Qualification		Module Number and Title
HD in Computing and Software Engineering		CSE5011- Mobile Application Development
Student Name & No.		**Assessor**
M.A.Chamath Shyamal		Mr. Roy Ian
Hand out date		**Submission Date**
2021-11-09		2022-01-09
Assessment type	**Duration/Length of Assessment Type**	**Weighting of Assessment**
Coursework	3000 words	100%

Learner declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Marks Awarded			
First assessor			
IV marks			
Agreed grade			
Signature of the assessor		Date	

FEEDBACK FORM
INTERNATIONAL COLLEGE OF BUSINESS & TECHNOLOGY

Module:

Student:

Assessor:

Assignment: Mobile Application for "Saloon SD"

Strong features of your work:

Areas for improvement:

Marks Awarded:

Coursework – Mobile Application for "Saloon SD"– 100 Marks

Learning outcomes covered

1. Explain mobile operating systems, development tools and technologies for the mobile application development
2. Design mobile application solutions
3. Develop mobile application solutions
4. Test mobile application solutions

Scenario

Saloon SD is a one of the famous saloons in Sri Lanka and they have expanded their services within a short period of time. Currently, all the appoints are managed manually and the management of the saloon is expecting to develop a user-friendly mobile application with multiple features supporting both customers and administrator.

There are two types of users in the system

Administrator's functions:

- Register & login to the system
- Add/update/delete appointments
- View appointments
- Finish appointment with the bill amount
- View history

Customer's functions:

- Register & login to the system
- Make an appointment
- Cancel an appointment
- View appointments

Provide a well-designed, user-friendly system addressing the following features:

- System should have different authentication levels. (Administrator, customer)
- Interactive user-friendly interfaces.
- Add suitable sets of reports, which you think will add more value to the entire business.

Tasks

a) Provide the UML for the given problem with clear explanations on the design decisions. (Use case, Class, ER diagrams, Sequence diagrams etc.). (LO 1) (20 Marks)

b) Design user interfaces to fulfill all functionalities. (LO 2) (50 Marks)

c) Include test data and proper test plan for tasks 2 & 4 at the end of this document. (LO 4) (20 Marks)

d) Create well formatted documentation. (LO 4) (10 Marks)

Acknowledgement

Primarily I thank God for being able to complete this assignment in a successful manner. Thus, I take this opportunity to express my deep sense of gratitude and my profound respect to the lecturer who guided and inspired me in doing this assignment. I had to get the guidance of a respected and responsible person at the preparation time and while continuing the assignment. So, I would like to thank our lecturer Mr. Roy Lan whose valuable guidelines and consultations been the ones that helped me patch this assignment and make this full proof success and finalize this successfully.

And also, his instructions which were given underlying the structure has served as the major contributor towards in completing this assignment and his instructions about the programming and coding was more helpful in implementing this Mobile Application for SalonSD. Through those and by this golden opportunity, I got the full knowledge on basics and some advance in Mobile Application Development as well as in Java Programming. I hope this knowledge you gave me will help me in learning more advance Mobile Application Development as well as my career. I really thankful for you because of the massive courage you had to teach us.

Then I would like to thank Mr. Chathura, Miss Manoda and Miss Ishani who is with us and help us in many sides since the beginning of the Bridging Program. However finally, I am really grateful because I managed to complete this assignment within the time period given by our lecturer Mr. Roy Lan. Thank you all!

Contents

Introduction

This Assignment is regarding the Mobile Application Development Module and this is the seventeenth assignment we got in our HD Program. Basically, this coursework explains mobile operating systems, development tools and technologies for the mobile application development, design mobile application solutions, develop mobile application solutions and test mobile application solutions. Studying about Mobile Application Development is more important for a student who hopes to become a Software Engineer and a Mobile Application Developer as Java Programming directly explains the basics and advance things in building mobile applications and how complex codes are made using the basic stuffs in order to implement a mobile application with a low effort. In my assignment, I have ideally shown some theories used for UML diagrams designing as well as for coding along with the evidences.

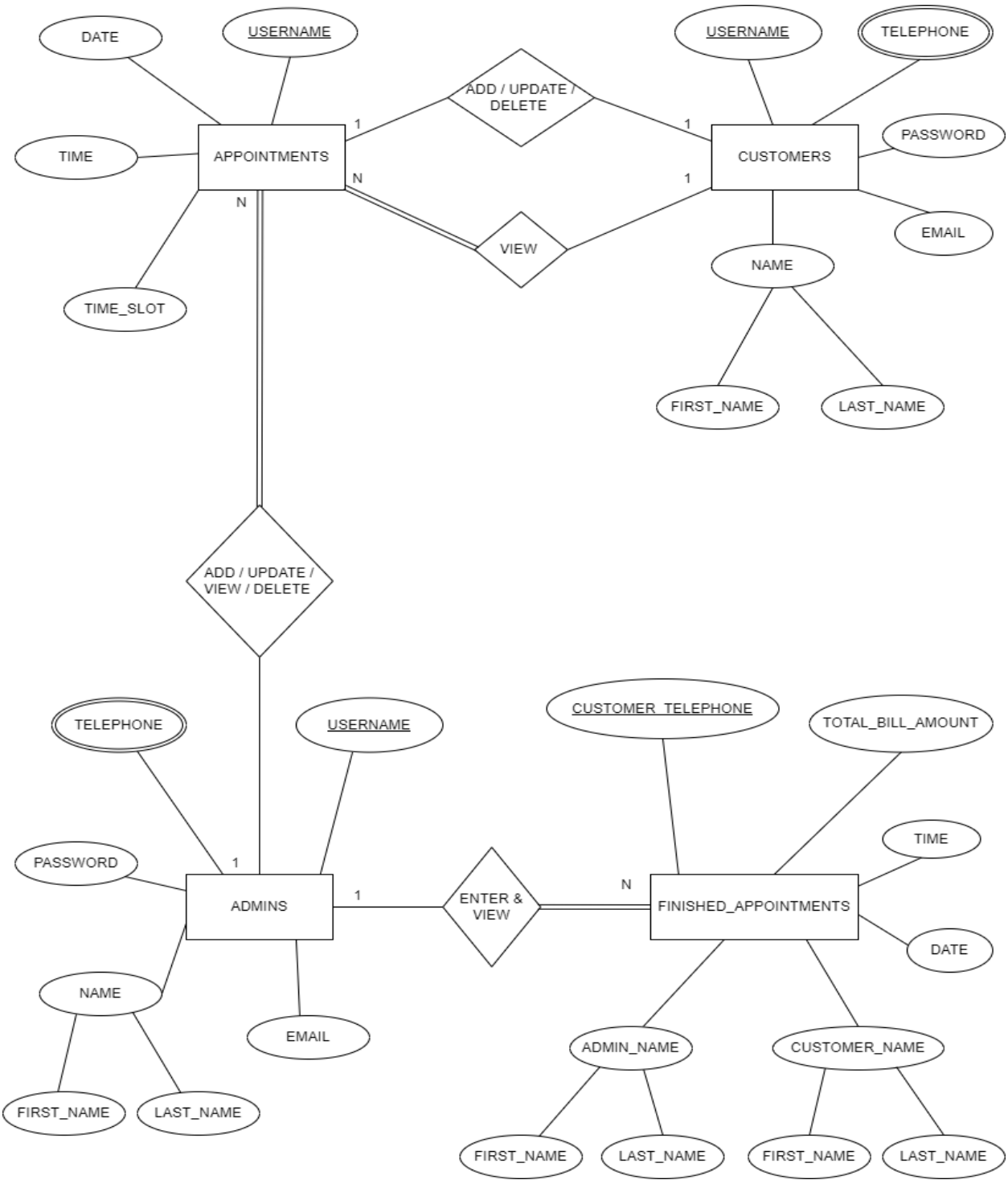
The primary and main objective of implementing this mobile application for Saloon SD is to support both customers and administrators to manage appointments which are currently handled manually. Saloon SD is a one of the famous saloons in Sri Lanka and they have expanded their services within a short period of time. They provide a quality service in hair cutting and give their admins control over their appointments is paramount at all SD Saloons. As this Mobile Application has two different authentication levels as Administrator and Customer, Admin will have the functionalities to register & login to the system, add/update/delete appointments, view booked appointments, finish appointment with the bill amount and view history of the finished appointments. Further, Customer will be able to register & login to the system, make an appointment, update an appointment, cancel an appointment and view booked appointments. In order to implement those functions, I had to include Java programming language which is more considerable when using to develop a Mobile Application in Android Studio. Thus, this SalonSD Mobile Application will help them to expand their business by increasing

customer base as this Mobile Application is more user friendly and easier to understand when using.

This document includes UML Diagrams such as an Entity Relationship diagram, an Use Case diagram, a Class diagram and Sequence diagrams with assumptions and explanations for each. In addition to them, this document consists with all the User Interfaces, analyzations along with description. Further, I have included a Test Plan and all relevant Test Cases to check validations and screenshots of the coding along with Android Virtual Device and assumptions that I have made. Thus, I hope this will be an ideal and clear assignment.

Task 1 - UML Diagrams

1.1 - ER Diagram



Within an information technology system like a Mobile Application, an ER diagram is an illustration which indicates relations with items, people, places, principles or activities. The major advantage of implementing an ERD is the database structure can be simply and perfectly understood after reviewing the diagram. Therefore, the above attached Entity Relationship diagram has been implemented to identify the entities, attributes and relationships in order to develop the database of the SalonSD Mobile Application (Arnicans, 1998).

When it comes to the ERD attached above, there are four entities called Appointments, Customers, Admins and Finished Appointments. Those entities are the database tables used. However, each of those entities consist with some attributes.

In Appointment entity, there are four attributes named as username, date, time and time slot. In that, username will be the primary key as it's the unique attribute in that entity and that's why it is underlined in the diagram. In Customers entity, there are five attributes named as username, telephone, password, email and name. In that, username will be the primary key as it's the unique attribute in that entity and that's why it is underlined in the diagram. Telephone will be a multi valued attribute as the existence of more than one phone number is possible. As there are values that are to be stored in an attribute can be further divided into sub-values and that's why name has drawn as a composite attribute. When it comes to the SalonSD mobile application, only a customer can add, update or delete an appointment which is only for his/her username. One customer shouldn't be able to delete other customers appointments and that's why the relationship between customers and appointments have named as one to one relationship. Only one customer can add, update or delete only one appointment which is for his/her username. But in order to make an appointment, a customer should be able to view all the booked appointments. Then he/she can see the lastly put time slot and reserve the next time slot for him/her. So, one customer can view many appointments which are booked and that's why it has named as one to many relationship. From customers entity, a single line is drawn to the relationship called View as it's a partial participation which means every customer won't make appointments. But from appointment entity, double lines are

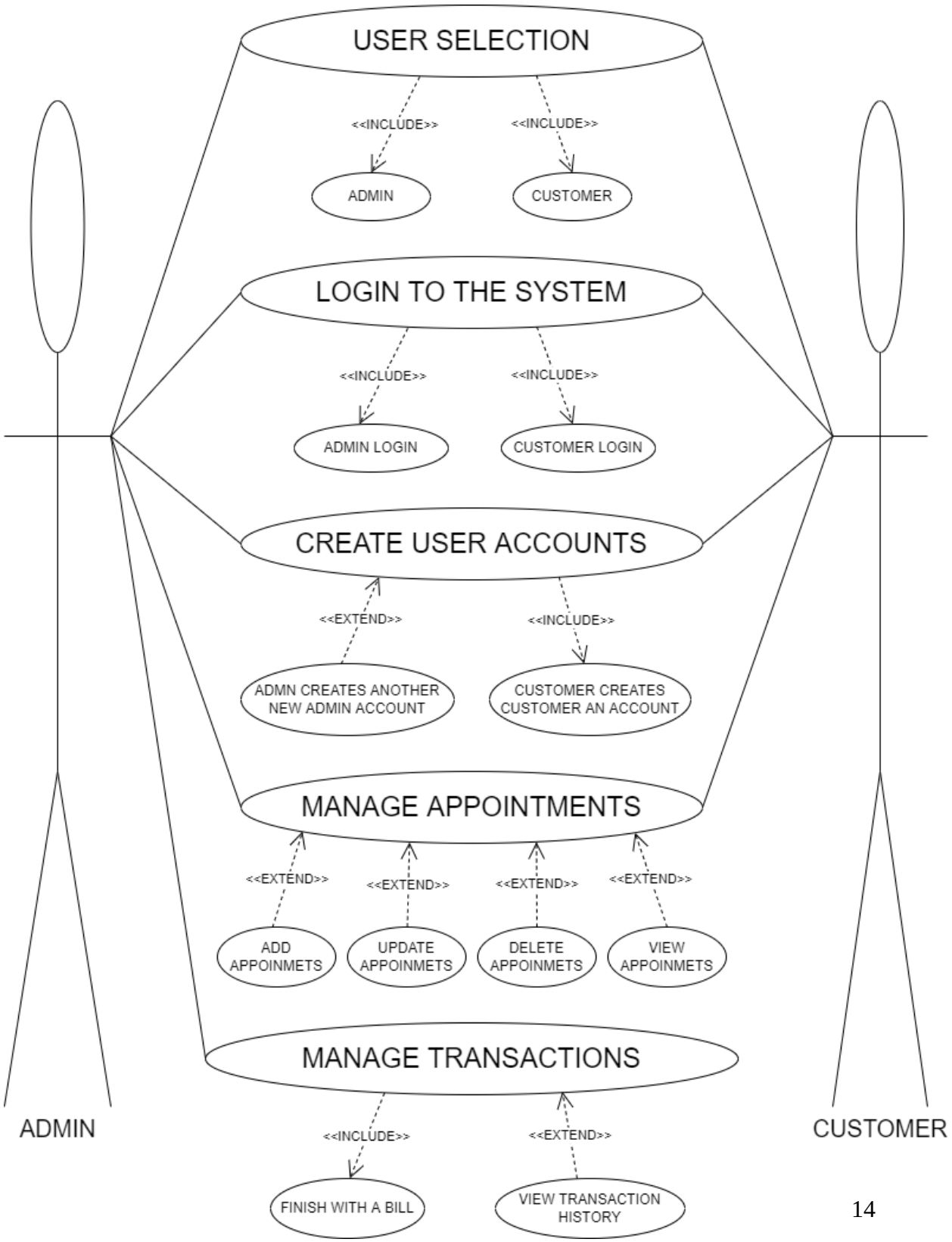
drawn to the relationship as it's a total participation which means one or many appointments can be made by one or many customers.

In Admins entity, there are five attributes named as username, telephone, password, email and name. In that, username will be the primary key as it's the unique attribute in that entity and that's why it is underlined in the diagram. Telephone will be a multi valued attribute as the existence of more than one phone number is possible. As there are values that are to be stored in an attribute can be further divided into sub-values and that's why name has drawn as a composite attribute. When it comes to the SalonSD mobile application, an admin has the functions to add, update, view or delete an appointment and appointment details. No matter that record is for whose username, an admin can update or delete any record from the appointment table and that's why the relationship between admins and appointments have named as one to many relationship. Because one admin can add, update, delete or view any and many appointments. From admin entity, a single line is drawn to the relationship as it's a partial participation which means every admin of the SalonSD won't make appointment. But from appointment entity, double lines are drawn to the relationship as it's a total participation which means one or many appointments can be made by one or many admins.

In Admins entity, there are six attributes named as customer telephone. Total bill amount, time, date, customer name and admin name. In that, customer telephone will be the primary key as it's the unique attribute in that entity and that's why it is underlined in the diagram. As there are values that are to be stored in an attribute can be further divided into sub-values and that's why admin name attribute and customer name attribute has drawn as composite attributes. When it comes to the SalonSD mobile application, an admin has the functions to enter finishing appointment details or view finished appointments and that's why the relationship between admins and finished appointments have named as one to many relationship. Because one admin can enter any and many finishing appointment details or view any or many finished appointments. From admin entity, a single line is drawn to the relationship called Enter & View as it's a partial

participation which means every admin of the SalonSD won’t finish billing appointments. But from finished appointments entity, double lines are drawn to the relationship as it’s a total participation which means one or many appointments can be finished and view by one or many admins.

1.2 - Use Case Diagram



The Use Case diagrams are being used to identify functions and how roles and actors interact with each of them. A Use Case diagram can be considered as a high-level picture of the system which emphasizes the roles that interact and the functionalities without focusing into the system's inner workings. (smartdraw, 1994-2022)

In the above attached Use Case diagram, there are five major functions. First main use case is User Selection where its mandatory for a user of Salon SD application to select his/her user type whether user is a customer or an admin. That's why sub use cases are matched to the main use case by using two include arrows. As shown in the diagram, both the users are accessible for that function in the first main use case.

Second main use case is Login To The System where its mandatory for both the users of Salon SD application to log in to the system in order to make appointments and move to the menu. Both the users have two different Login interfaces for each. As its mandatory for both users to login in order to make appointments and view menu, sub use cases are matched to the main use case by using two include arrows. As shown in the diagram, both the users are accessible for that function in the second main use case.

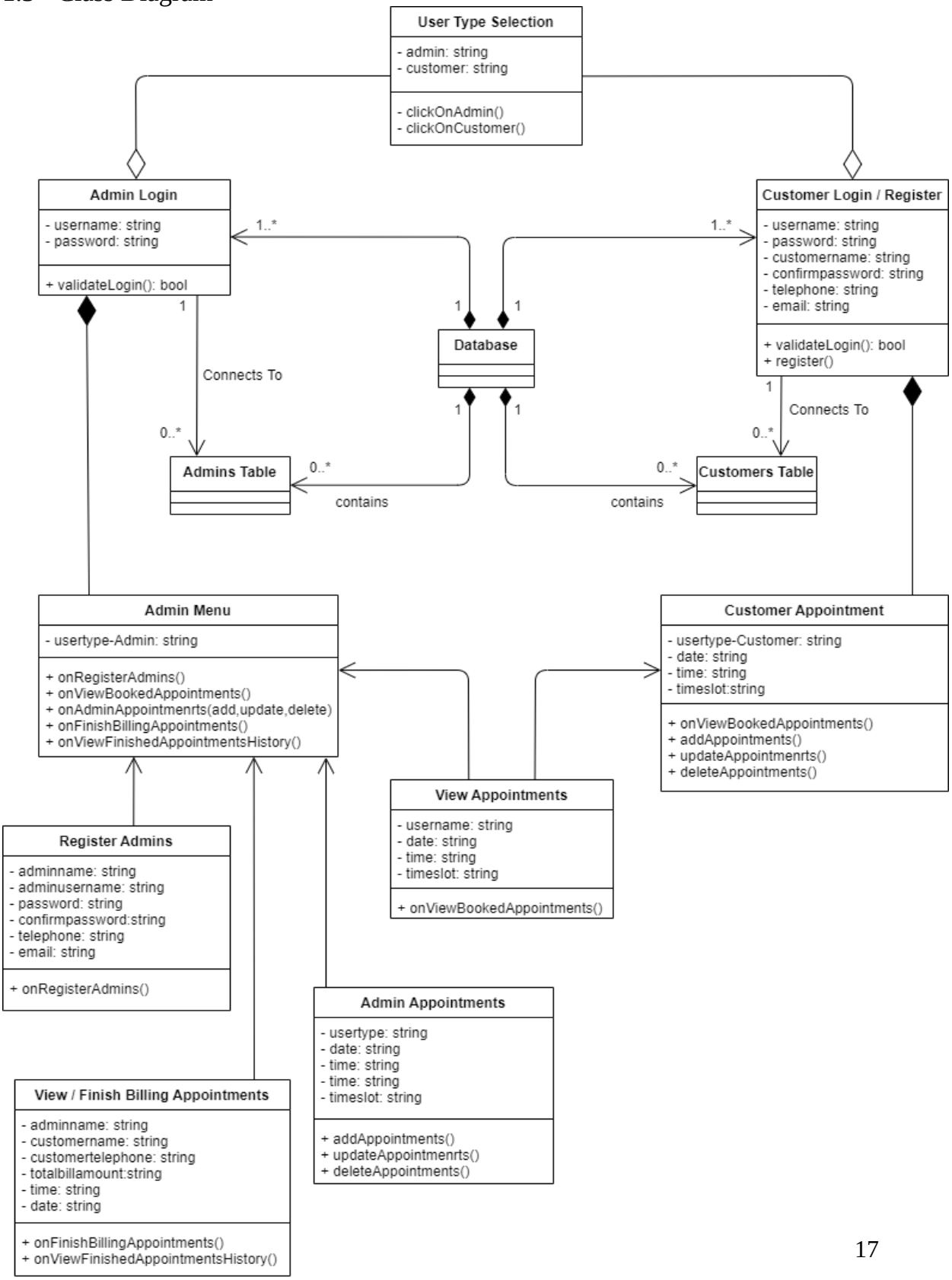
Third main use case is Create User Accounts where its not mandatory for an admin to create another admin account while it's a mandatory for a customer to register and then log in to make an appointment. Reason why it's not mandatory for an admin to create another admin account is there is a default username and a default password which are hardcoded for the main admin of the SalonSD to log in and manage works. That's why Creates Another Admin Account sub use case is matched to the main use case by using one extend arrow whereas Creates Customer Account sub use case is matched to the main use case by using an include arrow. As shown in the diagram, both the users are accessible for that function in the third main use case.

Fourth main use case is Manage Appointments where it's not mandatory for any user (admin/customer) to add, update, delete or view appointments of Salon SD application. That's why all four sub use cases are matched to the main use case by using four extend arrows. Because if a user wants to add, update, delete or view appointments only, he/she

can do that. As shown in the diagram, both the users are accessible for that function in the fourth main use case.

Last main use case is Manage Transactions where it’s mandatory for an admin to Finish With A Bill while it’s not mandatory for an admin to View Transaction History. Reason why its mandatory for an admin to finish with a bill is an admin should finish the appointments which are finished for a day. But its not mandatory to view transaction history by an admin. That’s why Finish With A Bill sub use case is matched to the main use case by using an include arrow whereas View Transaction History sub use case is matched to the main use case by using an extend arrow. As shown in the diagram, only an admin is accessible for that function in the last main use case.

1.3 - Class Diagram



Class diagrams are essential to the object modeling process because they represent a system's blueprints. Further, Class Diagrams are being used to represent the items that build a particular system like SalonSD Mobile Application, depict their connections and define what those objects perform. Basically, they explain how a system should be implemented. So, they are useful for visualizing, and documenting structural characteristics of SalonSD Mobile Application System. In Class Diagrams, an attribute inside a class is a significant piece of data containing values that describe each instance of that class. Attributes are also known as fields, variables or properties. After attributes, methods are put which are also known as functions or operations of a system. Methods allow to specify any behavioral feature of a particular class. (JavaTpoint., 2011-2021)

The visibility or "- sign" of an attribute or a method sets the accessibility for the attribute or method. In the above attached Class Diagram, I've used the visibility only for attributes to set the accessibility for those attributes. Further, this "- sign" indicates that all of the attributes are private. Methods inside all classes are public (indicated with + sign) as they can be accessed by any other class.

I have used Aggregation type of association which specify a whole and its parts where a part can exist outside the whole. The aggregation associations have been noted with an open diamond in the above attached Class Diagram.

The very first interface loaded is User Type Selection interface. Thats why two methods have used as one for Admins and the other for Customers. If an admin wants to login, there is a function that validates user login credentials. In both admin and customer login, there is a public method created for validating user returning a Boolean value.

Also, I have used relationships where the part can't exist outside the whole. Those type of relationships have been depicted with composition where I have noted with a colored and closed diamond. In composition type of relationships, a child object wouldn't be able to exist without its parent object.

Moreover, there won't load the Customer Menu interface if there is no any registered and logged in admins. Same as that, customers won't be able to make appointments and do things if there is no any registered and logged in customers. Thats why there are two composition relationships (with close diamond) from Admin Login class to Admin Menu class and Customer Login/Register class to Customer Appointment Class. If so, the database records wont exists.

Furthermore, I have used multiplicity in the above class diagram designed for SalonSD Mobile Application. Because Multiplicity allows to set numerical constraints on the relationships. In the class diagram, I have used specific number (1) multiplicity and zero to many (0..*) multiplicity. In attached diagram, it depicts that there can be only one database for the whole Mobile Application and that's why specific number (1) multiplicity has been used. Also there can be only one Admin Login for all the admins to log into the system and that's why specific number (1) multiplicity has been used. Same as for that, there can be only one Customer Login for all the customers of the SalonSD to log into the system and that's why specific number (1) multiplicity has been used there.

