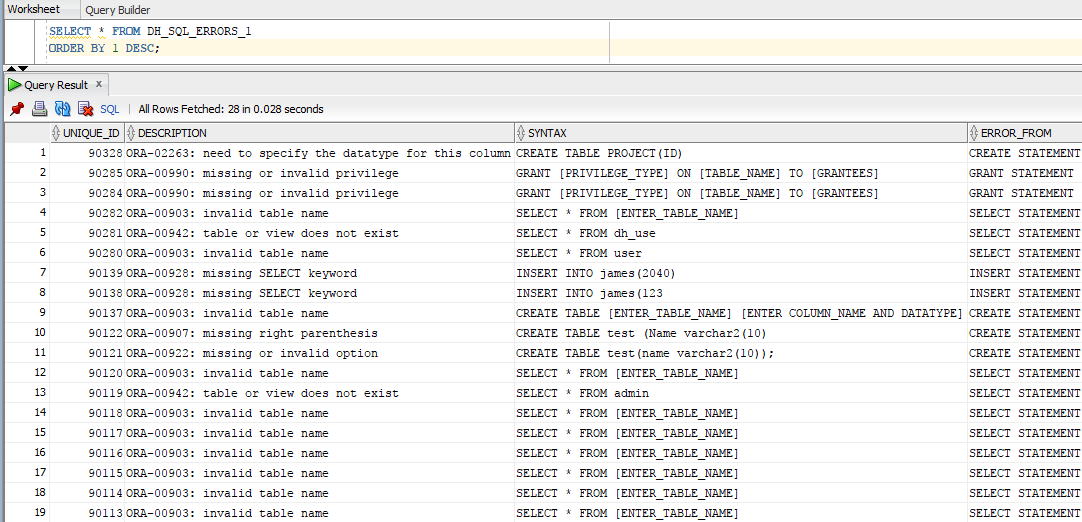
### Login Error Handler



### Empty Fields Error Handler



### SQL Error Handler



### Java Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Package Name** | **Class Name** | **Description** |
| 1 | ErrorHandler | EmptyFields\_Error\_Handler | If Empty Fields are filled |
| 2 | ErrorHandler | Login\_Error | If Login Validation fails |
| 3 | Models | ResultSetFetch.validateSQLQuerySelect | Validates Select Statement |
| 4 | Models | ResultSetFetch.validateSQLQueryDDL | Validates DDL Commands |
| 5 | Models | ResultSetFetch.validateSQLQueryDML | Validates DML Commands |
| 6 | Models | ResultSetFetch.validateSQLQueryDCL | Validates DCL Commands |
| 7 | Models | InsertInToDB.SQLSyntaxErrorMessage | Inserts SQL Error Message into the database |

### Database Reference

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table Name** | **Referred For** |
| 1. | DH\_LOGIN\_ERROR\_1 | Login Validation failure |
| 2. | DH\_EMPTY\_FIELDS\_1 | If any field is left empty |
| 3. | DH\_SQL\_ERRORS\_1 | SQL Exceptions |

# References

## Java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. NO** | **Package Name** | **Package Description** | **Class Name** | **Class Description** |
| **1** | **Dr. Help** | **Controller Package** | **Login and Registration** | |
| Login\_All.java | Main Program |
| Welcome\_Screen.java | New Registration Type |
| UserType\_Selection.java | Choose New Registration |
| newUser.java | New User Registration |
| newDoctor.java | New Doctor Registration |
| newManager.java | New Manager Registration |
| newAdmin.java | New Admin Registration |
|  |  |
| **User Section** | |
| IssueDescription.java | Choose Issue Type |
| UserAppConfirmation.java | Choose location |
| Hosp\_List.java | Choose Hospital and Time Slot |
| Appointment\_summary.java | Summary Screen of User |
|  |  |
| **Doctor Section** | |
| visit.java | Check Appointment for Date |
| appointment.java | Appointment summary |
|  |  |
| **Management Section** | |
| Management\_Report.java | Management Screen Main Menu |
| Hosp\_Category.java | Hospital Report Category wise |
| Hosp\_Venture.java | Hospital Venture Report |
| NearByMedicalShop.java | Search Hospital by location or Zip |
|  |  |
| **Admin Section** | |
| Admin\_Screen.java | Admin Main Menu |
| Admin\_DCL\_Execute.java | Admin DCL Exectuion |
| Admin\_DCL.java | Admin DCL Menu |
| Admin\_DML\_Execute.java | Admin DML Exectuion |
| Admin\_DML.java | Admin DML Menu |
| Admin\_DDL\_Execute.java | Admin DDL Exectuion |
| Admin\_DDL.java | Admin DDL Menu |
| Admin\_Help.java | Raise A Request |
| AdminTickets\_Status.java | Check tickets |
|  |  |  |  |  |
| **2** | **ErrorHandler** | **Error Handler Pacakge** | EmptyFields\_Error\_Handler.java | Captures Empty Fields errors |
| Login\_Error.java | Captures Login validation errors |
|  |  |  |  |  |
| **3** | **Models** | **CRUD Operations** | Connector.java | Database Connection |
| InsertInToDB.java | Insert/Update/Delete operations |
| ResultSetFetch.java | Results from database |
|  |  |  |  |  |
| **4** | **View** | **Jtable function** | Admin\_Select\_DDL.java | Select query populates in a Jtable |
| Admin\_Tickets\_View.java | Admin tickets populates in a Jtable |
| Hosp\_Category.java | Category wise hospital populates in a Jtable |
| HospVenture\_View.java | Hospital are populated in a Jtable based on Venture date |
| MedView.java | Populates Hospital based on Location or ZipCode |
| Report\_View.java | User appointment report populates in a Jtable |
| User\_Report.java | Doctor report populates in a Jtable |