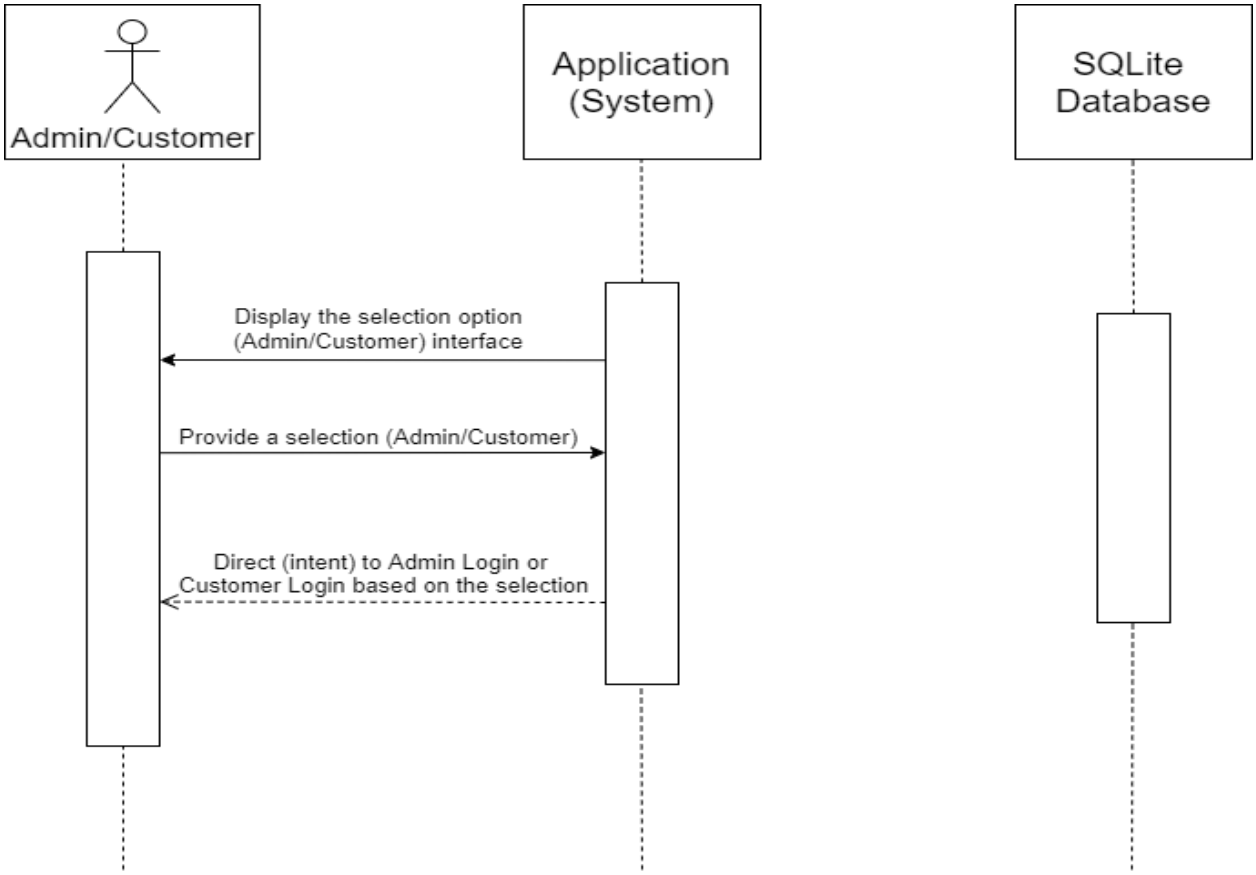
But there can be many numbers of Admin logins as well as Customer Logins. They will be connected with the Admins database table and Customers database table. Inside both Admins and Customers database tables, they may contain multiple user accounts for admins and customers. Thats what meant by the areas where zero to many (0..*) notation has been used.

All the classes which are matched to the Admin Menu class are child classes which are inherited from the parent class called Admin Menu. So, Methods inside the Admin Menu class have inherited to the child classes. Same as that, Customer Appointment parent class inherit the method of View Appointments. It's shown that both the Admin Menu and Customer Appointment parent classes inherit a same method called View Appointments which is a common child of both of that parent classes. But Customer Appointment has its own methods too.

1.4 - Sequence Diagrams

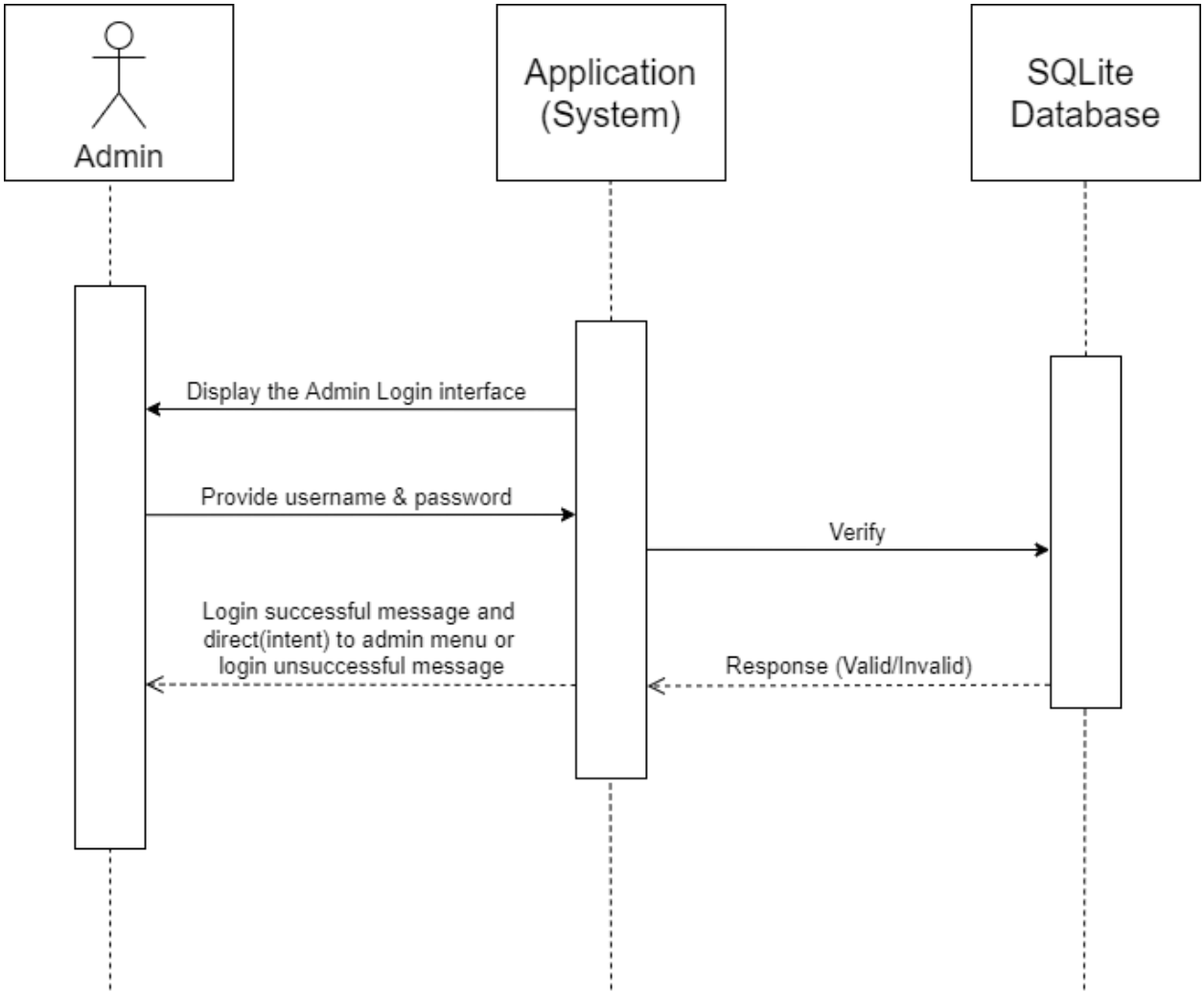
Sequence Diagram is also another kind of a Unified Modeling Language diagrams which describe the flow of messages, events and actions among objects. (Bell, 2004) When it comes to the SalonSD Mobile Application, there can be identified three objects as Admin/Customer, Application (System) and the SQLite Database.

1.4.1 - Very First Home (Starting) Interface



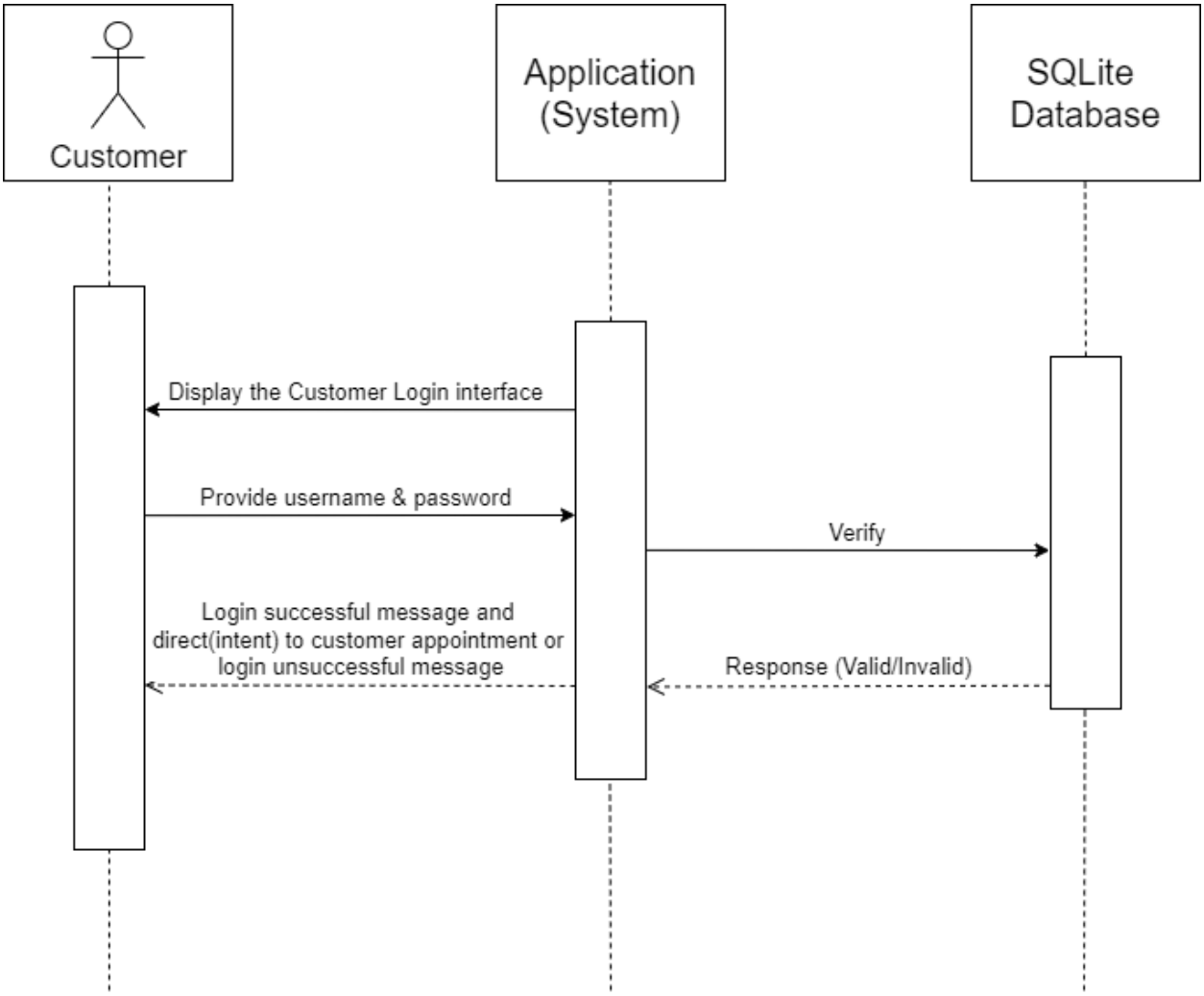
In the above sequence diagram, that is the very first interface which will be visible for both the users admin and customer. So, a user can select the button assigned for his/her user type and click on Admin or Customer. After user entering the relevant button, the system will direct user either to the Admin Login interface or the Customer Login Interface based on the user's input.

1.4.2 - Admin Login



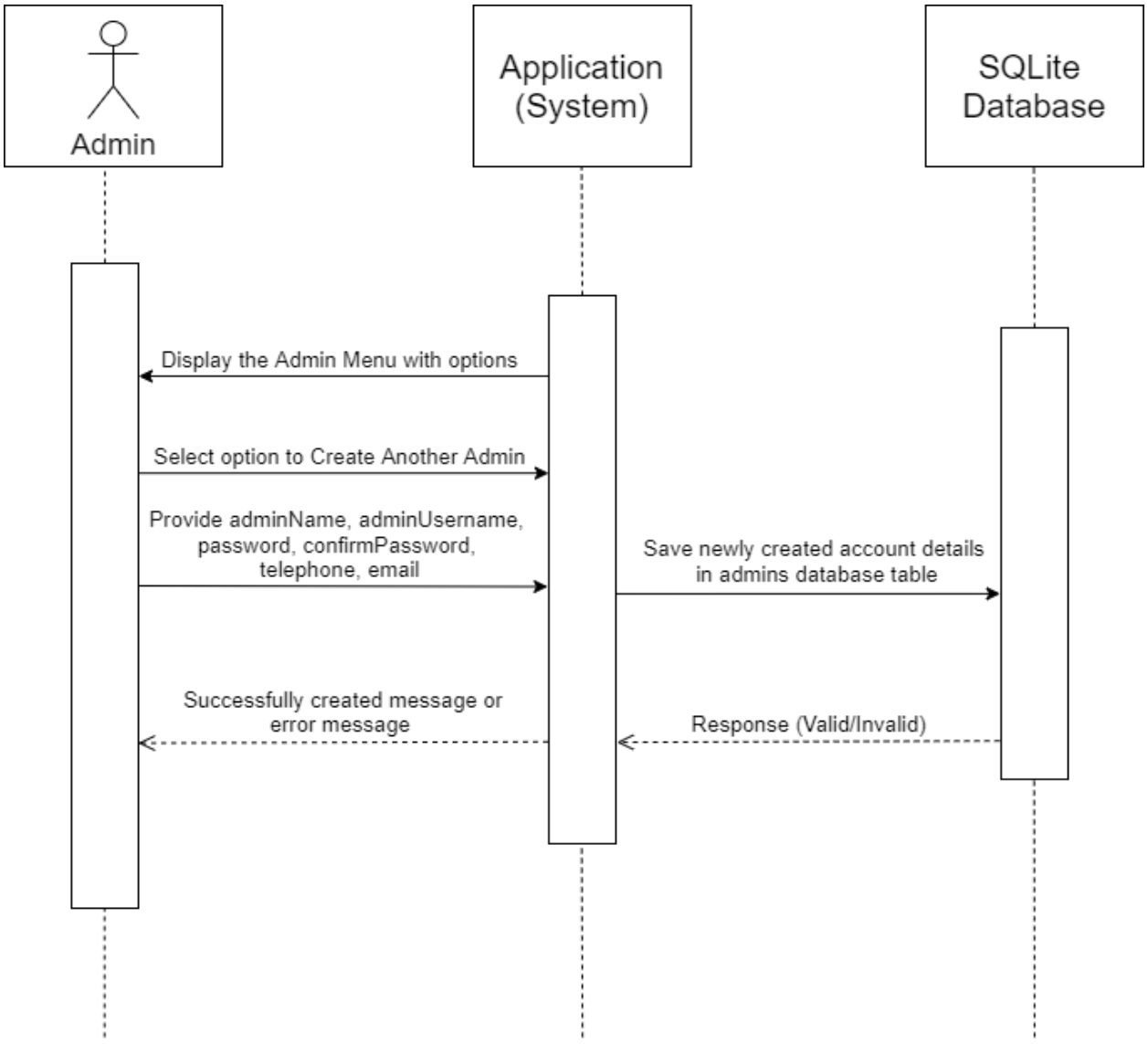
According to the above sequence diagram for Admin Login, the login interface will be displayed by the system for the user. Thus, admin should provide necessary details (username and password) and login. After admin entering the relevant data, the system verifies them with the use of SQLite Database and then return the response as Successful or not to the admin. If the admin entered login credentials are exists in the admins SQLite Database table, he/she is will be eligible to move to the Admin Menu Interface.

1.4.3 - Customer Login



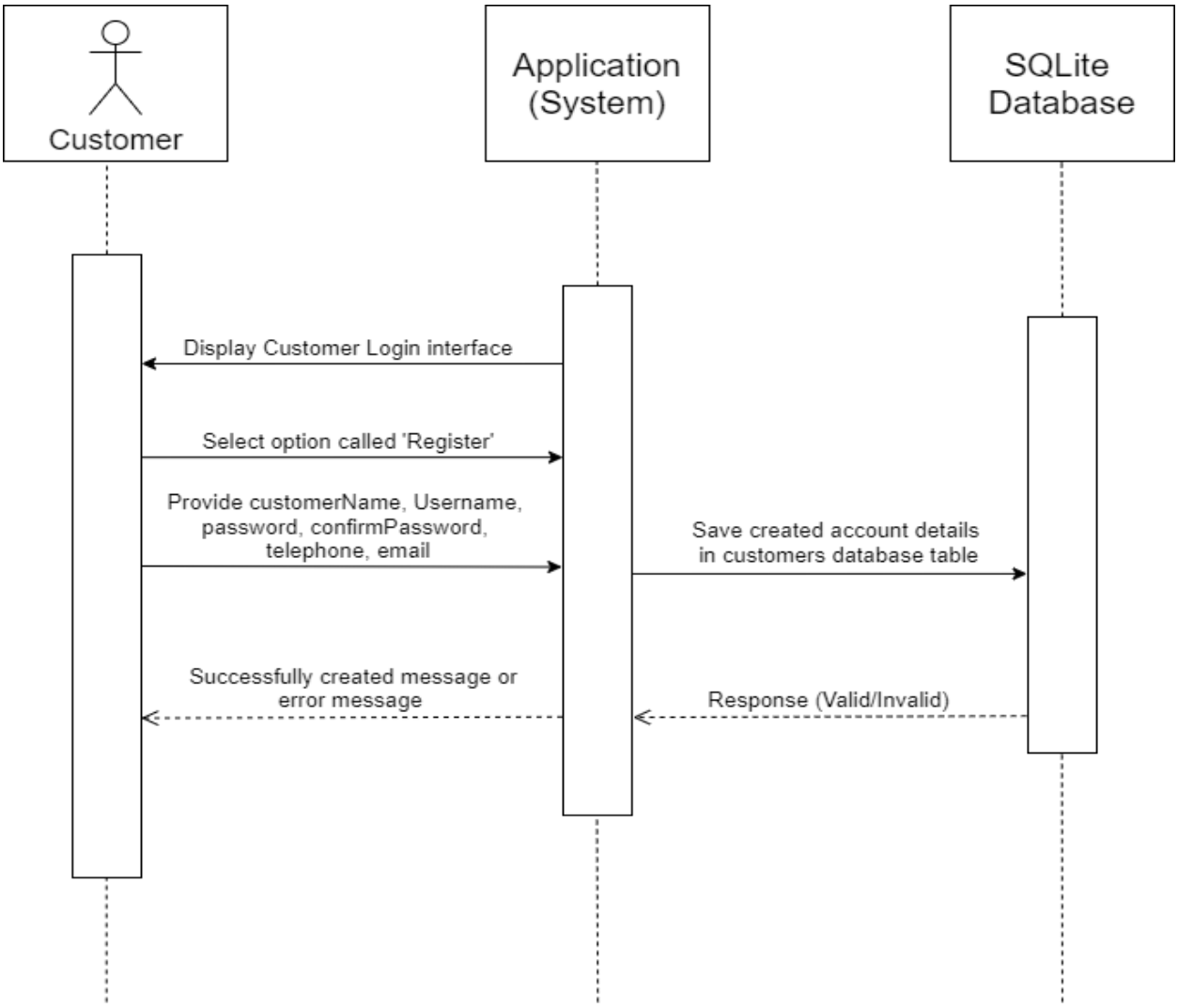
According to the above sequence diagram for Customer Login, the login interface will be displayed by the system for the user. But,a customer should register first in order to login. However, customers should provide necessary details (username and password) and login. After a customer entering the relevant data, the system verifies them with the use of SQLite Database and then return the response as Successful or not to the customer. If the customer entered login credentials are exists in the customers SQLite Database table, he/she is will be eligible to move to the Customer Appointment Interface and go ahead with making an appointment after viewing the booked appointments.

1.4.4 - Admin Register



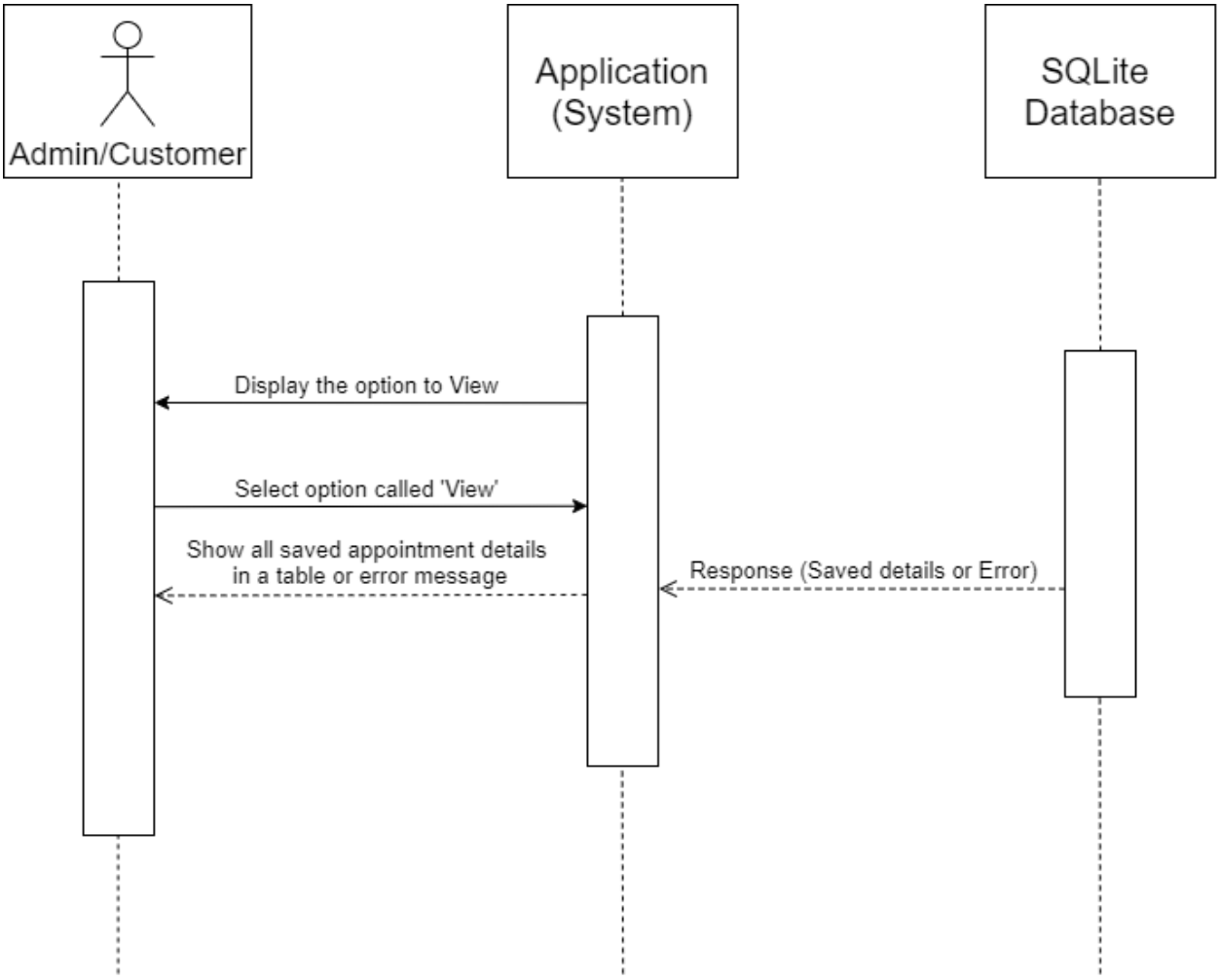
At first it is convenient to mention that this sequence diagram is relatable with admin as an admin can create more new accounts for admins. Thus, in the above sequence diagram for create Admin Account/s (Admin Registration), the system will display the manager menu primarily after logging. Then manager can select option to Create Another Admin Account and provide necessary information. That information will be saved in the relevant admin database table. Finally, a response will be sent to the user (Admin) mentioning whether account is successfully created or not.

1.4.5 - Customer Register



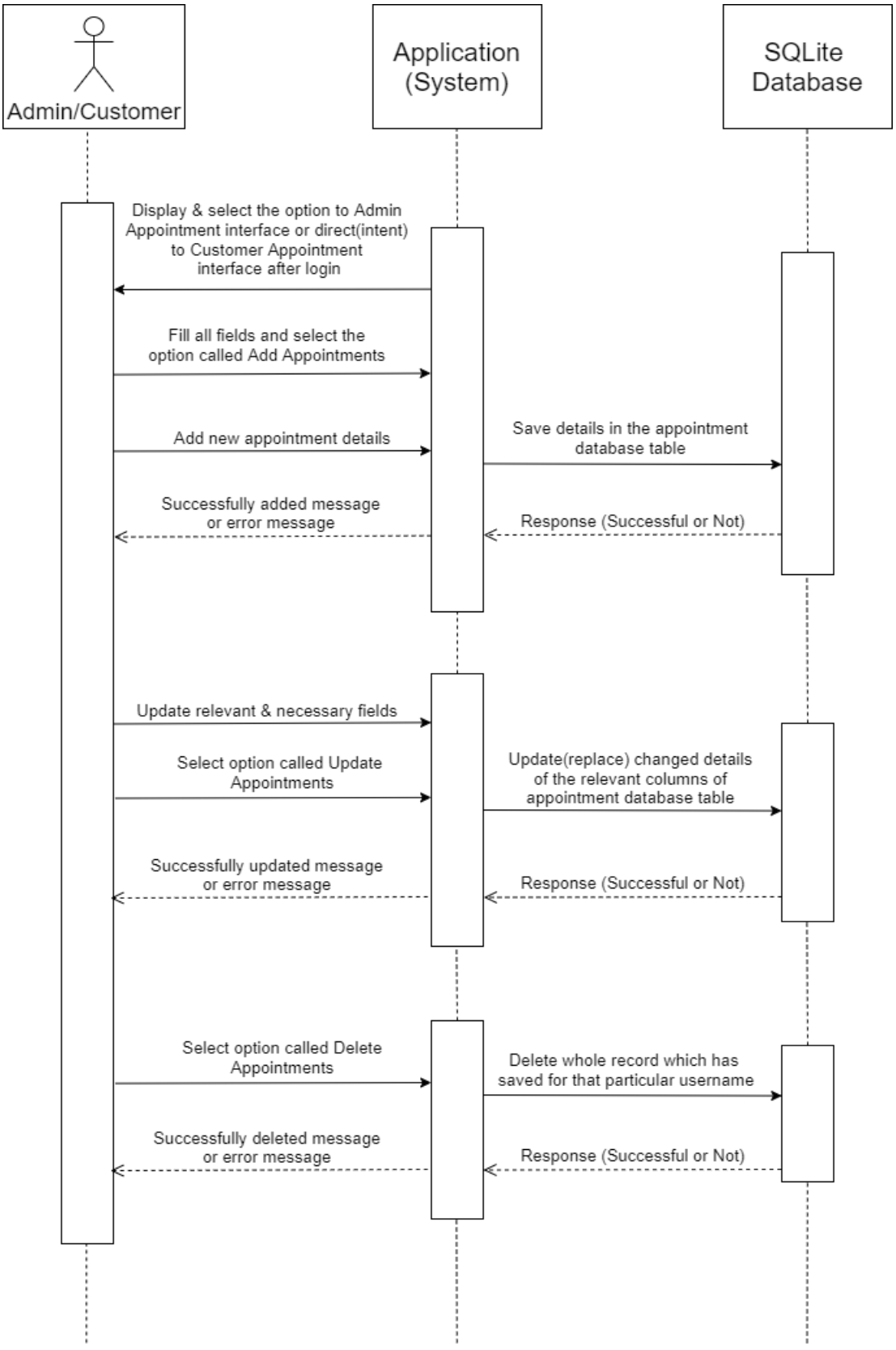
Here also, it is convenient to mention that this sequence diagram is relatable with customers as an customers should be registered first in order to login and make appointments after viewing booked appointments. Here the registration button will be in the login interface and customer should select on it. Then he/she will be directed into the customer registration interface. Thus, in the above sequence diagram for create Customer Registration. Then a customer can provide necessary information register. That information will be saved in the relevant customer database table. Finally, a response will be sent to the user (Customer) mentioning whether account is successfully created or not.

1.4.6 - Admin/Customer View Appointments



As in the above sequence diagram, its visible that there can be seen the function in the system as View Appointments. This particular function of viewing booked appointments can be accessed by both the users. If any admin wants to view appointments, first the admin menu will be displayed. Then admin has to select the option called View Booked Appointments and a response will be sent to the user by showing all the saved appointment details in a table or an error message. For customer, when they logged in, the Customer Appointment interface will be loaded and there is a transparent button in that interface with the texts on it called “Booked Appointments”. Therefore, customer can click on that and view the appointments.

1.4.7 - Admin/Customer Add/Update/Delete Appointments

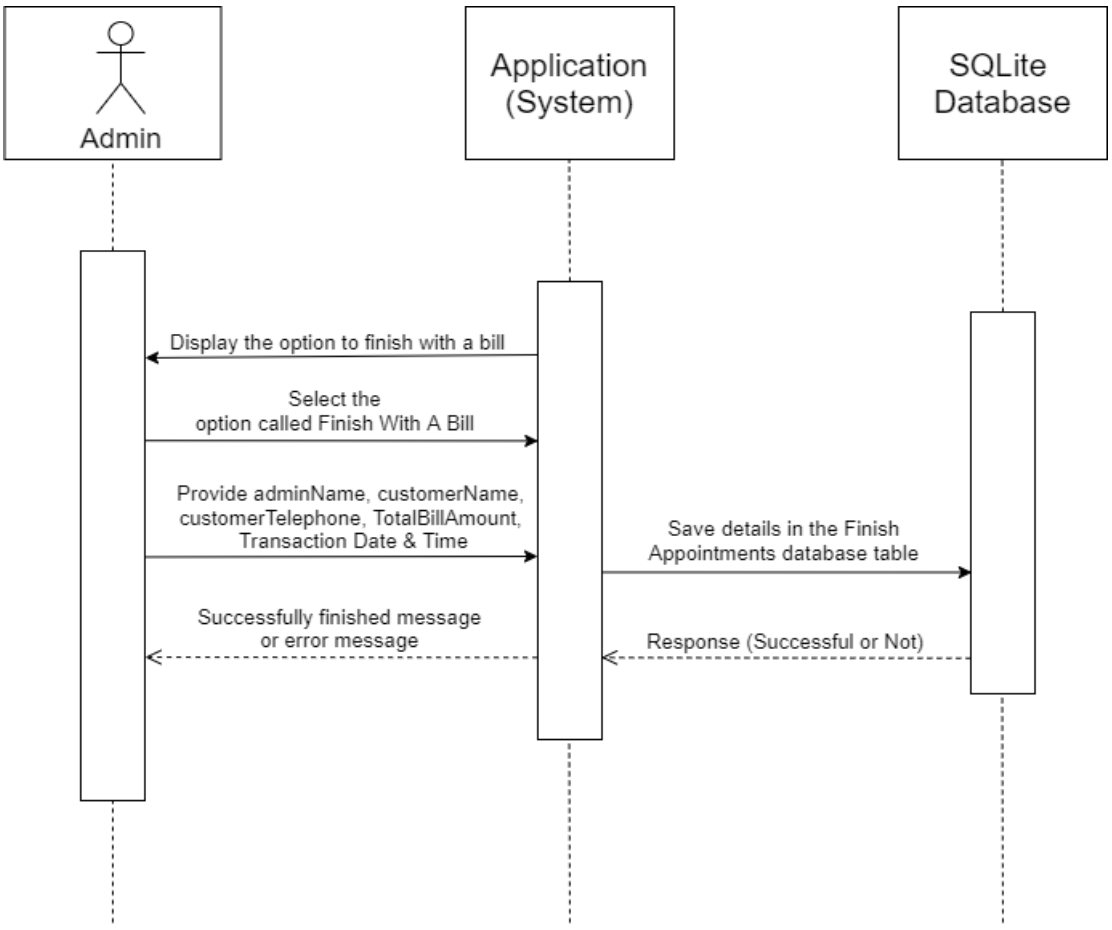


The above sequence diagram showing functions are accessible for both users. Customers can move to the interface of making an appointment after logged in whereas admin can move to the interface of making an appointment from the admin menu. If any of the users want to add an appointment, it's essential to view the booked appointments before making an appointment. Then user has to select the option called add appointment after entering particular appointment details on relevant fields. After that, particular appointment details will be saved in the appointment database table and a response will be sent to the user as successfully added or not.

If any of the users want to update an appointment, it's essential mention that admin can update any user's appointment details based on the username whereas customer can update only his/her appointment. Then user has to select the option called update appointment after changing particular appointment details on relevant fields. After that, particular appointment details will be updated in the appointment database table and a response will be sent to the user as successfully updated or not.

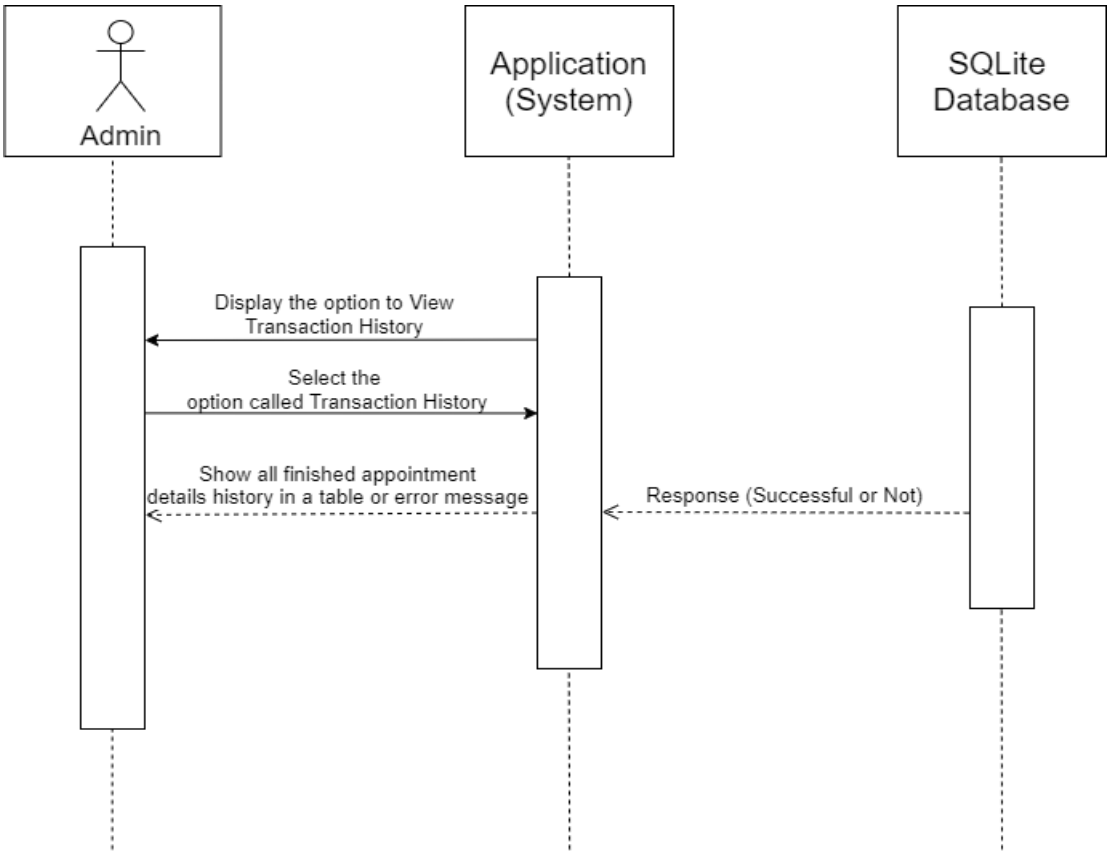
If any of the users want to delete an appointment, it's essential mention that admin can delete any user's appointment based on the username whereas customer can delete only his/her appointment. Admin should enter the deleting appointment's put user's username on the user type field and press on delete button while customer does not have anything to do expect just simply pressing on delete appointment button. After that, particular appointment record will be deleted from the appointment database table and a response will be sent to the user as successfully deleted or not.

1.4.8 - Admin Finish Billing Appointments



The above sequence diagram showing function of finish billing an appointment is accessible only for the user type admin. Admins can move to the interface of finish billing an appointment from the admin menu. Then user has to select the option called Finish With A Bill after entering particular appointment finishing details on relevant fields. After that, particular finished appointment details will be saved in the Finish Appointment database table and a response will be sent to the user as successfully finished or not.

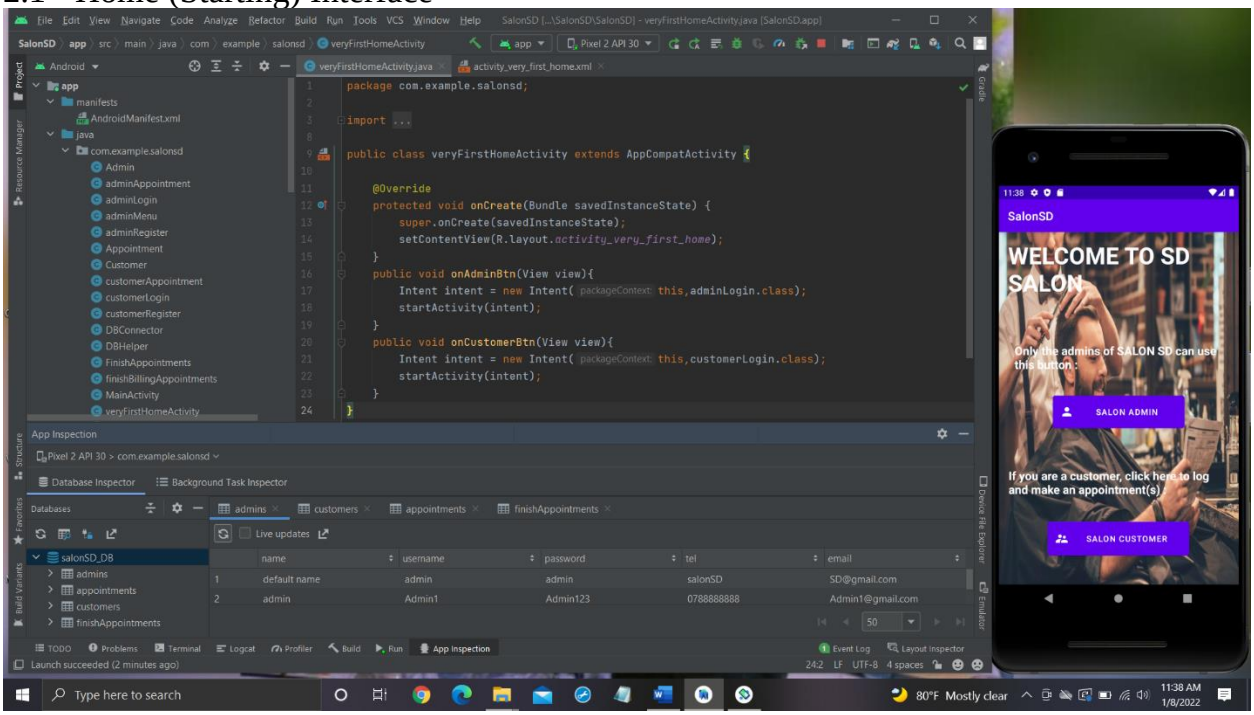
1.4.9 - View Finished Appointments History



As in the above sequence diagram, its visible that there can be seen the function in the system as View Finished Appointments. This particular function of viewing finished appointments can be accessed only by an admin. If any admin wants to view finished appointments, first the admin menu will be displayed. Then admin has to select the option called View Transaction History and a response will be sent to the user by showing all the saved finished appointment details in a table or an error message.

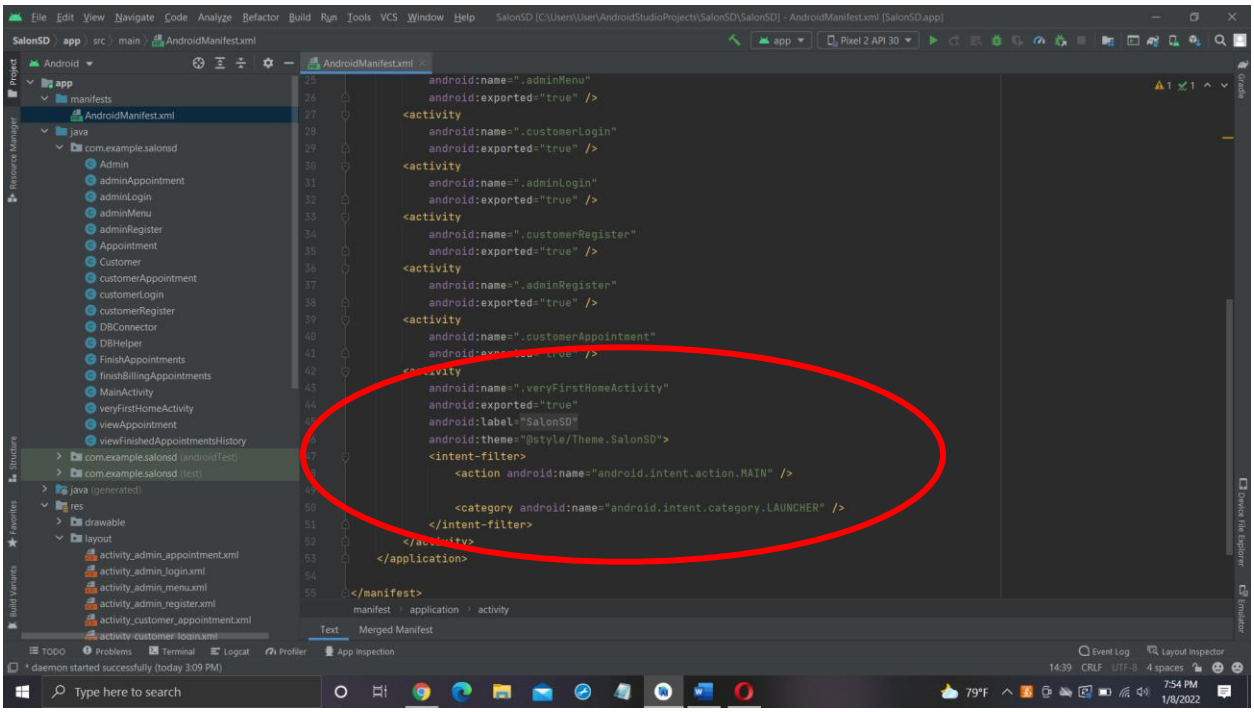
Task 2 - User Interfaces

2.1 - Home (Starting) Interface

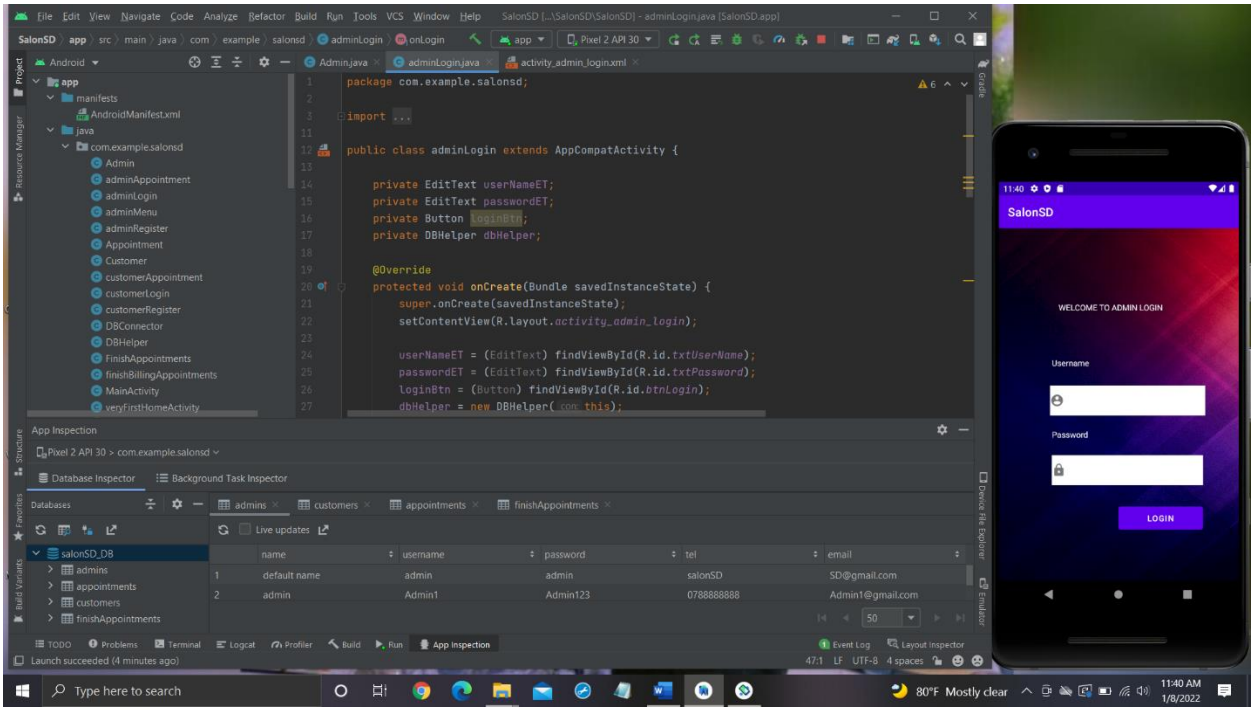


The interface shown in the attached screenshot is the very first interface which can be seen by a user. This UI consists with two buttons and three text views along with a background image. This Salon SD Mobile Application has only two user authentications. One user level is Admin while the other one is Customer. From the back-end, inside the class called veryFirstHomeActivity, I have created two on clicks in buttons and intent them separately as Admin button to the Admin Login interface and Customer button to the Customer Login. Thus, its clear that the very first interface which loads after run the application simple direct users to their respective login interfaces.

The below screenshot shows the codes used to load the veryFirstHomeActivity activity as the first activity to be loaded after running the application.

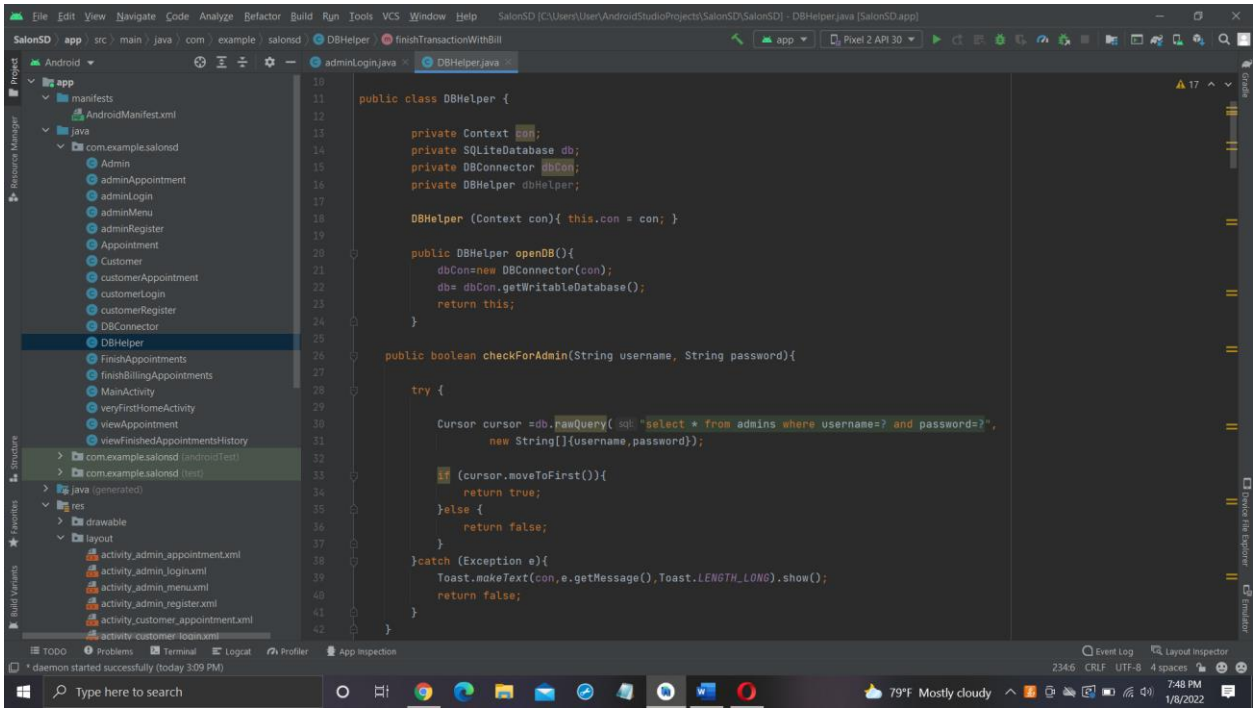


2.2 - Admin Login Interface

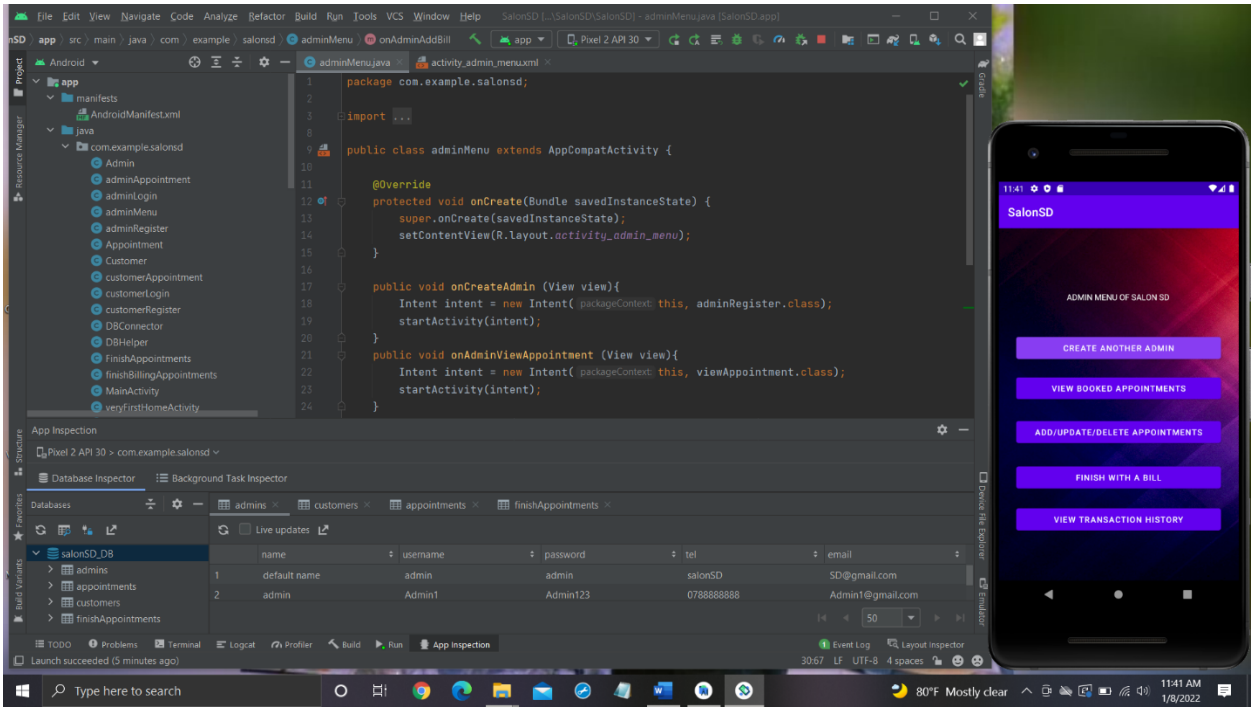


The interface shown in the attached screenshot can be seen by only a user who click on the admin button from the very first loaded interface. This Login interface is only for the Admins of the SalonSD. This UI consists with one button, two plain texts and three text views along with a background image. From the back-end, inside the class called adminLogin, I have created one on click in login button and intent it to the admin menu where admin functionalities are there. Here, when an admin sees this interface, he/she will have a default username called “admin” and a default password called “admin”. Only the admin who is with those two-default username and password can see the admin menu and create another admin if he/she wants. The validations used for this admin login interfaces include are empty field validation and correct username/password validation. Further, toast messages have used to show the messages inside the validations. All those coding has done inside the on click of the login button.