## Database

### List of tables

|  |  |  |
| --- | --- | --- |
| **Scheme Name** | **Table Name** | **Description** |
| ORA\_AAG | DH\_USERS\_1 | User Details |
| ORA\_AAG | DH\_DOCTORS\_1 | Doctor Details |
| ORA\_AAG | DH\_MANAGEMENTS\_1 | Manager Details |
| ORA\_AAG | DH\_ADMINS\_1 | Admin Details |
| ORA\_AAG | DH\_APPOINTMENT | User Appointment Details |
| ORA\_AAG | DH\_HOSPLIST\_1 | Hospital List |
| ORA\_AAG | DH\_HOSP\_VENTURE | Hospital Venture List |
| ORA\_AAG | DH\_DOC\_LIST\_1 | Doctor List |
| ORA\_AAG | DH\_ADMIN\_TICKETS | Admin Ticket Details |
| ORA\_AAG | DH\_FEEDBACK | Management Feedback |
| ORA\_AAG | DH\_EMAIL\_DOMAIN | Email Domain for Login |
| ORA\_AAG | DH\_EMPTY\_FIELDS\_1 | Empty Field Error Details |
| ORA\_AAG | DH\_LOGIN\_ERROR\_1 | Login Error Details |
| ORA\_AAG | DH\_SQL\_ERRORS\_1 | SQL Error Details |

### Column Details

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_ADMIN\_TICKETS** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| TICKET\_NUMBER | NUMBER(38,0) | No | Ticket ID |
| RAISED\_BY | VARCHAR2(50 BYTE) | Yes | Raise By |
| CONTACT\_NUMBER | VARCHAR2(20 BYTE) | Yes | Contact Number |
| ISSUE\_CATEGORY | VARCHAR2(20 BYTE) | Yes | Ticket Category |
| ISSUE\_DESCRIPTION | VARCHAR2(2000 BYTE) | Yes | Details about the ticket |
| PRIORITY\_LEVEL | NUMBER(38,0) | Yes | Priority Level |
| DATE\_SUBMITTED | DATE | Yes | Date submitted |
| TIME\_SUBMITTED | TIMESTAMP(6) | Yes | Time when created |
| TIME\_CLOSED | TIMESTAMP(6) | Yes | Resolved time |
| SLA\_DAYS | NUMBER(38,0) | Yes | Days remaining for this ticket |
| HOURS\_REMAINING | NUMBER(10,2) | Yes | Hours remaining for this ticket |
| STATUS | VARCHAR2(10 BYTE) | Yes | Status of the ticket |
| ASSIGNED\_TO | VARCHAR2(50 BYTE) | Yes | Assigned to |
| COMMENTS | VARCHAR2(100 BYTE) | Yes | Comments |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_ADMINS\_1** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| A\_ID | NUMBER(38,0) | Yes | Admin ID |
| NAME | VARCHAR2(50 BYTE) | Yes | Admin Name |
| PASSWORD | VARCHAR2(64 BYTE) | Yes | Admin Password Encrypted |
| DOB | VARCHAR2(8 BYTE) | Yes | Admin Date of Birth |
| AGE | NUMBER(38,0) | Yes | Admin Age |
| EMAIL\_ID | VARCHAR2(50 BYTE) | No | Admin Email-Id |
| GENDER | VARCHAR2(6 BYTE) | Yes | Admin Gender |
| DATE\_OF\_REGISTRATION | DATE | Yes | Admin Registered Date |
| LAST\_LOGIN | TIMESTAMP(6) | Yes | Admin Last Login |
| LAST\_LOGOUT | TIMESTAMP(6) | Yes | Admin Last Logout |
| ACCOUNT\_LOCKED | VARCHAR2(1 BYTE) | Yes | Account Status |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_USERS\_1** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| U\_ID | NUMBER(38,0) | Yes | User Id |
| NAME | VARCHAR2(50 BYTE) | Yes | User Name |
| PASSWORD | VARCHAR2(64 BYTE) | Yes | User Password Encrypted |
| DOB | VARCHAR2(10 BYTE) | Yes | User Date of Birth |
| AGE | NUMBER(38,0) | Yes | User Age |
| GENDER | VARCHAR2(6 BYTE) | Yes | User Gender |
| EMAIL\_ID | VARCHAR2(50 BYTE) | No | User Id |
| MOBILE | NUMBER(38,0) | Yes | User Mobile Number |
| DATE\_OF\_REGISTRATION | DATE | Yes | User Registered Date |
| LAST\_LOGIN | TIMESTAMP(6) | Yes | User Last Login |
| LAST\_LOGOUT | TIMESTAMP(6) | Yes | User Last Logout |
| ACCOUNT\_LOCKED | VARCHAR2(1 BYTE) | Yes | Account Status |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_DOCTORS\_1** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| D\_ID | NUMBER(38,0) | Yes | Doctor ID |
| NAME | VARCHAR2(50 BYTE) | Yes | Doctor Name |
| PASSWORD | VARCHAR2(64 BYTE) | Yes | Doctor Password Encrypted |
| DOB | VARCHAR2(8 BYTE) | Yes | Doctor Date of Birth |
| AGE | NUMBER(38,0) | Yes | Doctor Age |
| GENDER | VARCHAR2(6 BYTE) | Yes | Doctor Email-Id |
| EMAIL\_ID | VARCHAR2(50 BYTE) | No | Doctor Gender |
| MOBILE | NUMBER(38,0) | Yes | Doctor Mobile Number |
| CATEGORY | NUMBER(38,0) | Yes | Doctor Category |
| HOSPITAL\_NAME | VARCHAR2(50 BYTE) | Yes | Doctor's Hospital |
| DATE\_OF\_REGISTRATION | DATE | Yes | Doctor Registered Date |
| LAST\_LOGIN | TIMESTAMP(6) | Yes | Doctor Last Login |
| LAST\_LOGOUT | TIMESTAMP(6) | Yes | Doctor Last Logout |
| ACCOUNT\_LOCKED | VARCHAR2(1 BYTE) | Yes | Account Status |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_DOC\_LIST\_1** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| NAME | VARCHAR2(50 BYTE) | Yes | Doctor's Name |
| ADDRESS | VARCHAR2(50 BYTE) | Yes | Doctors Address |
| CITY | VARCHAR2(50 BYTE) | Yes | Doctor's City |
| COUNTY | VARCHAR2(50 BYTE) | Yes | Doctor's County |
| STATE | VARCHAR2(2 BYTE) | Yes | Doctor's State |
| ZIP | VARCHAR2(5 BYTE) | Yes | Zip Code |
| MOBILE | VARCHAR2(12 BYTE) | Yes | Doctor's Mobile Number |
| EMAIL\_ID | VARCHAR2(50 BYTE) | No | Doctor's Email-Id |
| DATE\_OF\_JOIN | DATE | Yes | Date joined |
| CATEGORY | NUMBER(38,0) | Yes | Category Registered |
| SALARY | NUMBER(10,2) | Yes | Doctor's Salary |
| RATING | NUMBER(38,0) | Yes | Doctor's Performance Rating |
| ACTIVE | VARCHAR2(1 BYTE) | Yes | Active or not |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_APPOINTMENT** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| AID | NUMBER(38,0) | Yes | Appointment ID |
| H\_ID | NUMBER | No | Hospital Id |
| HOSPITAL\_NAME | VARCHAR2(100 BYTE) | No | Hospital Name |