The below screenshot shows the database connection that is used to check the correctness of an admin's username and password. Select query has been used for that.

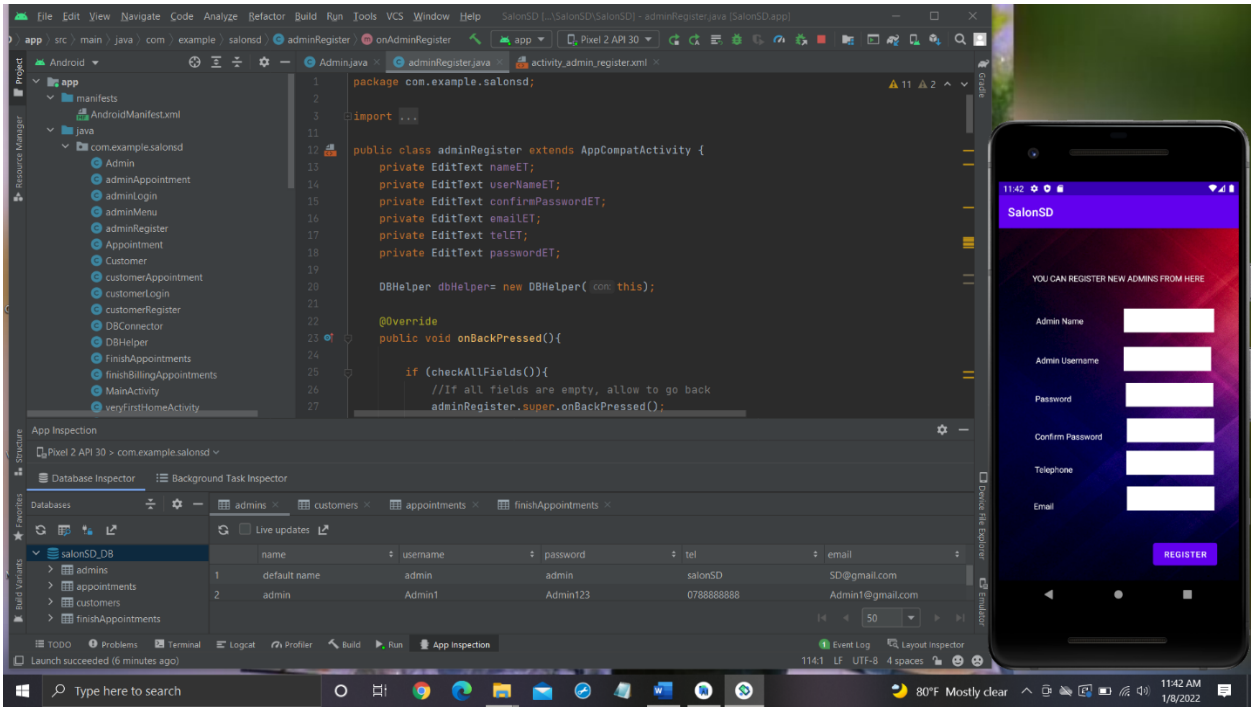


2.3 - Admin Menu Interface



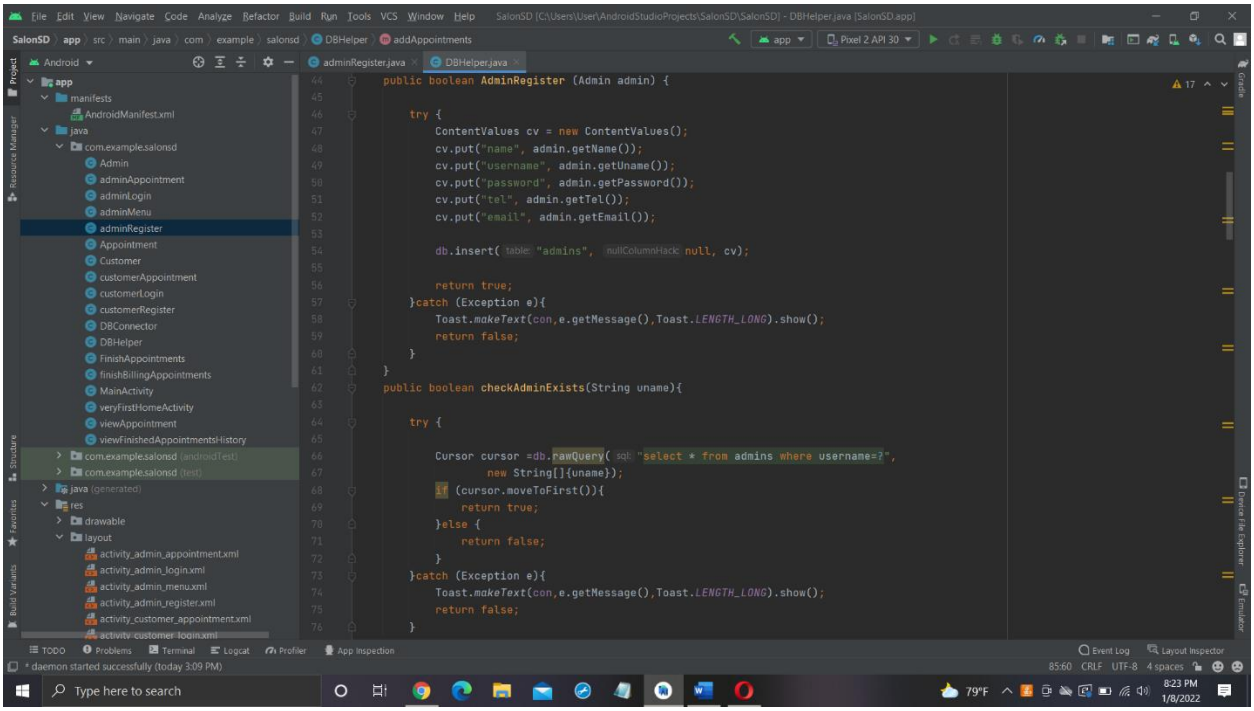
The interface shown in the above screenshot can be only seen by an admin. Because admin menu is the interface which loads when an admin entered correct username/password and logged in. This UI consists with five buttons and one text view along with a background image. From the back-end, inside the class called veryFirstHomeActivity, I have created five on clicks in buttons and intent them separately as Create Another Admin button to the Admin Register interface, View Booked Appointments button to the View Appointments interface, Add/Update/Delete Appointment button to the Admin Appointment interface, Finish With A Bill button to the Finish Billing Appointments interface and View Transaction History button to View Finished Appointment History interface. Thus, it's clear that above shown interface can be access only by an admin whos with a valid admin username and a valid admin password.

2.4 - Admin Registration Interface

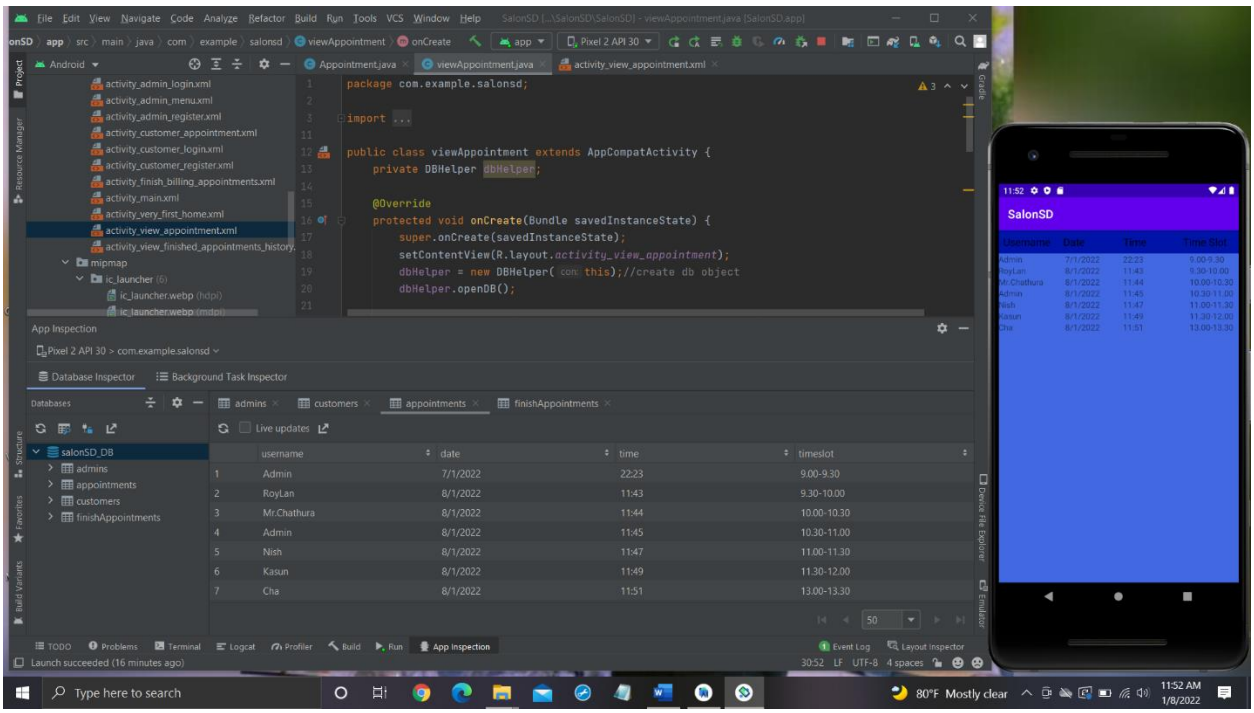


The interface shown in the attached screenshot will be appeared only when admin clicks Create Another Admin button from the Admin Menu. Only after the main admin log using default username and password can create another admin. This UI consists with one button, six plain texts and seven text views along with a background image. From the back-end, inside the class called adminRegister, I have created all the validations for the admin registration. The validations used for this admin registration include are empty field validation, password and confirm password matching validation, email validation and username validation which displays the message as a user already exists or not. Further, toast messages have used to show the messages inside the validations. All those coding has done inside the on click of the register button.

The below screenshot shows the methods and database connections that are used to check for existing users where select query has been used and the query that has used for inserting data obtain from the admin register interface.

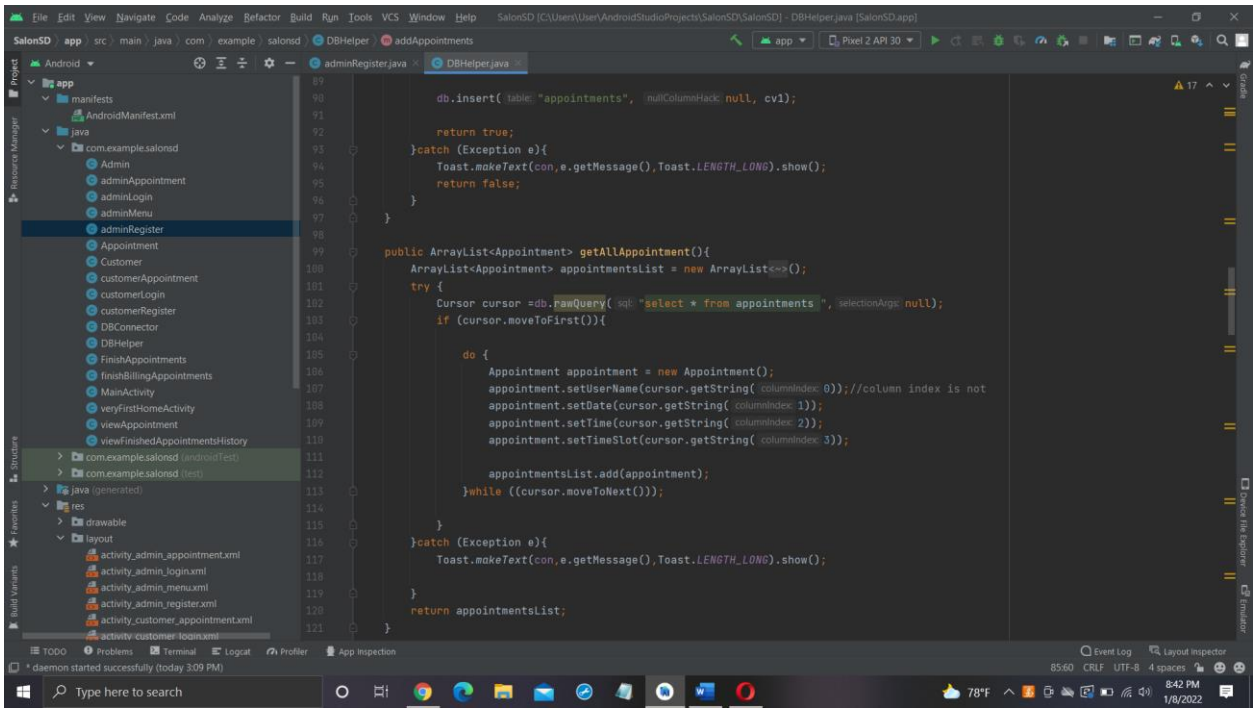


2.5 - Admin/Customer View Appointments Interface

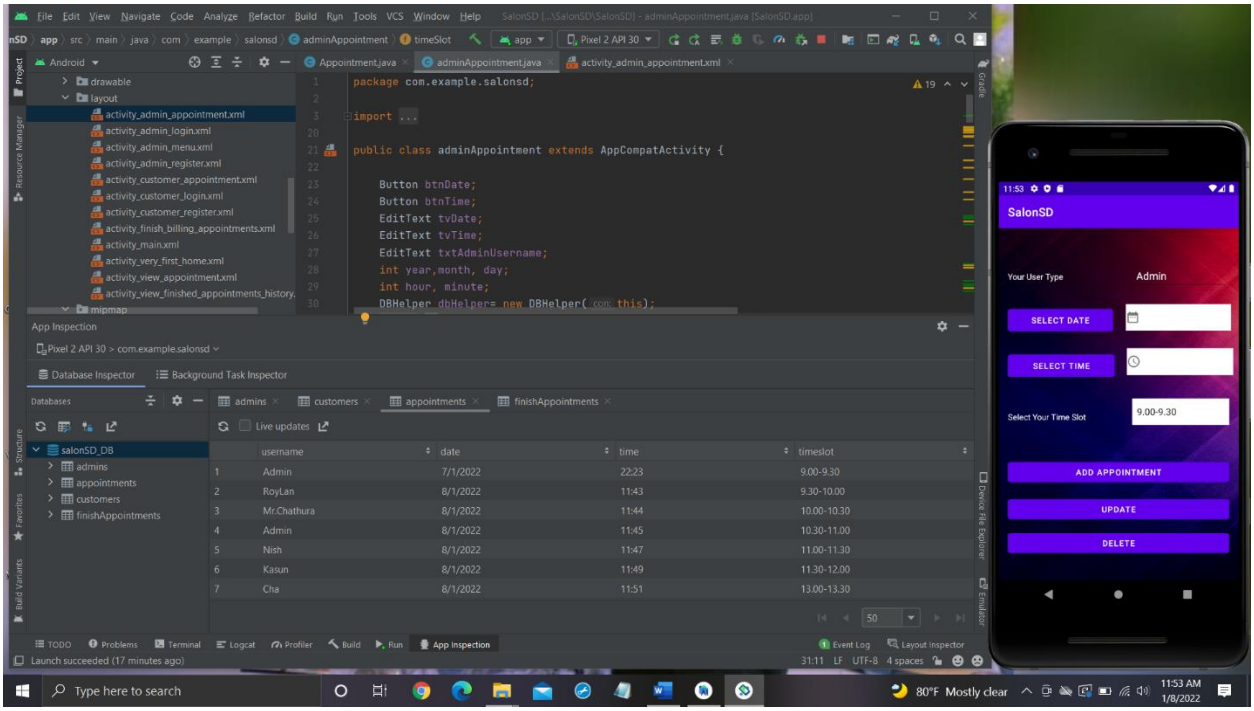


This above attached screenshot of the interface will be loaded when an admin clicks on the View Booked Appointments button from the Admin Menu interface. This is a common interface for both admins and customers. Customer can see this interface when making an appointment. The layout used for this interface is a table layout. The table consists of four columns as username, date, time and time slot. There is no attached background image for this interface. This interface is with the blue color in the background. From the back-end, inside the class called viewAppointments, I have designed the table layout to display columns and rows along with an Array List.

The below screenshot shows the methods and database connections that are used to retrieve booked appointment details with the select query.

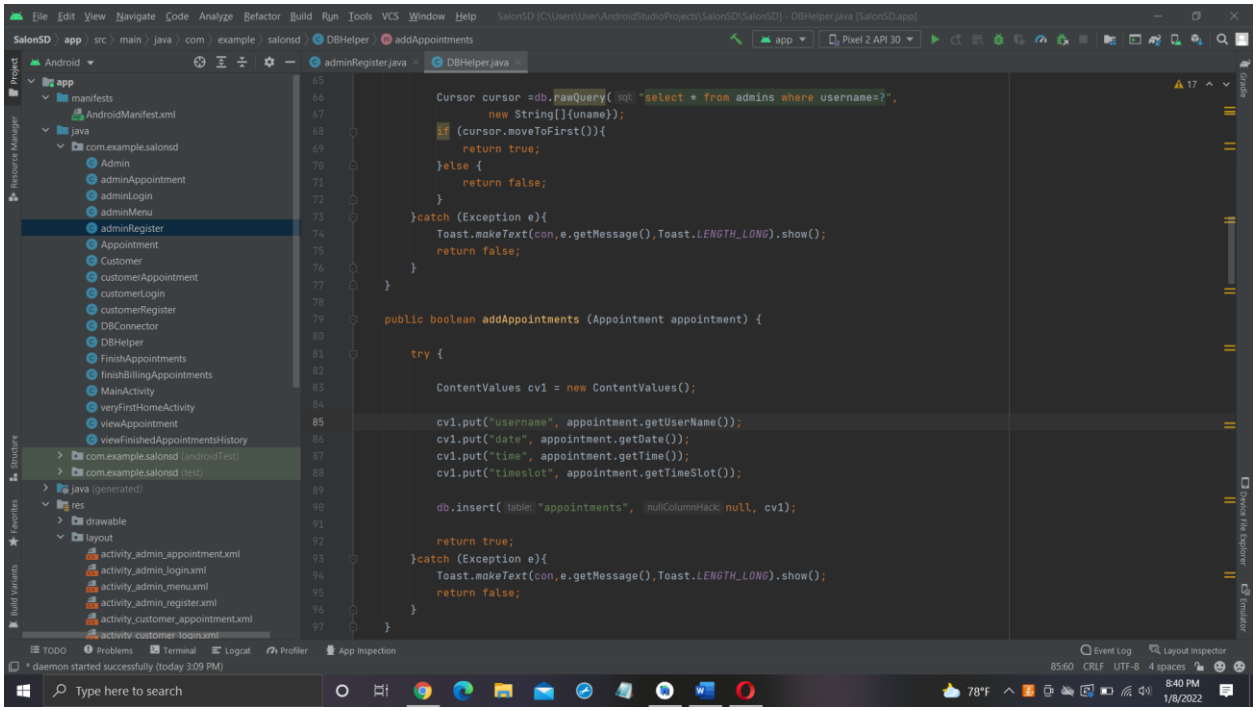


2.6 - Admin Appointment Interface (Add/Update/Delete)



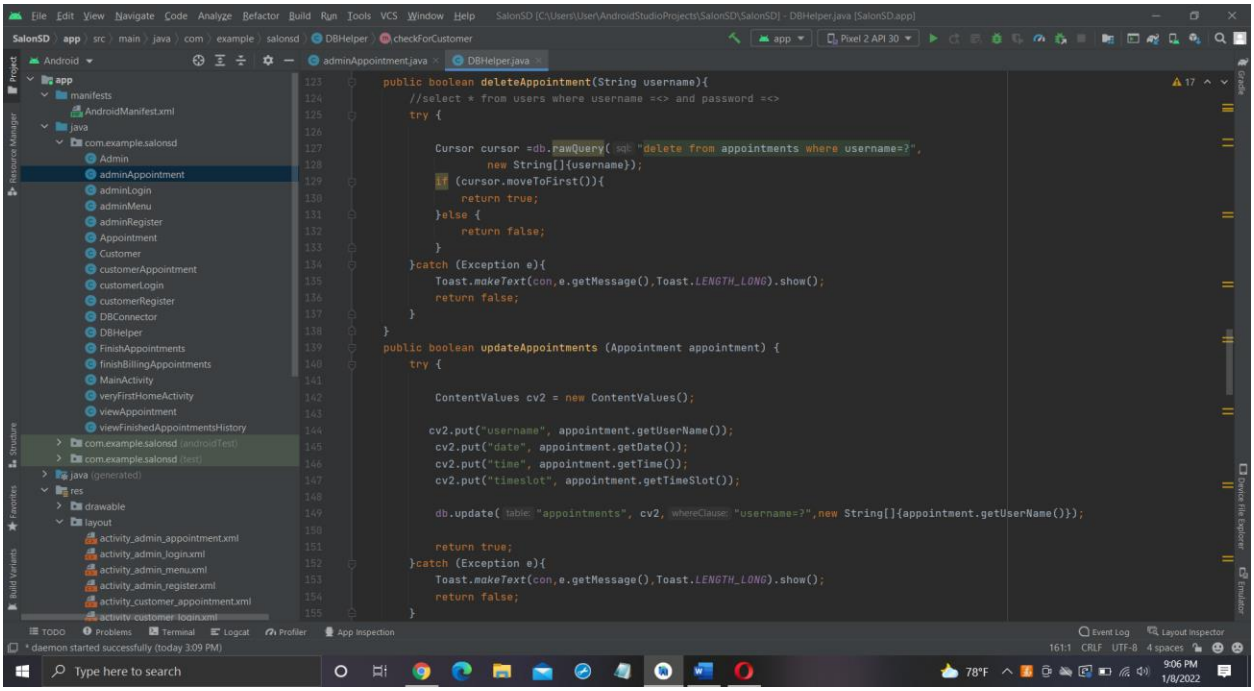
The interface shown in the attached screenshot will be appeared only when admin clicks Add/Update/Delete Appointment button from the Admin Menu. This UI consists with five buttons, four plain texts and two text views along with a background image. There is a date and time picker to get the current date and the time. Time slot can be selected only after viewing the booked appointments and a spinner has used for that. From the back-end, inside the class called adminAppointment, I have created on click listeners for date and time picker and three on clicks have been made separately for three buttons called Add, Update, Delete and validations have included inside them. The validations used for this admin appointment include are empty field validation and successfully inserted or not validation. Further, toast messages have used to show the messages inside the validations.

The below screenshot shows the methods and database connections that are used to insert appointment details.



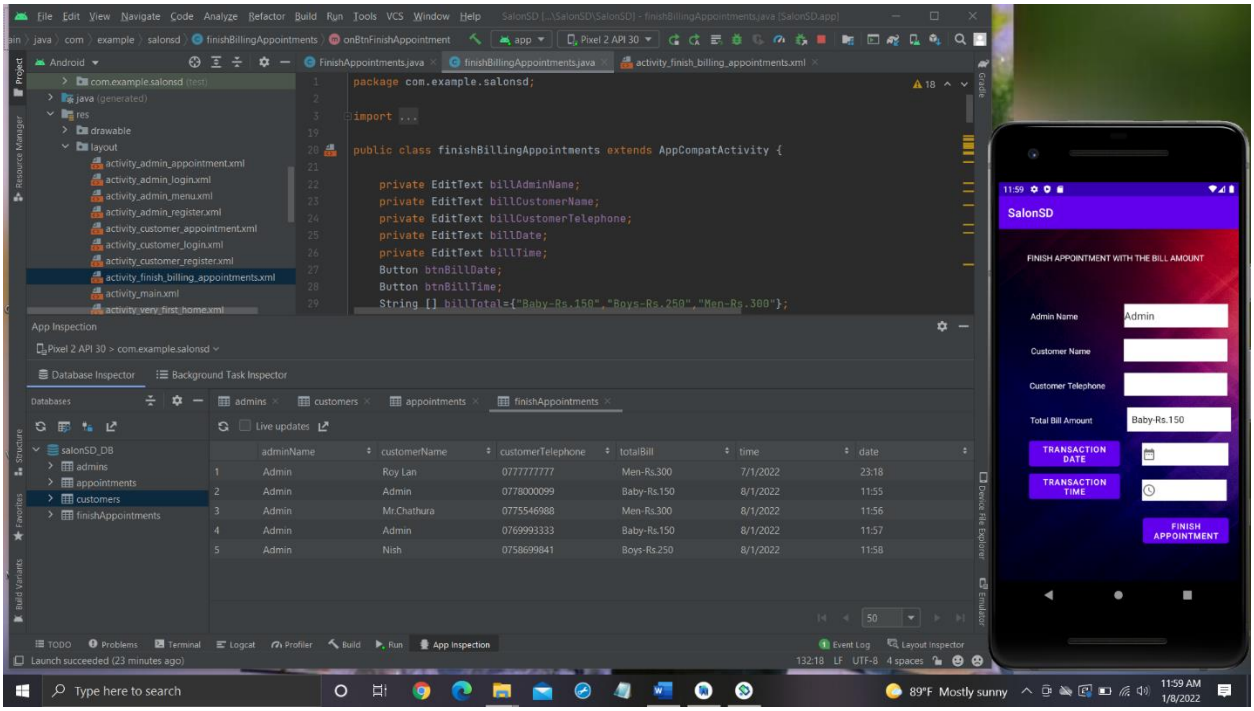
```
Cursor cursor =db.rawQuery( sql: "select * from admins where username=?",
        new String[]{uname});
if (cursor.moveToFirst()){
    return true;
}else {
    return false;
}
}catch (Exception e){
Toast.makeText(con,e.getMessage(),Toast.LENGTH_LONG).show();
return false;
}
}

public boolean addAppointments (Appointment appointment) {

try {

    ContentValues cv1 = new ContentValues();

    cv1.put("username", appointment.getUserName());
    cv1.put("date", appointment.getDate());
    cv1.put("time", appointment.getTime());
    cv1.put("timeslot", appointment.getTimeSlot());

    db.insert( table: "appointments",  nullColumnHack: null, cv1);

    return true;
}catch (Exception e){
    Toast.makeText(con,e.getMessage(),Toast.LENGTH_LONG).show();
    return false;
}
}
```

The below screenshot shows the methods and database connections that are used to update/delete appointment details.