| DATE\_SCHEDULED | VARCHAR2(20 BYTE) | No | Date of Appointment |
| TIME\_SLOT | VARCHAR2(20 BYTE) | No | Time Slot registered |
| U\_EMAIL\_ID | VARCHAR2(100 BYTE) | Yes | User Email-Id |
| D\_EMAIL\_ID | VARCHAR2(100 BYTE) | Yes | Doctor Email-Id |
| CATEGORY | NUMBER(38,0) | Yes | Issue Category |
| DIAGONSIS | VARCHAR2(4000 BYTE) | Yes | Diagonsis prescribed |
| DRUG\_PRESCRIBED | VARCHAR2(4000 BYTE) | Yes | Drugs prescribed |
| NEXT\_VISIT | VARCHAR2(20 BYTE) | Yes | Next appointment |
| REFER\_TO | VARCHAR2(100 BYTE) | Yes | Refer to another doctor |
| COMMENTS | VARCHAR2(4000 BYTE) | Yes | General comments |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_HOSPLIST\_1** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| H\_ID | NUMBER | No | Hospital ID |
| HOSPITAL\_NAME | VARCHAR2(100 BYTE) | Yes | Hospital Name |
| ADDRESS | VARCHAR2(100 BYTE) | Yes | Hospital Address |
| CITY | VARCHAR2(20 BYTE) | Yes | Hospital City |
| STATE | VARCHAR2(2 BYTE) | Yes | Hospital State |
| ZIPCODE | VARCHAR2(5 BYTE) | Yes | Hospital ZipCode |
| PHONE\_NUMBER | NUMBER | Yes | Hospital Phone Number |
| HOSPITAL\_OWNERSHIP | VARCHAR2(50 BYTE) | Yes | Govt or Private Owned |
| MEDICAL\_INSURANCE | NUMBER | Yes | Does it cover Medical Insurance |
| CATEGORY | NUMBER | Yes | Which category hospital |
| DOCTOR\_NAME | VARCHAR2(50 BYTE) | Yes | Doctor Assigned |
| DOCTOR\_EMAIL\_ID | VARCHAR2(50 BYTE) | Yes | Doctor's Email ID |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_HOSP\_VENTURE** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| H\_ID | NUMBER | No | Hospital ID |
| HOSPITAL\_NAME | VARCHAR2(100 BYTE) | Yes | Hospital Name |
| ADDRESS | VARCHAR2(100 BYTE) | Yes | Hospital Address |
| CITY | VARCHAR2(20 BYTE) | Yes | Hospital City |
| STATE | VARCHAR2(2 BYTE) | Yes | Hospital State |
| ZIPCODE | VARCHAR2(5 BYTE) | Yes | Hospital ZipCode |
| PHONE\_NUMBER | NUMBER | Yes | Hospital Phone Number |
| HOSPITAL\_OWNERSHIP | VARCHAR2(50 BYTE) | Yes | Govt or Private Owned |
| MEDICAL\_INSURANCE | NUMBER | Yes | Does it cover Medical Insurance |
| CATEGORY | NUMBER | Yes | Which category hospital |
| DATE\_OF\_VENTURE | DATE | Yes | Date of Venture with Dr. Help |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_MANAGEMENTS\_1** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| M\_ID | NUMBER(38,0) | Yes | Manager ID |
| NAME | VARCHAR2(50 BYTE) | Yes | Manager Name |
| PASSWORD | VARCHAR2(64 BYTE) | Yes | Doctor Password Encrypted |
| GENDER | VARCHAR2(6 BYTE) | Yes | Manager Gender |
| EMAIL\_ID | VARCHAR2(50 BYTE) | No | Manager Email-Id |
| DATE\_OF\_REGISTRATION | DATE | Yes | Manager Registered Date |
| LAST\_LOGIN | TIMESTAMP(6) | Yes | Manager Last Login |
| LAST\_LOGOUT | TIMESTAMP(6) | Yes | Manager Last Logout |
| ACCOUNT\_LOCKED | VARCHAR2(1 BYTE) | Yes | Account Status |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_EMAIL\_DOMAIN** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| DOMAIN\_NAME | VARCHAR2(50 BYTE) | No | Email-ID Domain |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_EMPTY\_FIELDS\_1** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| UNIQUE\_ID | NUMBER(38,0) | Yes | Unique ID |
| DESCRIPTION | VARCHAR2(50 BYTE) | Yes | Empty Field Description |
| SCREEN\_NAME | VARCHAR2(50 BYTE) | Yes | From which screen |
| ERROR\_DATE | DATE | Yes | Error Date |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_FEEDBACK** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| DATE\_PROVIDED | DATE | Yes | Date |
| COMMENTS | VARCHAR2(4000 BYTE) | Yes | Comments |

|  |  |  |  |
| --- | --- | --- | --- |
| **DH\_SQL\_ERRORS\_1** | | | |
| **Column Name** | **Data Type** | **Nullable** | **Description** |
| UNIQUE\_ID | NUMBER(38,0) | Yes | Error ID |
| DESCRIPTION | VARCHAR2(3000 BYTE) | Yes | SQL Error Code with description |
| SYNTAX | VARCHAR2(200 BYTE) | Yes | Syntax Entered |
| ERROR\_FROM | VARCHAR2(100 BYTE) | Yes | Error from which screen |
| ERROR\_DATE | DATE | Yes | Error Date |

# Future Scope

* Implement Machine Learning techniques for the error handlings to improve the application.
* Enable additional email domain for Login.
* Lock your account after three (3) wrong attempts.
* Provide predictive analysis of diagnosis without medical consultation.
* Introduce additional time slots for appointment scheduling.
* Implement charts for Management reports, Users diagnosis, Admin tickets.
* Export Reports in a .csv, pdf format