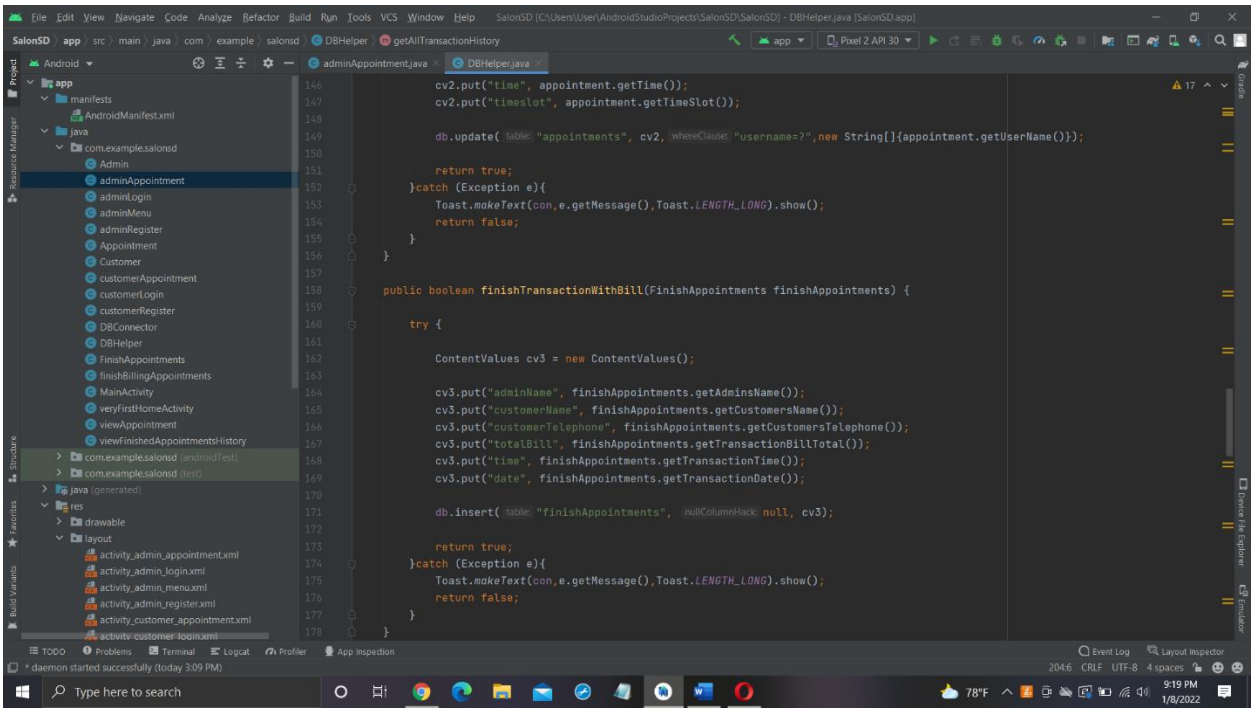
```
public boolean deleteAppointment(String username){
    //select * from users where username =<> and password =<>
    try {

        Cursor cursor =db.rawQuery( sql: "delete from appointments where username=?",
                new String[]{username});
        if (cursor.moveToFirst()){
            return true;
        }else {
            return false;
        }
    }catch (Exception e){
        Toast.makeText(con,e.getMessage(),Toast.LENGTH_LONG).show();
        return false;
    }
}
public boolean updateAppointments (Appointment appointment) {
    try {

        ContentValues cv2 = new ContentValues();

       cv2.put("username", appointment.getUserName());
        cv2.put("date", appointment.getDate());
        cv2.put("time", appointment.getTime());
        cv2.put("timeslot", appointment.getTimeSlot());

        db.update( table: "appointments", cv2, whereClause: "username=?",new String[]{appointment.getUserName()});

        return true;
    }catch (Exception e){
        Toast.makeText(con,e.getMessage(),Toast.LENGTH_LONG).show();
        return false;
    }
```

2.7 - Admin Finish Billing Appointments Interface

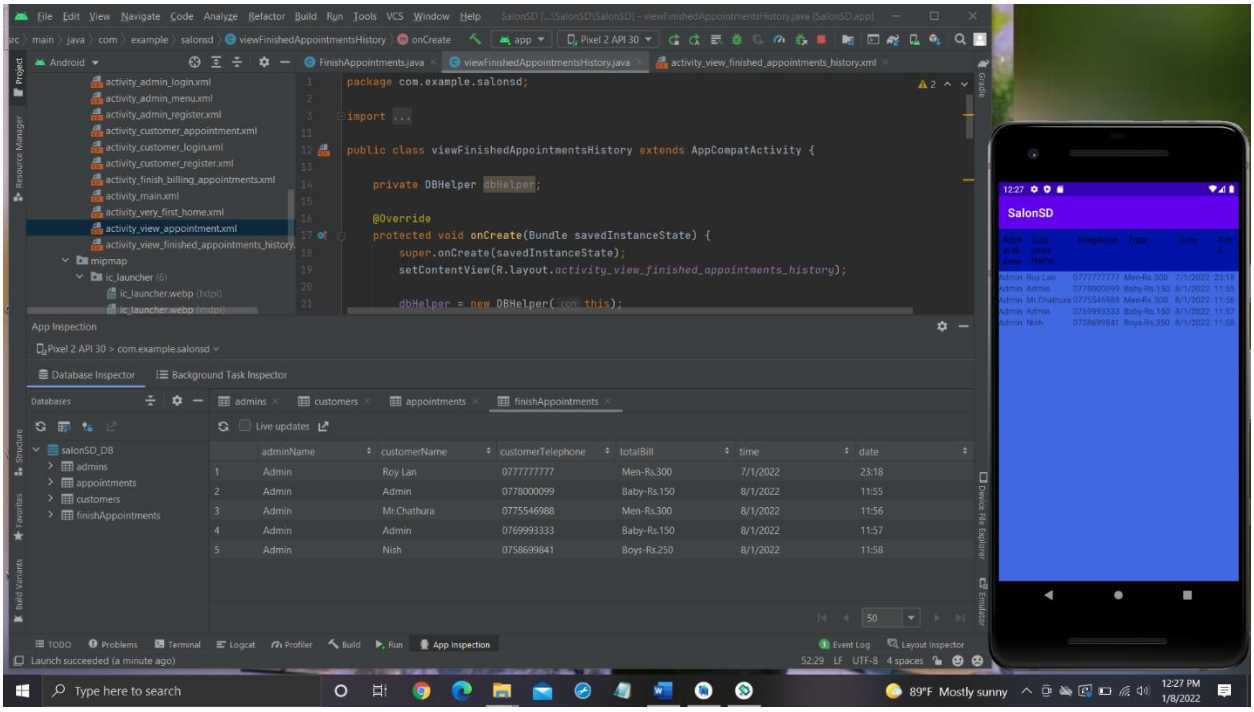


The interface shown in the attached screenshot will be appeared only when admin clicks Finish Appointments button from the Admin Menu. This UI consists with three buttons, six plain texts and five text views along with a background image. There is a date and time picker to get the current date and the time. Total Bill Amount can be selected and a spinner has used for that. From the back-end, inside the class called finishBillingAppointments, I have created on click listeners for date and time picker and a on click has been made for the button called Finish Appointment and validations have included inside it. The validations used for this finish billing appointment include are empty field validation and successfully finished or not validation. Further, toast messages have used to show the messages inside the validations.

The below screenshot shows the methods and database connections that are used to insert finished appointment details.

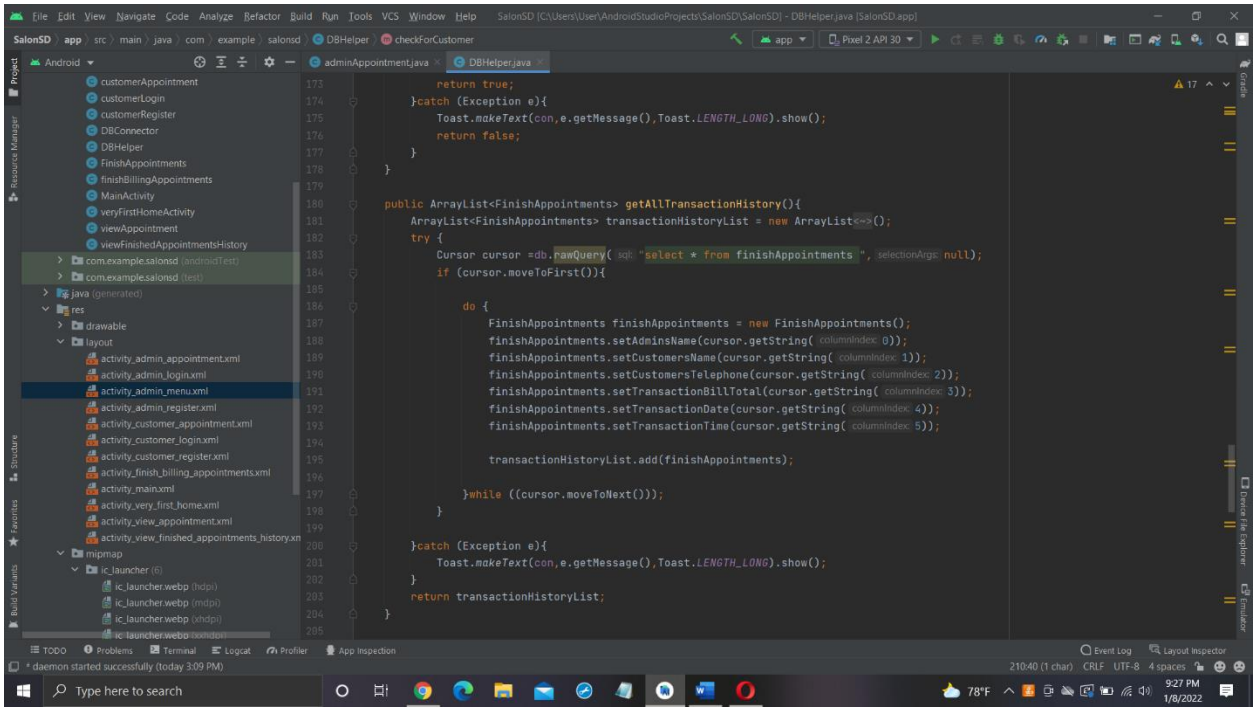


2.8 - Admin View Finished Appointments History Interface

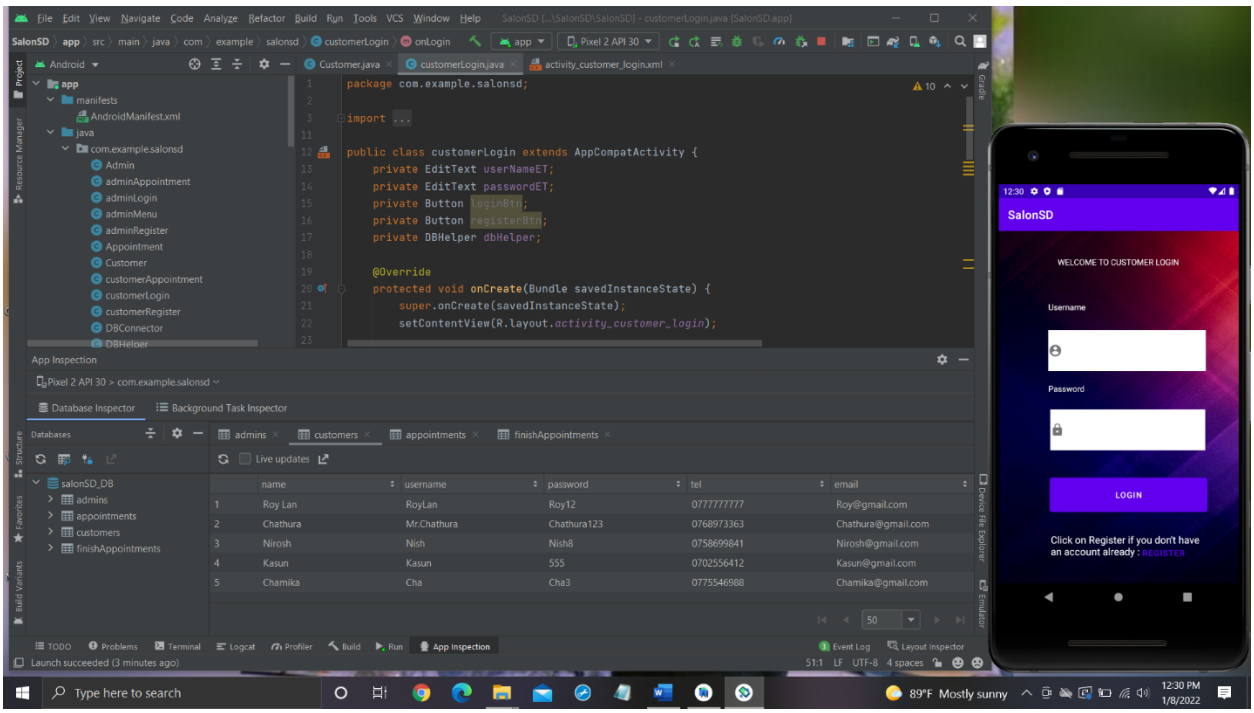


This above attached screenshot of the interface will be loaded when an admin clicks on the View Transaction History button from the Admin Menu interface. This is interface is only accessible for an admin. The layout used for this interface is a table layout. The table consists of six columns as admin name, customer name, telephone, total, date and time. There is no attached background image for this interface. This interface is with the blue color in the background. From the back-end, inside the class called viewFinishedAppointmentsHistory, I have designed the table layout to display columns and rows along with an Array List.

The below screenshot shows the methods and database connections that are used to retrieve finished appointment details with the select query.



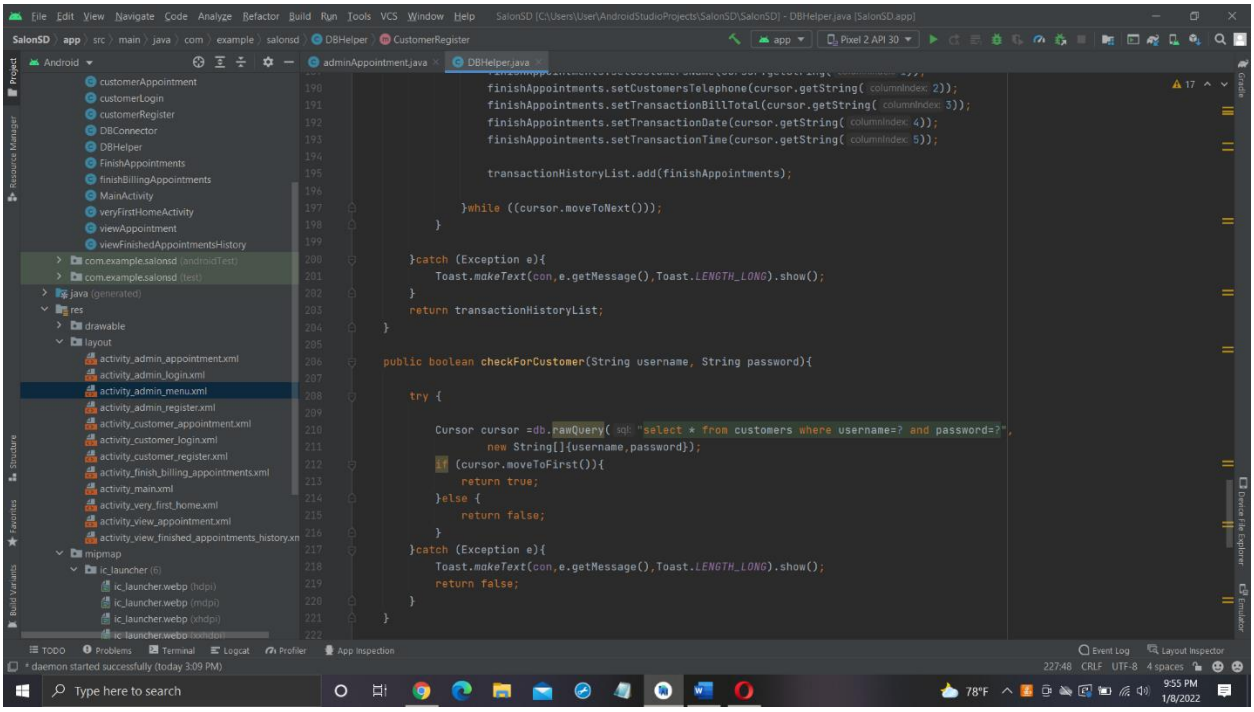
2.9 - Customer Login Interface



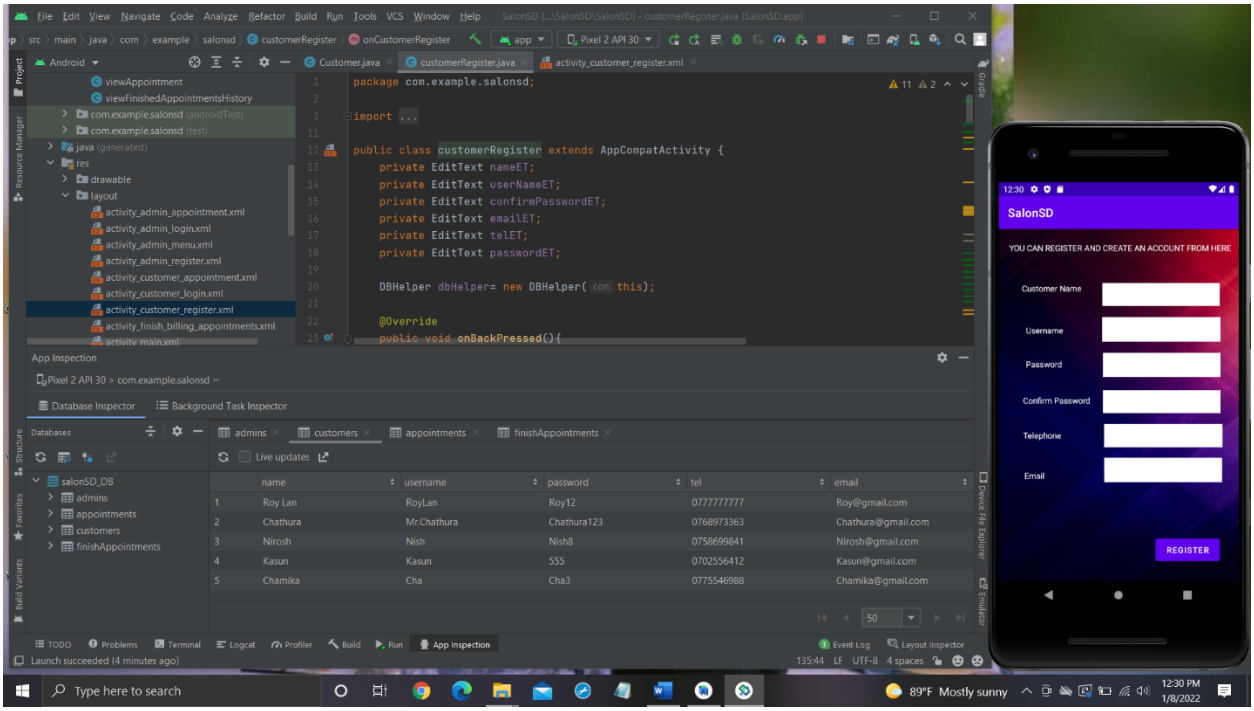
The interface shown in the attached screenshot can be seen by only a user who click on the customer button from the very first loaded interface after running the system. This Login interface is only for the Customers who wants to make an appointment. This UI consists with two buttons, two plain texts and three text views along with a background image. From the back-end, inside the class called customerLogin, I have created two on clicks as one in register button in which background is transparent and intent it to the customer register interface and in the login button and intent it to the customer appointment interface where a customer can make an appointment and view booked appointments before making an appointment. Here, when a customer sees this interface, he/she has the ability to move to the customer register interface if customer hasn't registered. If a customer processes an already created user account, he/she is eligible to log in. The only difference in this customer login interface and admin interface is customer login has the ability to move to the register interface without being logged in whereas admin has that register functionality inside the admin menu. The validations

used for this customer login interfaces include are empty field validation and correct username/password validation. Further, toast messages have used to show the messages inside the validations. All those coding has done inside the on click of the login button.

The below screenshot shows the database connection that is used to check the correctness of a customer's username and password. Select query has been used for that.

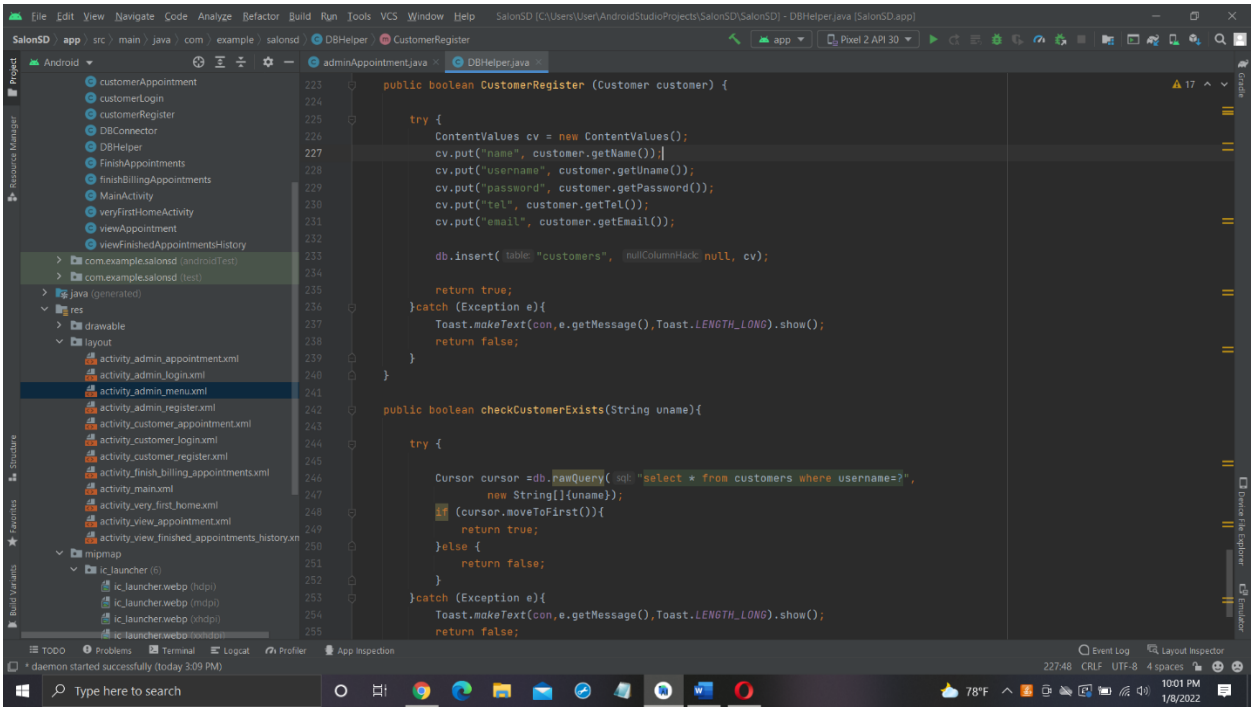


2.10 - Customer Registration Interface

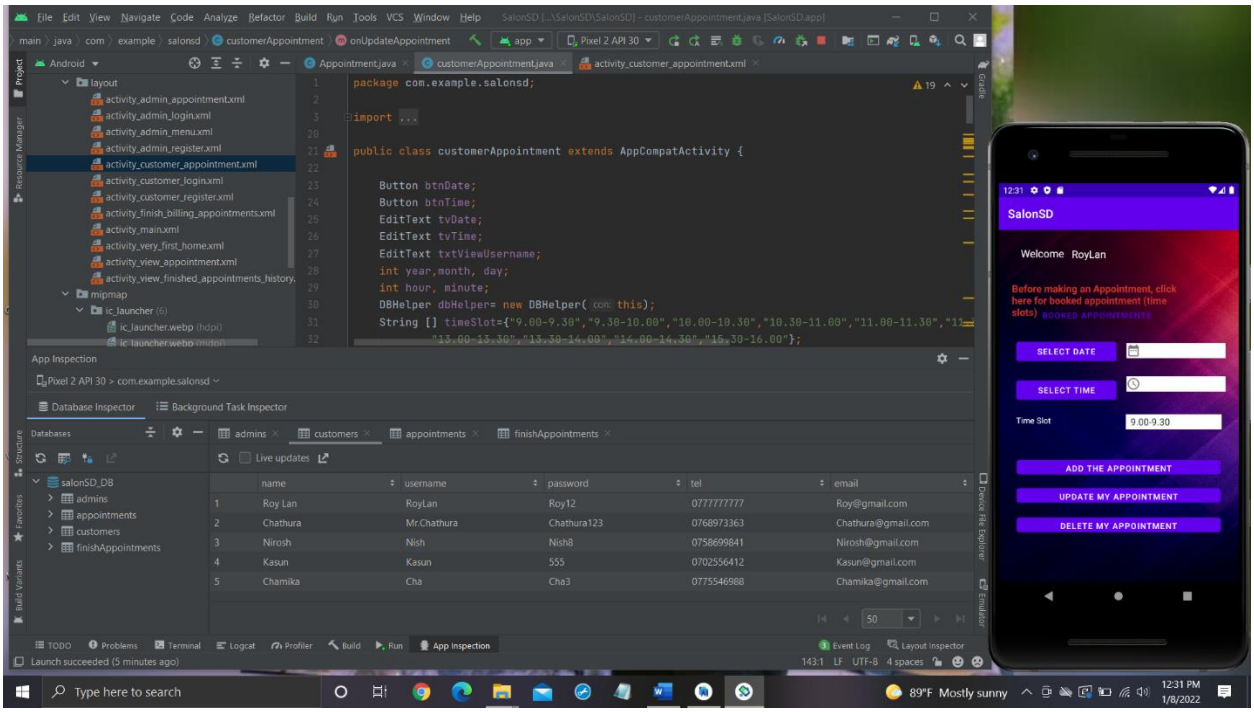


The interface shown in the attached screenshot will be appeared only when customer clicks Register button from the Customer Login interface. This UI consists with one button, six plain texts and seven text views along with a background image. From the back-end, inside the class called customerRegister, I have created all the validations for the customer registration. The validations used for this customer registration include are empty field validation, password and confirm password matching validation, email validation and username validation which displays the message as a user already exists or not. Further, toast messages have used to show the messages inside the validations. All those coding has done inside the on click of the register button.

The below screenshot shows the methods and database connections that are used to check for existing users where select query has been used and the query that has used for inserting data obtain from the customer register interface.



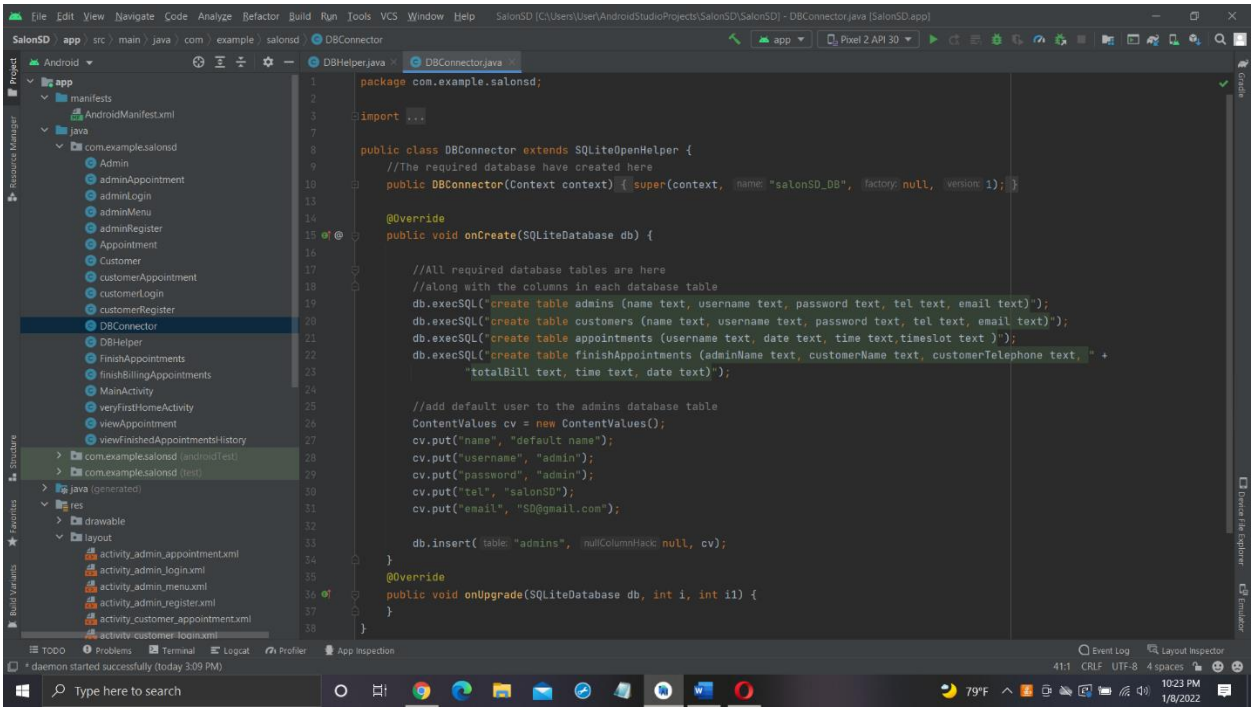
2.11 - Customer Add Appointments Interface



The interface shown in the attached screenshot will be appeared only when a customer successfully logged in. This UI consists with six buttons as one to View Appointment, one to select date, one to select time, one to Add Appointments, one to Update Appointment and the other to Delete Appointment, four plain texts and four text views along with a background image. In here, a plain text is used to display logged customer's username. There is a date and time picker to get the current date and the time. Time slot can be selected only after viewing the booked appointments from the background transparent button called Booked Appointments. To select the time slot, a spinner has used. From the back-end, inside the class called customerAppointment, I have created on click listeners for date and time picker and four on clicks have been made separately for four buttons called Add, Update, Delete, Booked Appointments and validations have included inside them. The validations used for this customer appointment include are empty field validation and successfully inserted or not validation. Further, toast messages have used to show the messages inside the validations.

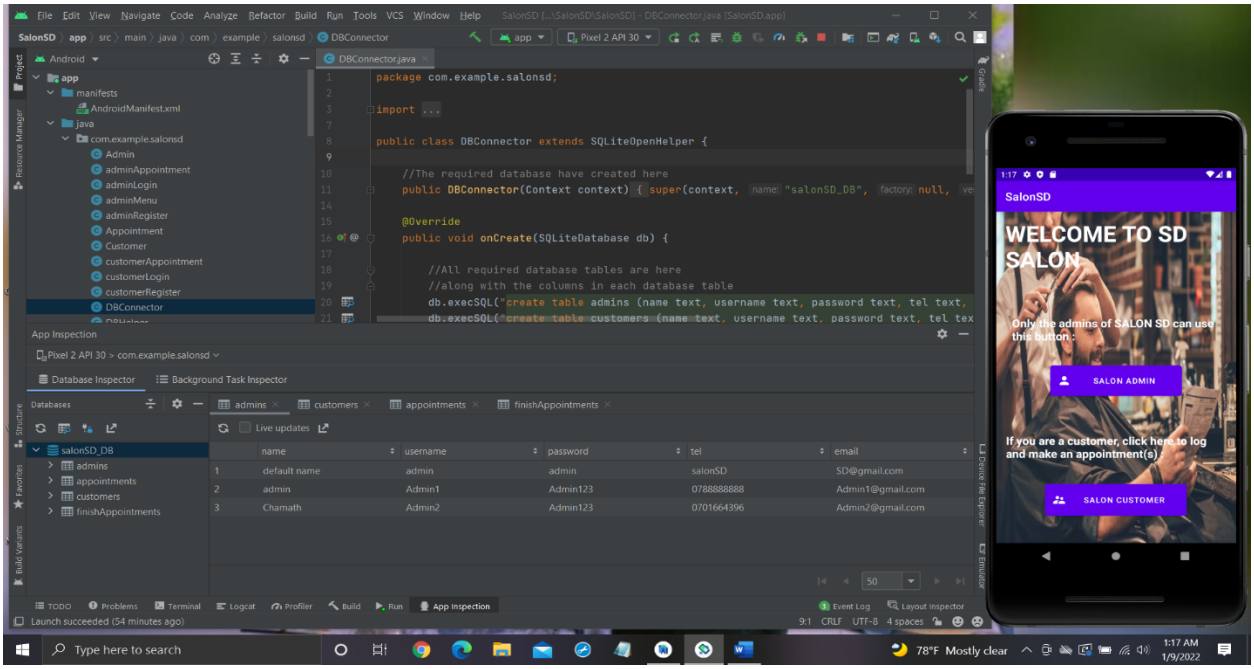
Another specialty in customer appointment is it uses the same methods and database connections which were used to insert, update and delete appointment details by the admin. Only the interfaces are changed and intent.

2.12 - Database Structure



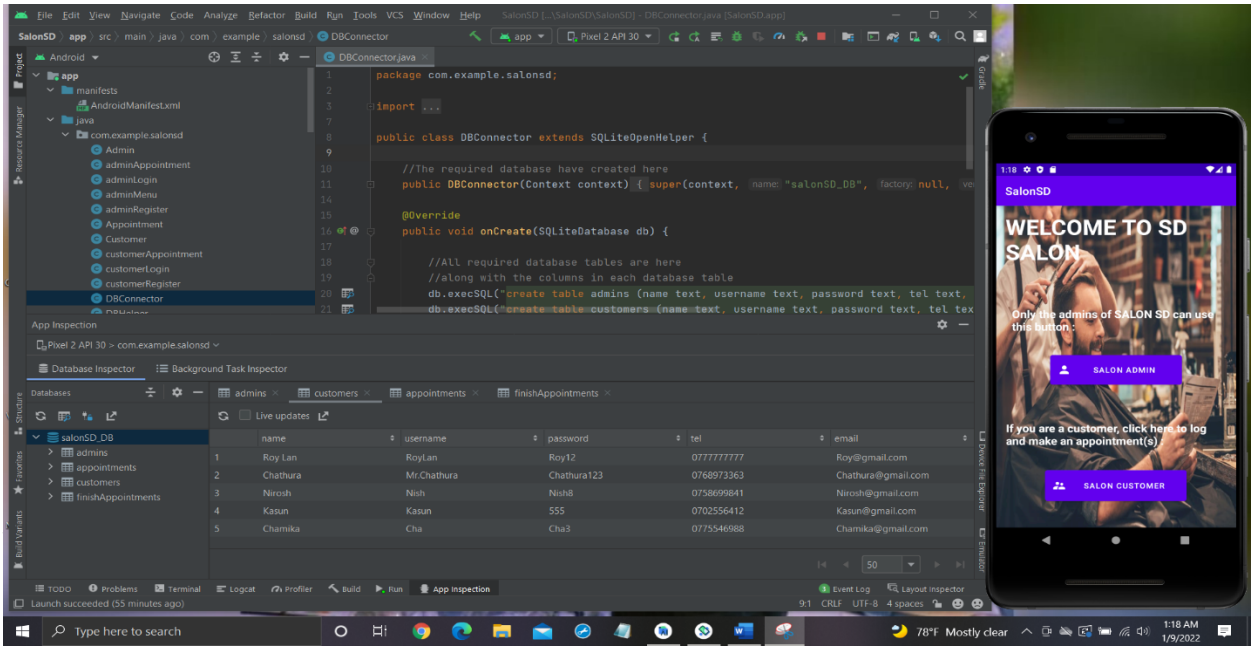
In the above screenshot, it depicts how the coding has been done to make a database and database table along with columns. All those coding has been coded inside an activity called DBConnector. Its visible in the screenshot that the database name is coded as salonSD_DB. Moreover, there are four database tables named as admins, customers, appointments and finishAppointments. The admins database table consists with the columns called name, username, password, telephone and email. The customers database table consists with the columns called name, username, password, telephone and email. The appointments database table consists with the columns called username, date, time and time slot. The finishAppointments database table consists with the columns called admin name, customer name, customer telephone, total bill, time and date.

2.12.1 - Admins Database Table

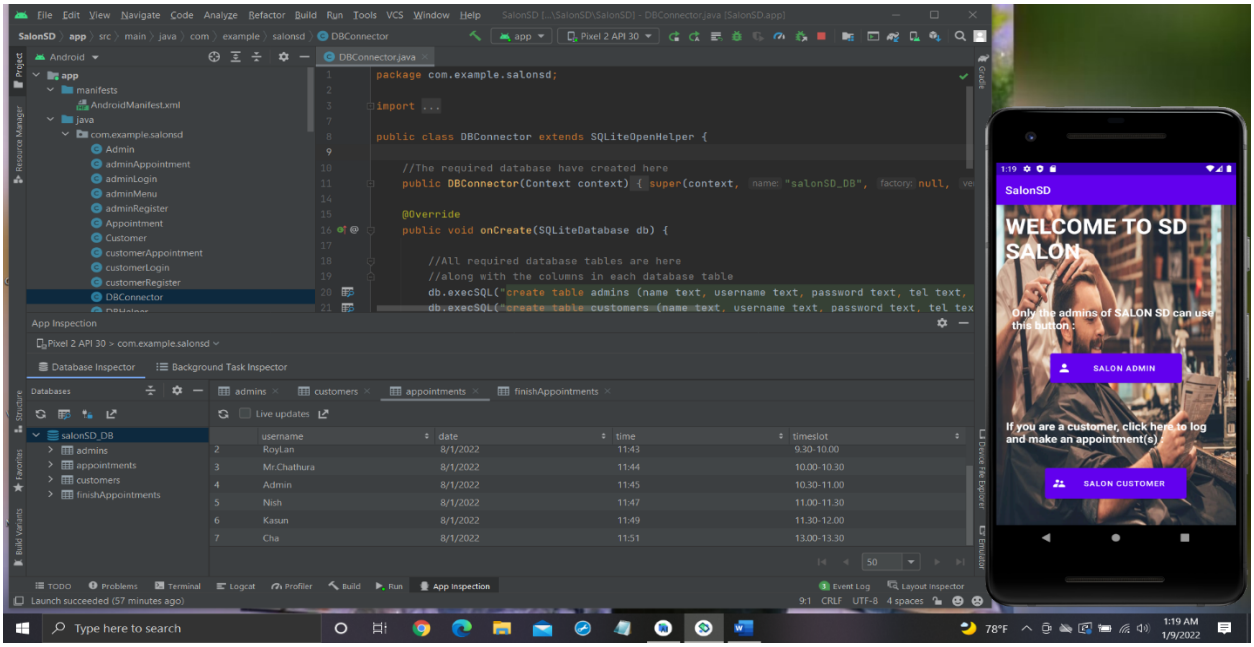


This above screenshot depicts the graphical representation of the database table which has named as “admins”. Further, it shows three records inside the table as table rows.

2.12.2 - Customers Database Table

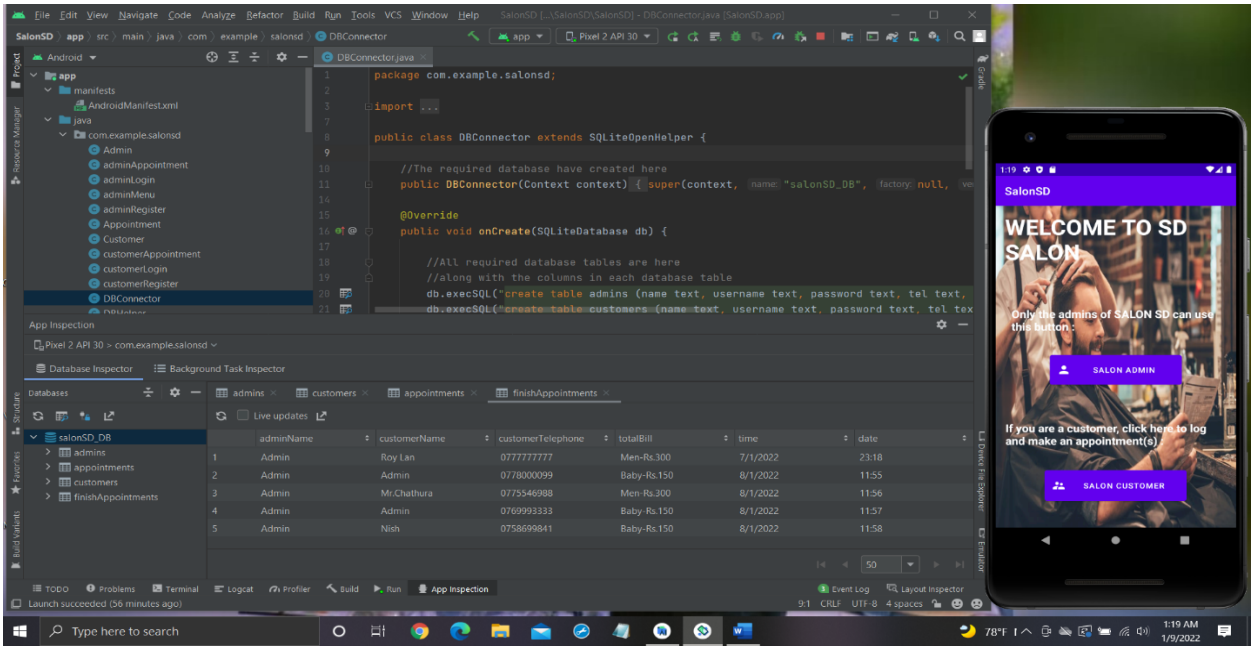


2.12.3 - Appointments Database Table



This above screenshot depicts the graphical representation of the database table which has named as “appointments”. Further, it shows seven records inside the table as table rows.

2.12.4 - Finish Appointments Database Table



This above screenshot depicts the graphical representation of the database table which has named as “finishAppointments”. Further, it shows five records inside the table as table rows.

Task 3 - Testing

Testing a system involves in testing the existing overall system. All most all the functions are linked together to see if the system performs as planned. The main purpose of system testing is to produce a high-quality system with well-functioning outputs.

Therefore, Black-Box system testing method will be used for this Mobile Application of Salon SD. Because any internal knowledge about the coding is not required for Black-Box testing whereas internal knowledge about the coding is required for White-Box testing. System testing covers both functional and non-functional functions and many other aspects of testing. Thus, they are tested using a Black-Box approach as its easier just to submit the input into the system and confirm the result.

Many terminologies are used to describe software quality assurance through system testing. People who are specialized in this industry uses various software terminologies for software quality assurance. Among those terminologies, Test Plans and Test Cases are widely used for system testing in order to assure the quality of a software. (Dondeti, 2012)

Thus, below shows the Test Plan and the Test Cases that are drawn for testing the Mobile Application of the Salon SD which plays a key part in providing a high-quality product. There I've used the Android Virtual Device called Pixel 2 API 30 which runs the Android 11.0 and the resolution is 1080*1920: 420 dpi.

3.1 - Test Plan

TEST CASE ID	TEST CASE NAME	DESCRIPTION	EXPECTED RESULT
TC1	Admin Login		
TC1.1	Check Login empty fields	Have to press on login button without filling any field	Should display an Error Message as Fields can't be blank
TC1.2	Check fields by entering values one by one into fields		
TC1.2.1	Check by filling only username field	Have enter a username inside the username field and press login button.	Should display the same Error Message which displayed previously as Fields can't be blank
TC1.2.2	Check by filling only password field	Have to enter only a password and press login button separately	Should display the same Error Message which displayed previously as Fields can't be blank
TC1.3	Check by wrong username or password		
TC1.3.1	Check with wrong username and correct password	Have to enter a wrong username along with the correct password and press login button	Should display an Error Message as Please enter valid username & password
TC1.3.2	Check with wrong password and correct username	Have to enter a wrong password along with the correct username and press login button	Should display an Error Message as Please enter valid username & password
TC1.4	Check Login working without any empty field and with default (hard coded) login credentials	Have to enter a default username(admin) and password(admin) correctly	Admin menu interface should be loaded along with a message as Login granted successfully
TC1.5	Check Login working without any empty field and with saved &	Have to enter the admin registered and saved username and password correctly	Admin menu interface should be loaded along with a message as Login granted successfully

	correct login credentials		
TC2	Admin Registration		
TC2.1	Check Register empty fields	Have to press on Register button without filling any fields	Should display an Error Message as Fields can't be blank
TC2.2	Check by entering values one by one into fields		
TC2.2.1	Check by entering only admin name	Have to enter an admin name inside the username field and press register button	Should display an Error Message as Fields can't be blank
TC2.2.2	Check by entering only username	Have to enter a username inside the username field and press register button	Should display an Error Message as Fields can't be blank
TC2.2.3	Check by entering only password	Have to enter only a password and press register button	Should display an Error Message as Fields can't be blank
TC2.2.4	Check by entering only confirm password	Have to enter a confirm password and press register button separately	Should display an Error Message as Fields can't be blank
TC2.2.5	Check by entering only telephone	Have to enter a telephone and press register button separately	Should display an Error Message as Fields can't be blank
TC2.2.6	Check by entering only email	Have to enter an email and press register button separately	Should display an Error Message as Fields can't be blank
TC2.3	Check by entering already saved username into the username field	Have to fill all other fields, enter an already registered admin's username and press register button	Should display an Error Message as Entered username already exists
TC2.4	Check password and confirm password matching	Have to fill all other fields, enter a password and a different confirm password and press register button	Should display an Error Message as Password & confirm password should match
TC2.5	Check email format	Have to fill all other fields correctly, enter an email address with the @ symbol and press register button	Should display an Error Message as Invalid Email format

TC2.6	Check Registration working without any empty field and with new correct data	Have to enter a new username, matching password with confirm password, correct email format, without empty fields and press register button	Should display a message as User account created successfully
TC2.7	Check on back when some fields are filled		
TC2.7.1	Check back button notification when some fields are filled	Fill only few fields along with some empty fields and press on back button of the phone	A notification should pop up and ask whether user wants to move back and get user's response.
TC2.7.2	Check back button notification when user's response is YES	Press on YES	If user provided YES, the previous interface will be displayed.
TC2.7.3	Check back button notification when user's response is NO	Press on NO	If user provided NO, that notification will fade away and a message will pop up mentioning that Continue Editing
TC3	Admin Appointment		
TC3.1	Check Add Appointment with empty fields	Have to press on Add Appointment button without filling any fields	Should display an Error Message as Fields can't be blank
TC3.2	Check Add Appointment working without any empty field	Have to fill all fields and press on Add Appointment button	Should display a message as Appointment added successfully
TC3.3	Check Update Appointment working without any empty and updated fields	Have to change the necessary fields without any empty field by entering the update needed user's username on the user type field and press on Update Appointment button	Should display a message as Appointment updated successfully

TC3.4	Check Delete Appointment working	Press on Delete Appointment button by entering the delete needed user's username on the user type field and press on Delete Appointment button	Should display a message as Appointment deleted successfully
TC4	Finish Billing Appointments		
TC4.1	Check Finish Appointment with empty fields	Have to press on Finish Appointment button without filling any fields	Should display an Error Message as Fields can't be blank
TC4.2	Check Finish Appointment working without any empty field	Have to fill all fields and press on Finish Appointment button	Should display a message as Appointment Finished successfully
TC5	Customer Registration		
TC5.1	Check Register empty fields	Have to press on Register button without filling any fields	Should display an Error Message as Fields can't be blank
TC5.2	Check by entering values one by one into fields		
TC5.2.1	Check by entering only customer name	Have to enter a customer name inside the username field and press register button	Should display an Error Message as Fields can't be blank
TC5.2.2	Check by entering only username	Have to enter a username inside the username field and press register button	Should display an Error Message as Fields can't be blank
TC5.2.3	Check by entering only password	Have to enter only a password and press register button	Should display an Error Message as Fields can't be blank
TC5.2.4	Check by entering only confirm password	Have to enter a confirm password and press register button separately	Should display an Error Message as Fields can't be blank
TC5.2.5	Check by entering only telephone	Have to enter a telephone and press register button separately	Should display an Error Message as Fields can't be blank

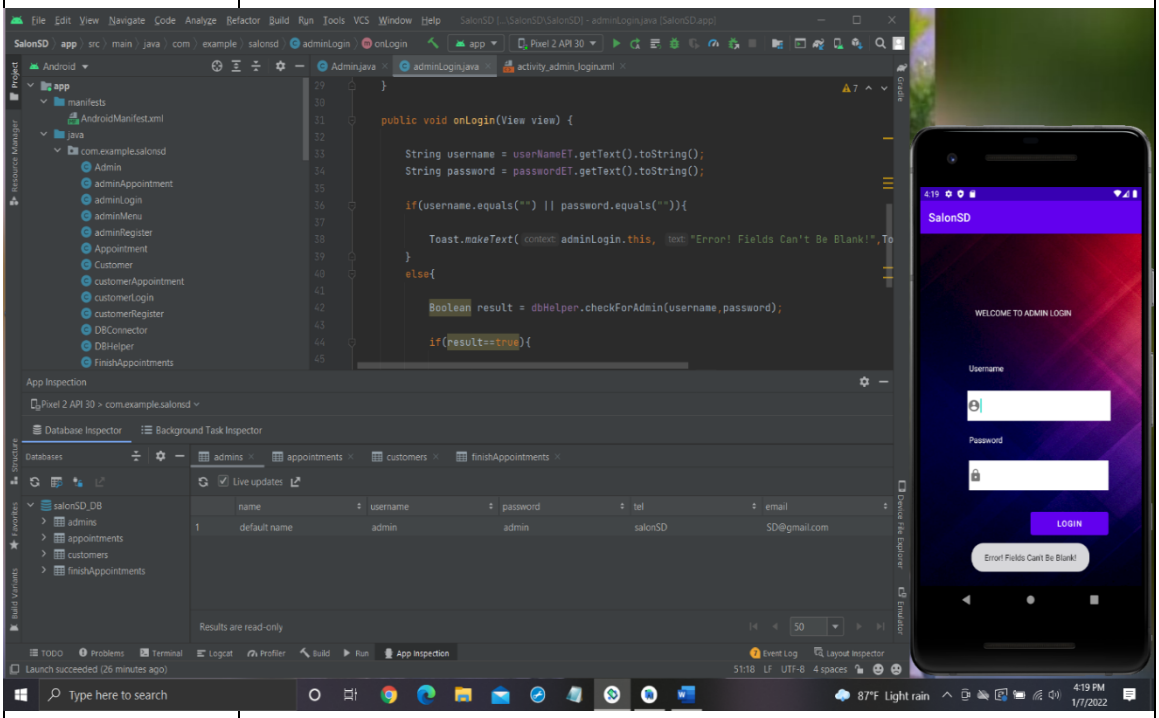
TC5.2.6	Check by entering only email	Have to enter an email and press register button separately	Should display an Error Message as Fields can't be blank
TC5.3	Check by entering already saved username into the username field	Have to fill all other fields, enter an already registered customer's username and press register button	Should display an Error Message as Entered username already exists
TC5.4	Check password and confirm password matching	Have to fill all other fields, enter a password and a different confirm password and press register button	Should display an Error Message as Password & confirm password should match
TC5.5	Check email format	Have to fill all other fields correctly, enter an email address with the @ symbol and press register button	Should display an Error Message as Invalid Email format
TC5.6	Check Registration working without any empty field and with new correct data	Have to enter a new username, matching password with confirm password, correct email format, without empty fields and press register button	Should display a message as User account created successfully
TC5.7	Check on back when some fields are filled		
TC5.7.1	Check back button notification when some fields are filled	Fill only few fields along with some empty fields and press on back button of the phone	A notification should pop up and ask whether user wants to move back and get user's response.
TC5.7.2	Check back button notification when user's response is YES	Press on YES	If user provided YES, the previous interface will be displayed.
TC5.7.3	Check back button notification when user's response is NO	Press on NO	If user provided NO, that notification will fade away and a message will pop up mentioning that Continue Editing

TC6	Customer Login		
TC6.1	Check Login empty fields	Have to press on login button without filling any field	Should display an Error Message as Fields can't be blank
TC6.2	Check fields by entering values one by one into fields		
TC6.2.1	Check by filling only username field	Have enter a username inside the username field and press login button.	Should display the same Error Message which displayed previously as Fields can't be blank
TC6.2.2	Check by filling only password field	Have to enter only a password and press login button separately	Should display the same Error Message which displayed previously as Fields can't be blank
TC6.3	Check by wrong username or password		
TC6.3.1	Check with wrong username and correct password	Have to enter a wrong username along with the correct password and press login button	Should display an Error Message as Please enter valid username & password
TC6.3.2	Check with correct username and wrong password	Have to enter a wrong password along with the correct username and press login button	Should display an Error Message as Please enter valid username & password
TC6.4	Check Login working without any empty field and with saved & correct login credentials	Have to enter the customer registered and saved username and password correctly	Customer Appointment interface should be loaded along with a message as Login granted successfully
TC7	Customer Appointment		
TC7.1	Check Add Appointment with empty fields	Have to press on Add Appointment button without filling any fields	Should display an Error Message as Fields can't be blank
TC7.2	Check Add Appointment working without any empty field	Have to fill all fields and press on Add Appointment button	Should display a message as Appointment added successfully

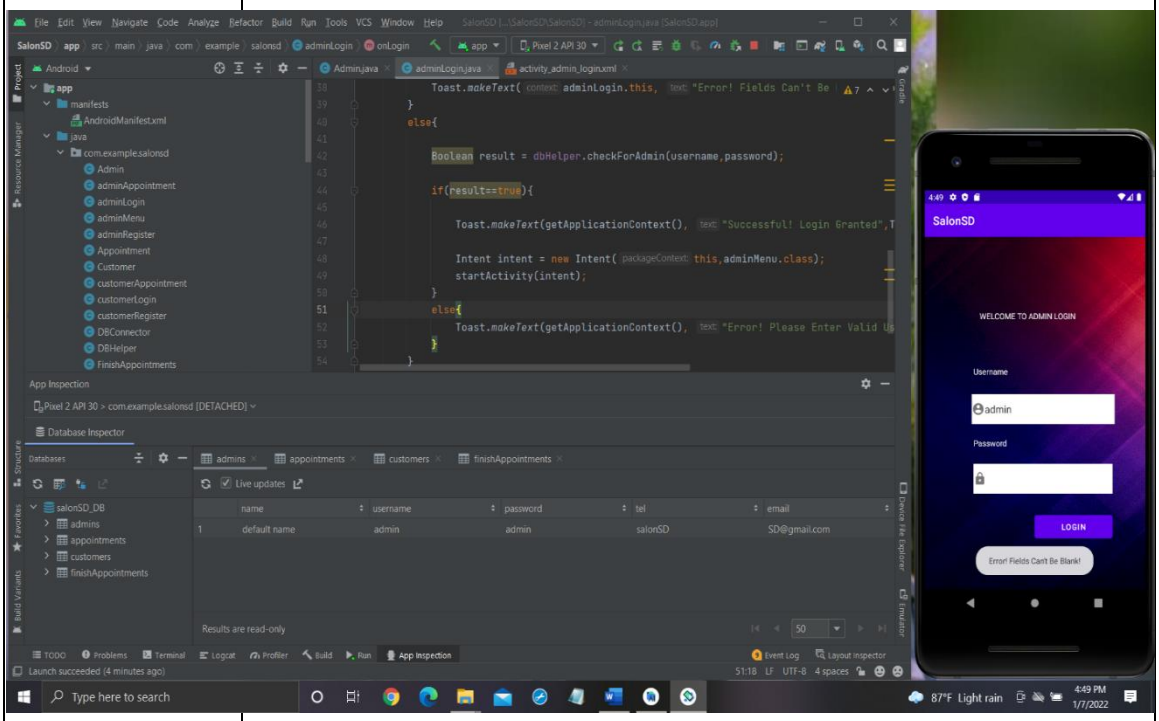
TC7.3	Check Update Appointment working without any empty and updated fields	Have to change the necessary fields without any empty field and press on Update Appointment button	Should display a message as Appointment updated successfully
TC7.4	Check Delete Appointment working	Press on Delete Appointment button where particular logged user’s username is displayed at the right top of the interface	Should display a message as Appointment deleted successfully

3.2 - Test Cases

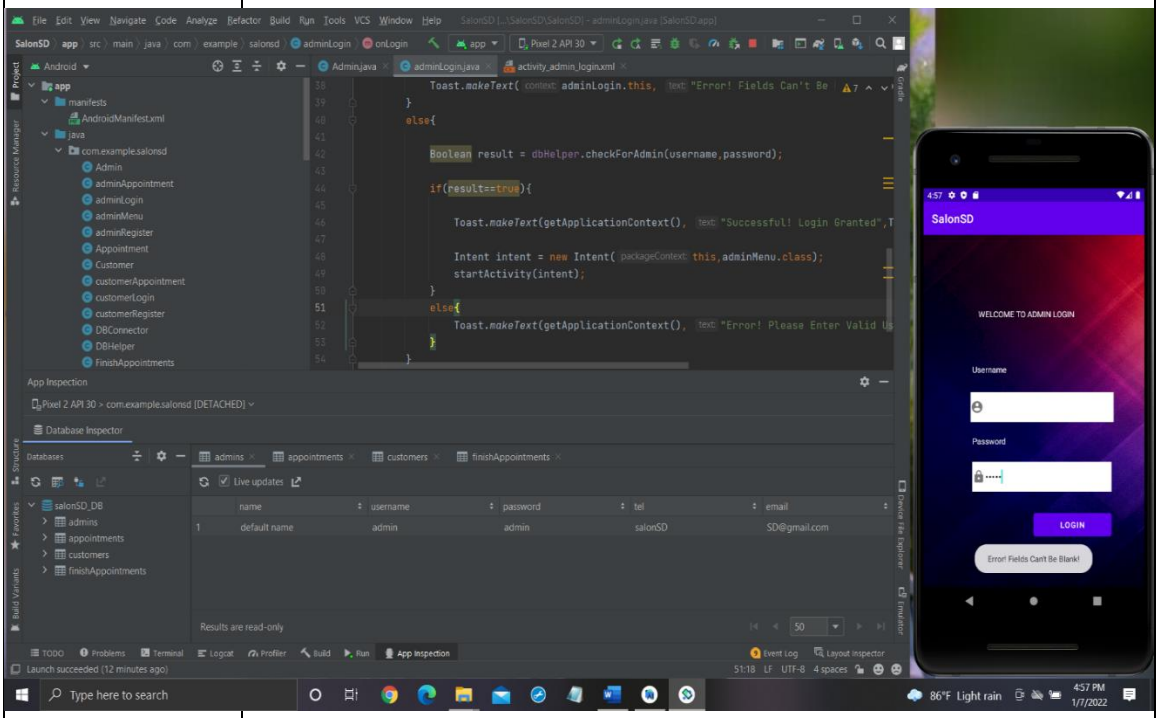
TC1 - Admin Login

TEST CASE ID	TC1.1
TEST CASE NAME	Check Login empty fields
TEST SUMMARY	Checking the empty fields validation in Admin Login interface
TEST STEPS	Press on login button (without filling any field)
TEST DATA	Empty fields
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

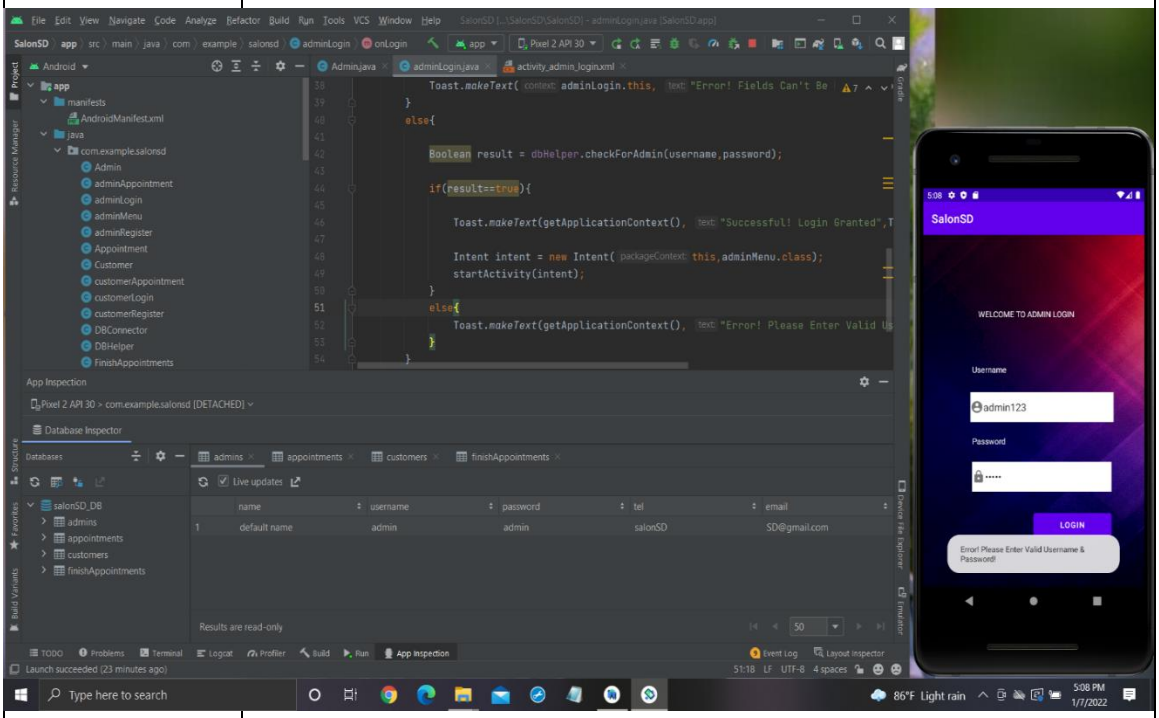
TC1.2 - Check fields by entering values one by one into fields

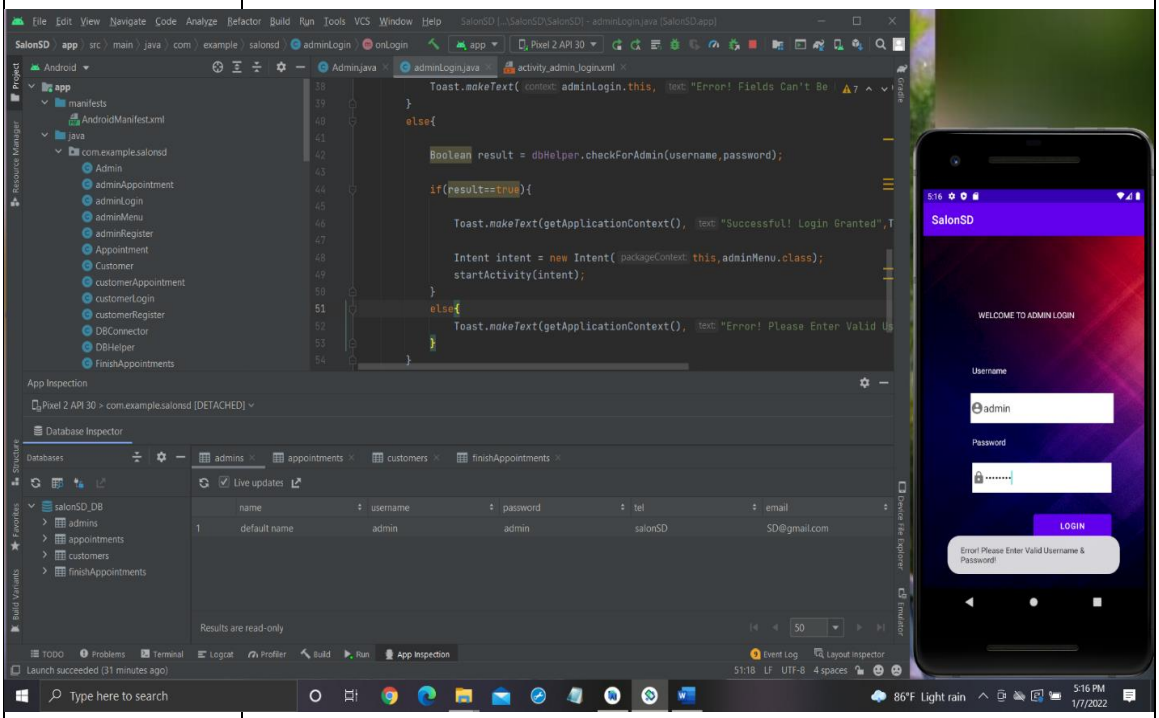
TEST CASE ID	TC1.2.1
TEST CASE NAME	Check by filling only username field
TEST SUMMARY	Checking empty field validation by filling only one field
TEST STEPS	1. Fill only username field with a valid admin username 2. Press on Login button
TEST DATA	Username: admin Password: empty
EXPECTED RESULT	Should display the same Error Message which displayed previously as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages displayed as a toast message with very clear outcomes and the same error displayed in the above test case has received here also.
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC1.2.2
TEST CASE NAME	Check by filling only password field
TEST SUMMARY	Checking empty field validation by filling only one field
TEST STEPS	1. Fill only password field with a valid admin password 2. Press on Login button
TEST DATA	Username: empty Password: admin
EXPECTED RESULT	Should display the same Error Message which displayed previously as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages displayed as a toast message with very clear outcomes and the same error displayed in the above test case has received here also.
STATUS (PASS/FAIL)	Pass

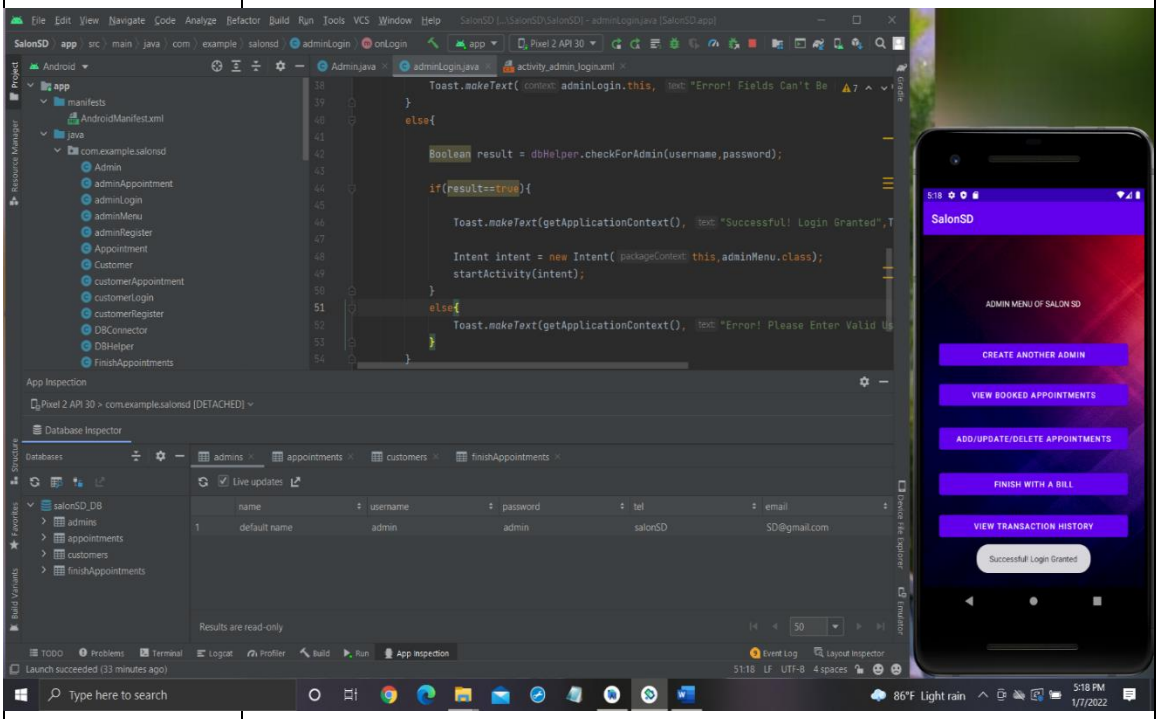


TC1.3 - Check by wrong username or password

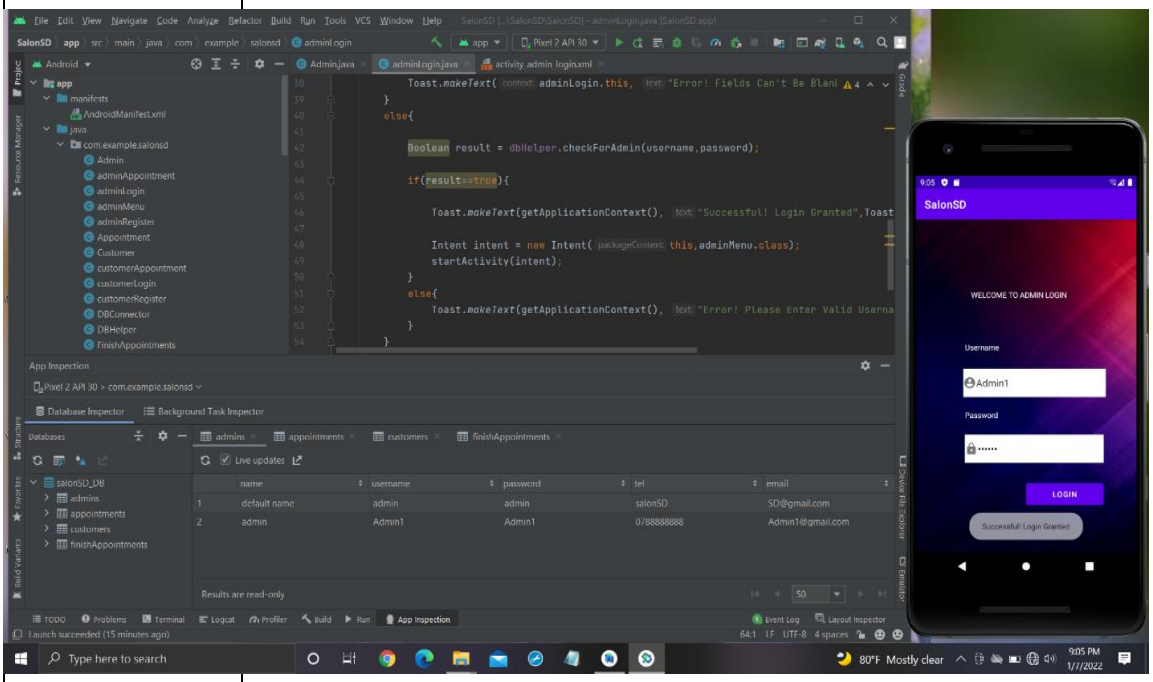
TEST CASE ID	TC1.3.1
TEST CASE NAME	Check with wrong username and correct password
TEST SUMMARY	Checking correctly filled password field validation whereas username is wrong
TEST STEPS	1. Fill only password field with a valid admin password 2. Fill username field with a wrong username 3. Press on Login button
TEST DATA	Username: admin123 Password: admin
EXPECTED RESULT	Should display an Error Message as Please enter valid username & password
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

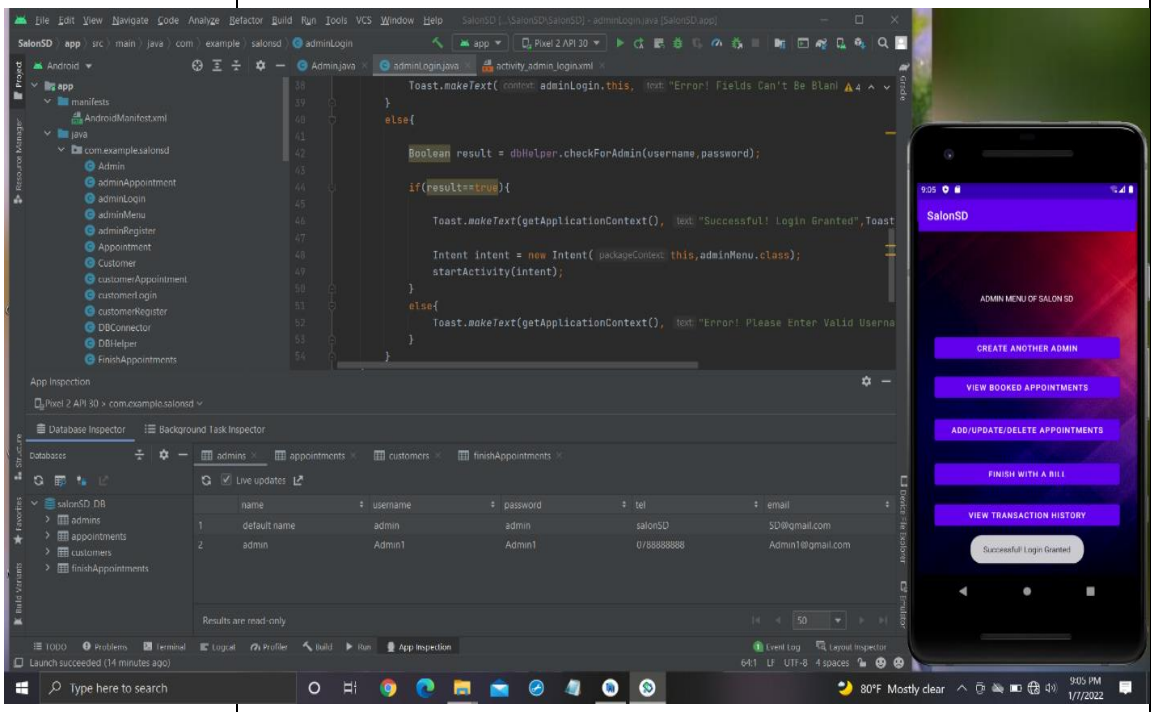
TEST CASE ID	TC1.3.2
TEST CASE NAME	Check with correct username and wrong password
TEST SUMMARY	Checking correctly filled username field validation whereas password is wrong
TEST STEPS	1. Fill only username field with a valid admin username 2. Fill password field with a wrong password 3. Press on Login button
TEST DATA	Username: admin Password: admin123
EXPECTED RESULT	Should display an Error Message as Please enter valid username & password
ACTUAL RESULT	
CONCLUSION	The expected error messages displayed as a toast message with very clear outcomes and the same error displayed in the above test case has received here also.
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC1.4
TEST CASE NAME	Check Login working without any empty field and with default (hard coded) login credentials
TEST SUMMARY	Checking whether correctly filled username and password is working or not
TEST STEPS	1. Fill username field with a valid(hardcoded) admin username 2. Fill password field with a valid(hardcoded) password 3. Press on Login button
TEST DATA	Username: admin Password: admin
EXPECTED RESULT	Admin menu interface should be loaded along with a message as Login granted successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

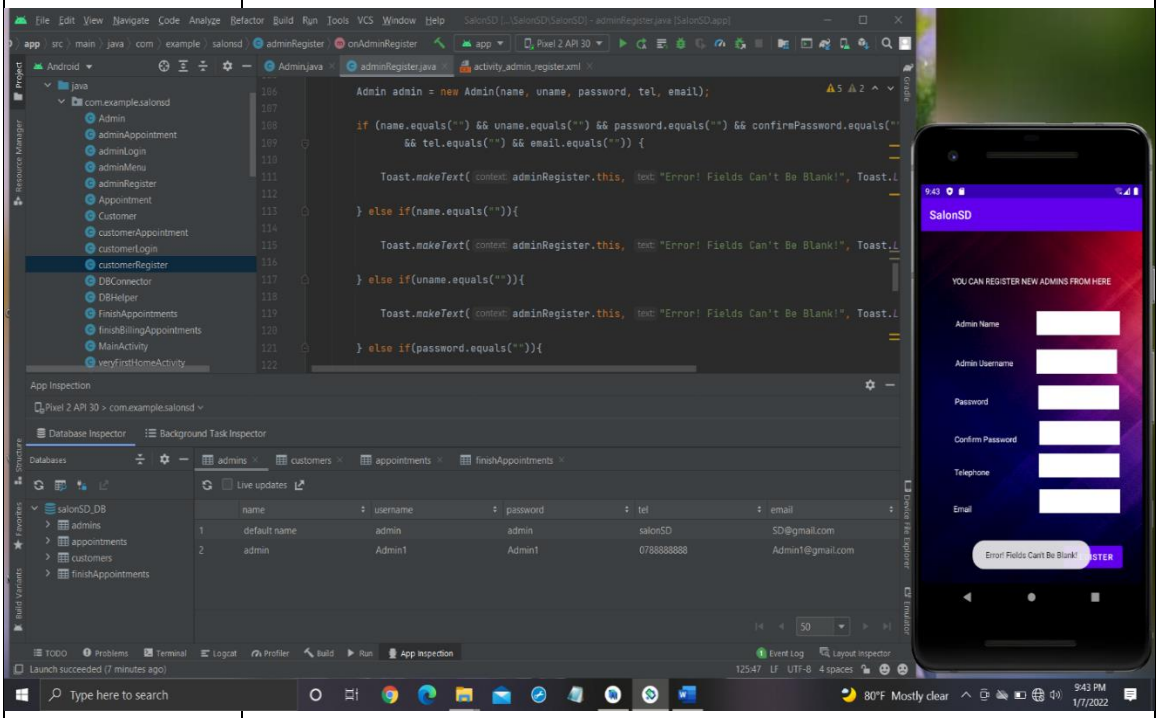


TEST CASE ID	TC1.5
TEST CASE NAME	Check Login working without any empty field and with saved & correct login credentials
TEST SUMMARY	Checking whether correctly filled username and password is working or not using newly registered username and password
TEST STEPS	1. Fill username field with a valid newly registered admin username 2. Fill password field with a valid newly registered password 3. Press on Login button
TEST DATA	Username: Admin1 Password: Admin123
EXPECTED RESULT	Admin menu interface should be loaded along with a message as Login granted successfully
ACTUAL RESULT	

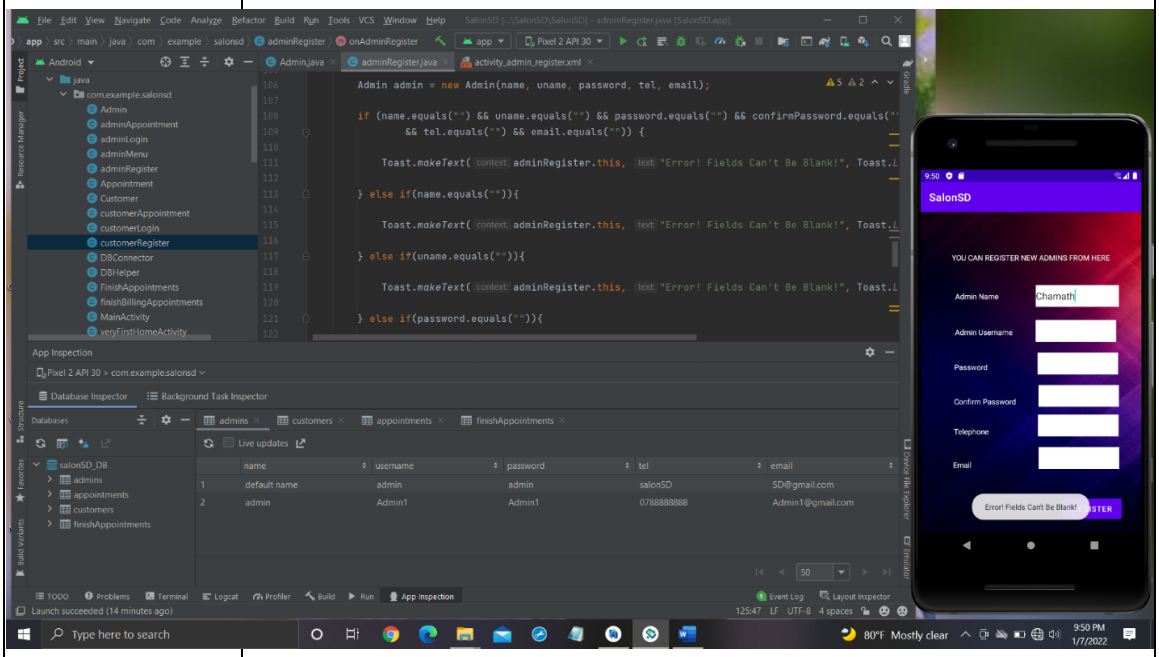


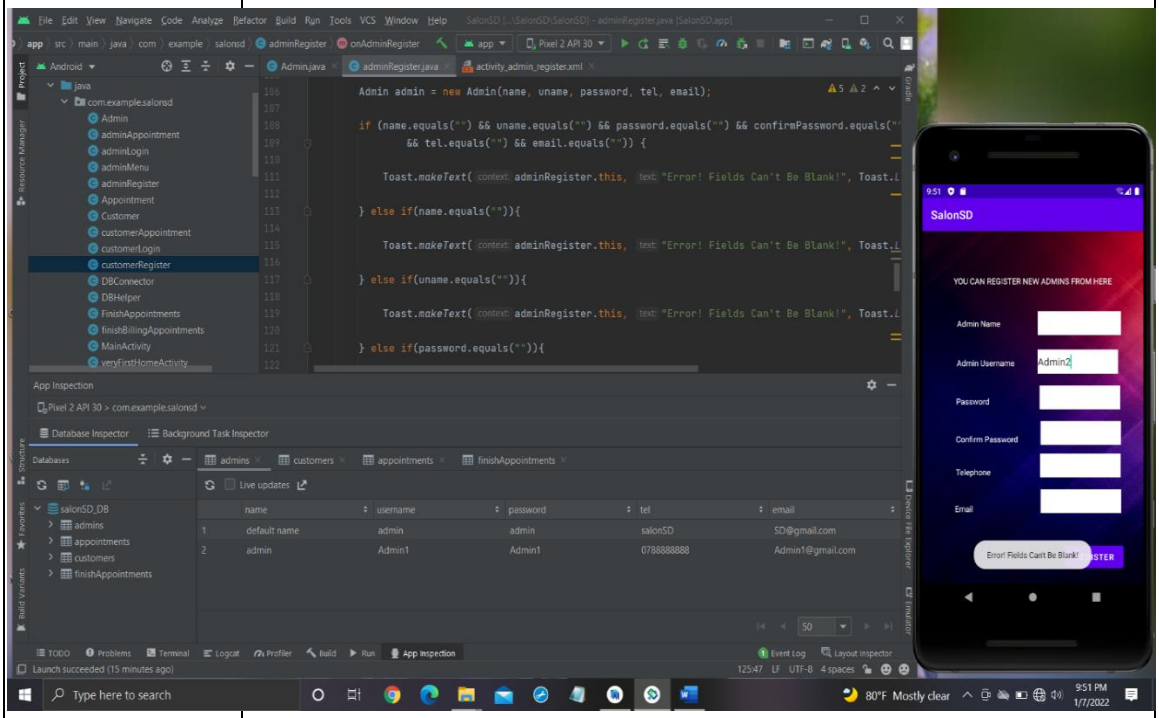
	
CONCLUSION	The expected login successful message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

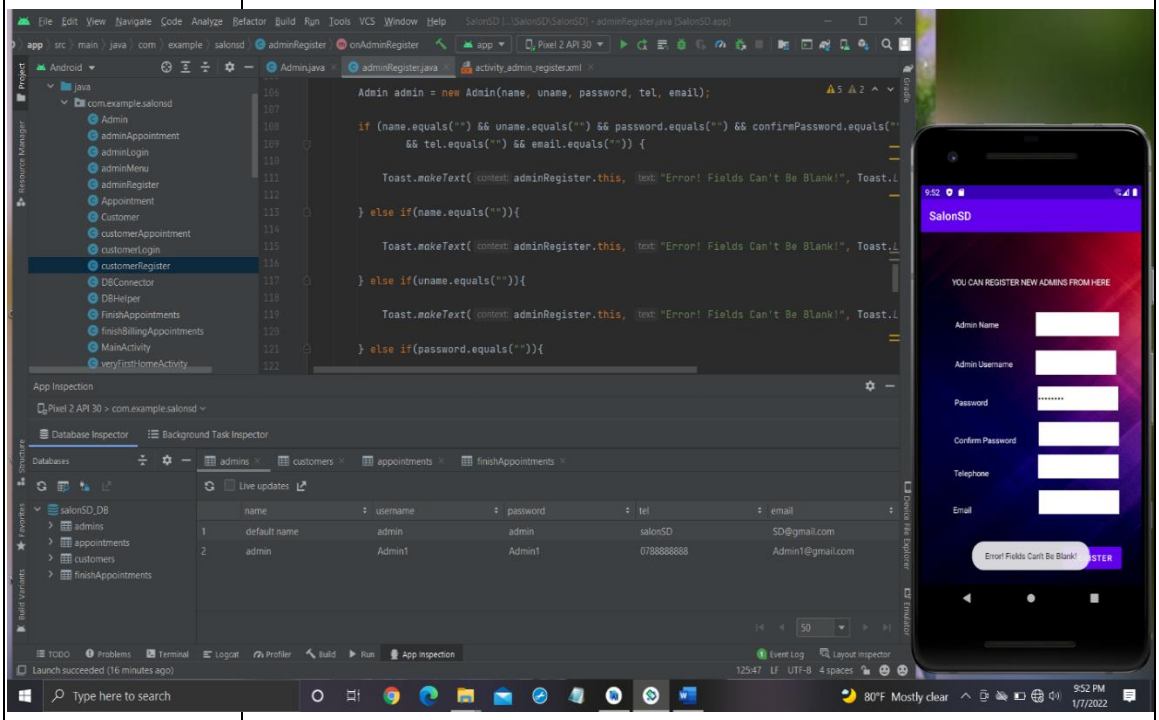
TC2 - Admin Registration

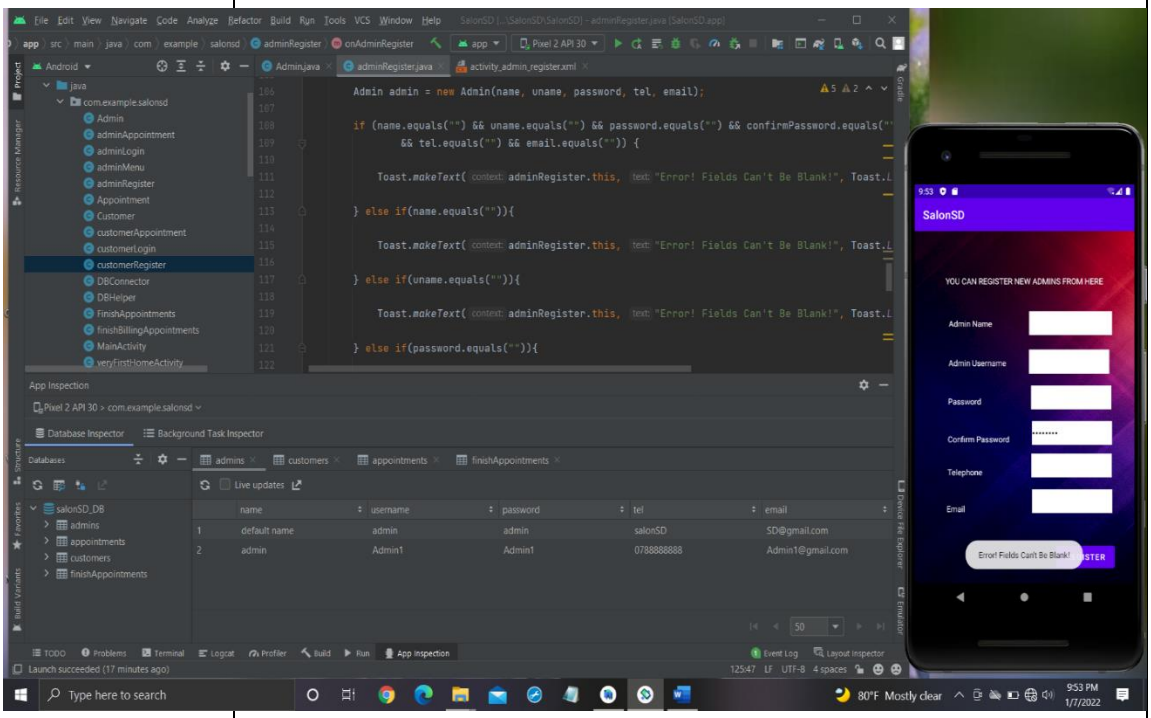
TEST CASE ID	TC2.1
TEST CASE NAME	Check Register empty fields
TEST SUMMARY	Checking the empty fields validation in Admin Register interface
TEST STEPS	Press on register button (without filling any field)
TEST DATA	Empty fields
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

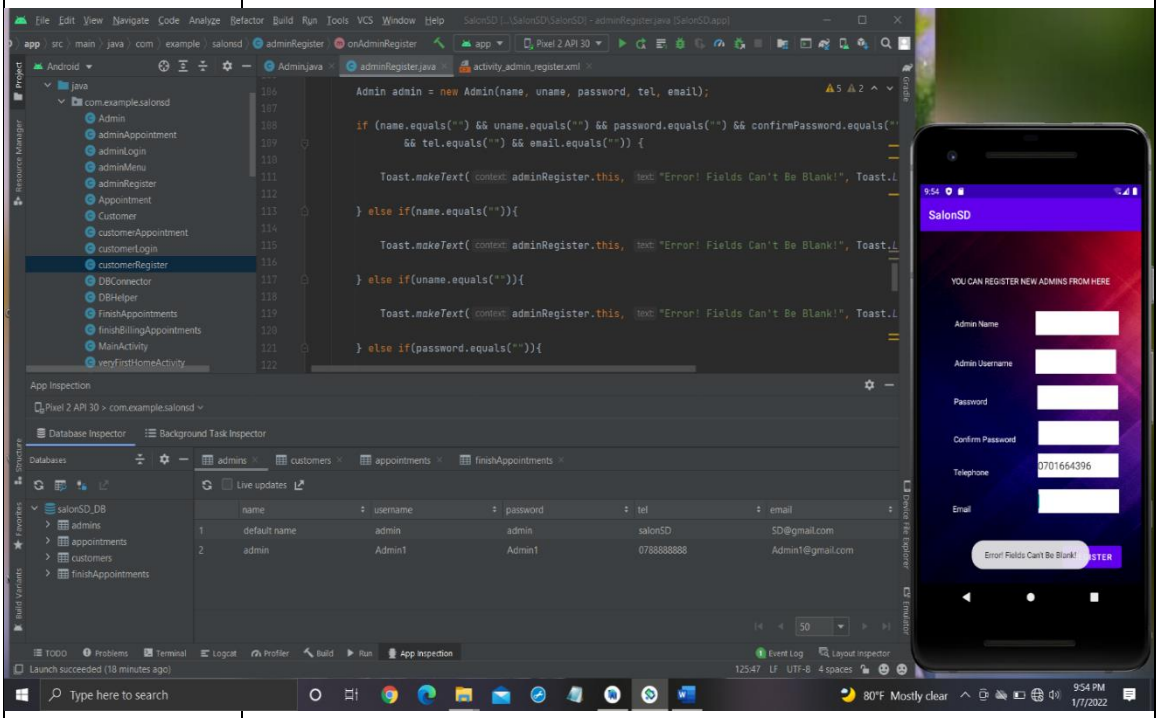
TC2.2 - Check by entering values one by one into fields

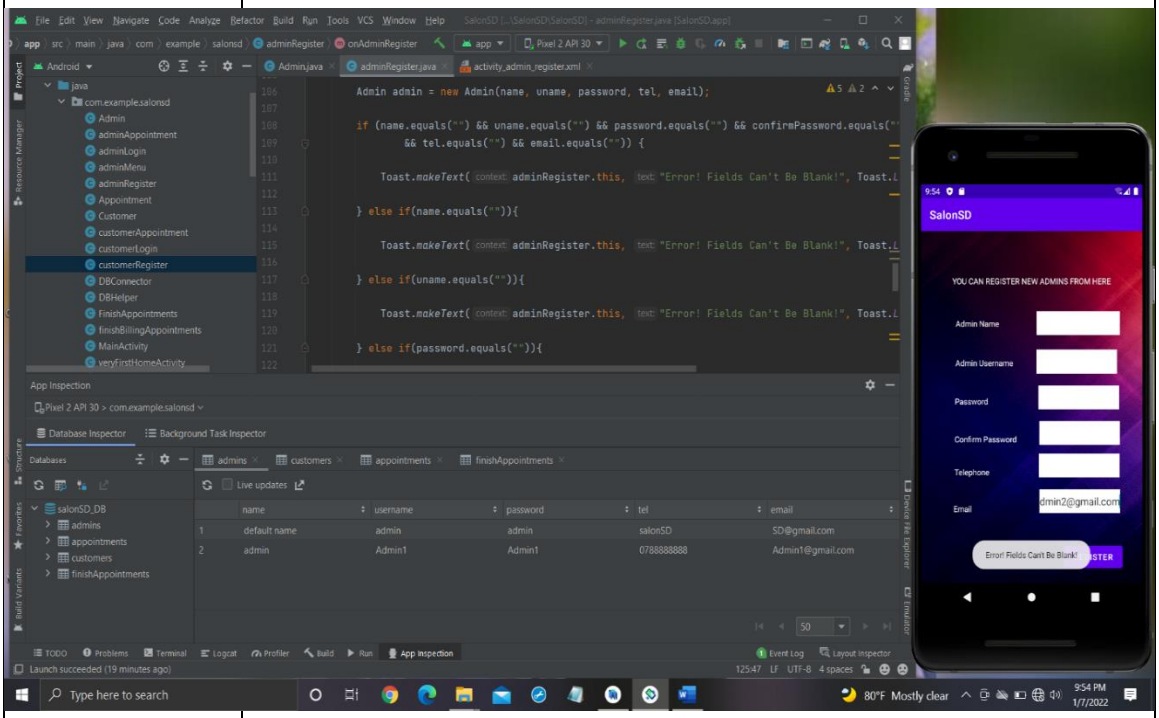
TEST CASE ID	TC2.2.1
TEST CASE NAME	Check by entering only admin name
TEST SUMMARY	Checking the empty fields validation in Admin Register by filling only admin name
TEST STEPS	1. Fill only admin name field 2. Press on register button
TEST DATA	Admin name: Chamath
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC2.2.2
TEST CASE NAME	Check by entering only username
TEST SUMMARY	Checking the empty fields validation in Admin Register by filling only username
TEST STEPS	1. Fill only username field 2. Press on register button
TEST DATA	Username: Admin2
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

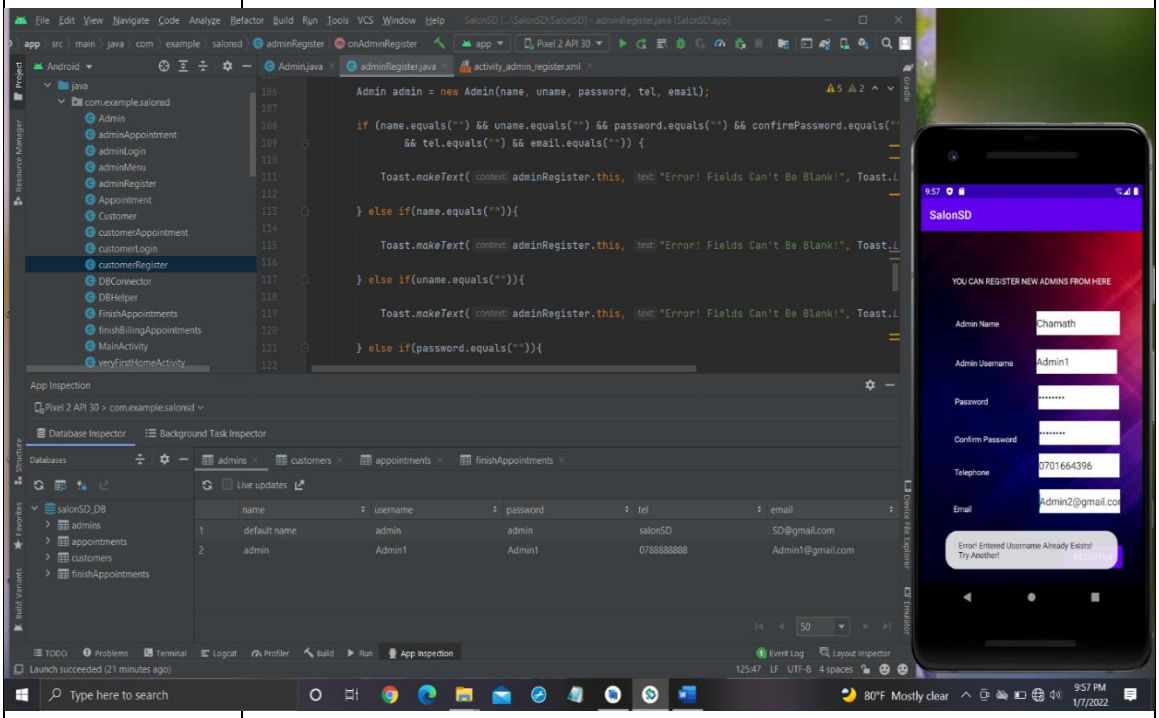
TEST CASE ID	TC2.2.3
TEST CASE NAME	Check by entering only password
TEST SUMMARY	Checking the empty fields validation in Admin Register by filling only password
TEST STEPS	1. Fill only password field 2. Press on register button
TEST DATA	Password: Admin123
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

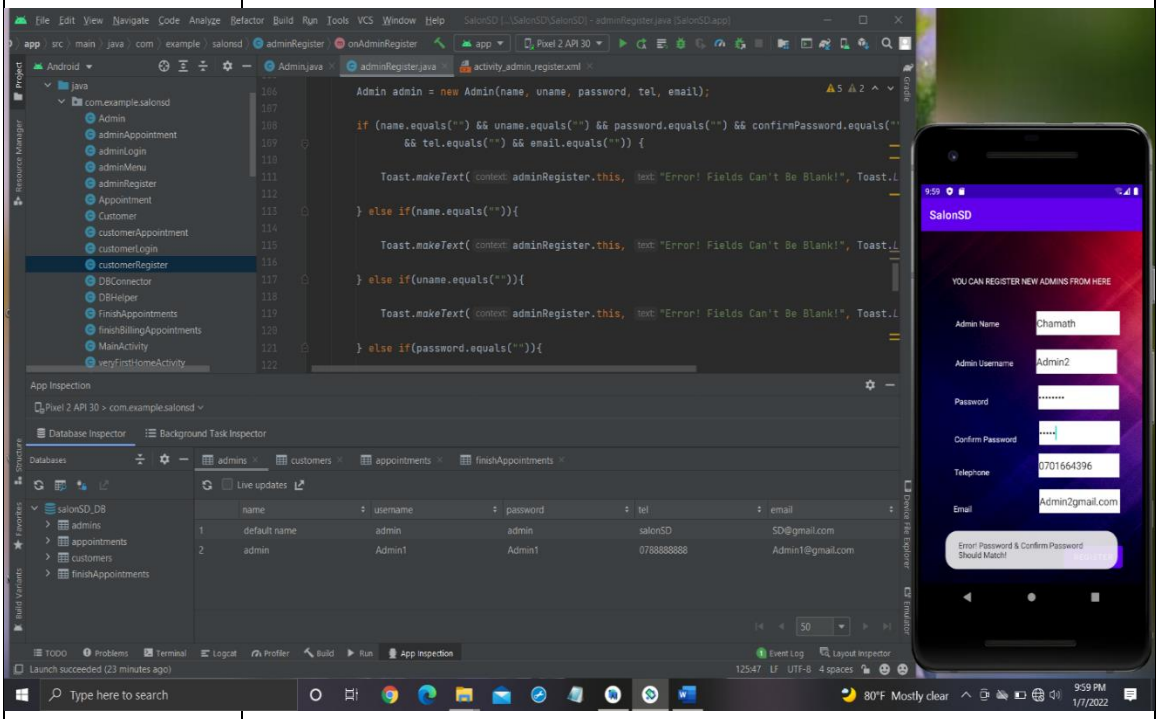
TEST CASE ID	TC2.2.4
TEST CASE NAME	Check by entering only confirm password
TEST SUMMARY	Checking the empty fields validation in Admin Register by filling only confirm password
TEST STEPS	1. Fill only confirm password field 2. Press on register button
TEST DATA	Confirm password: Admin123
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

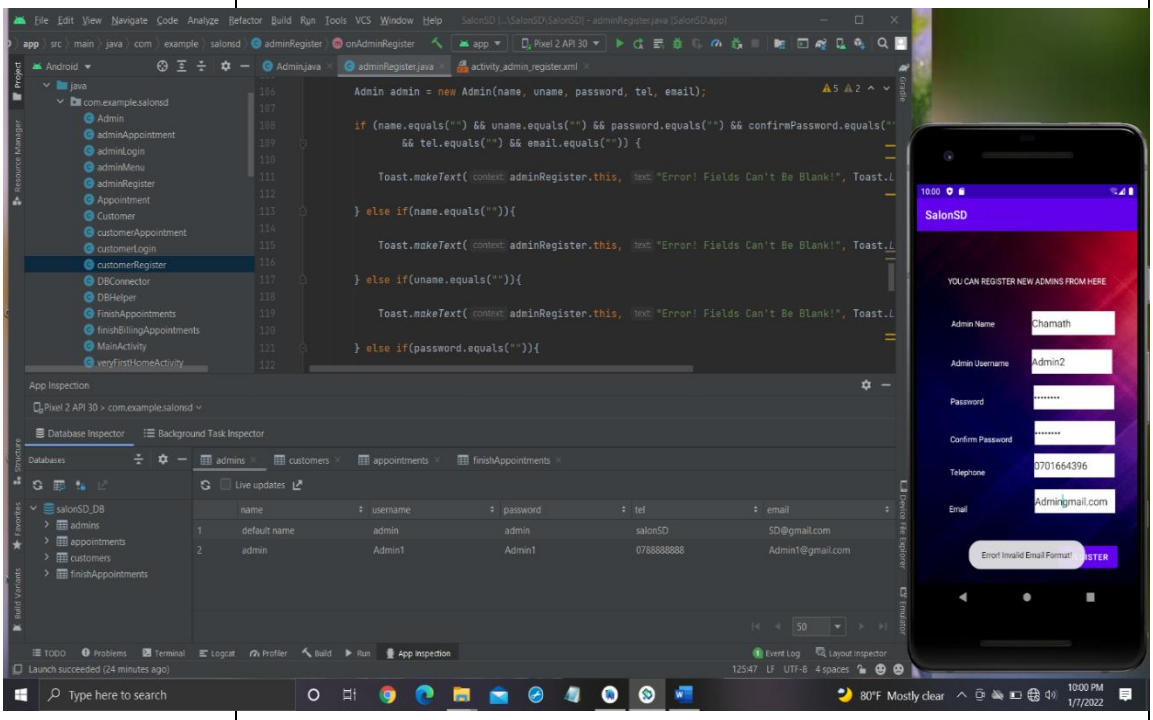
TEST CASE ID	TC2.2.5
TEST CASE NAME	Check by entering only telephone
TEST SUMMARY	Checking the empty fields validation in Admin Register by filling only telephone
TEST STEPS	1. Fill only telephone field 2. Press on register button
TEST DATA	Telephone: 0701664396
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

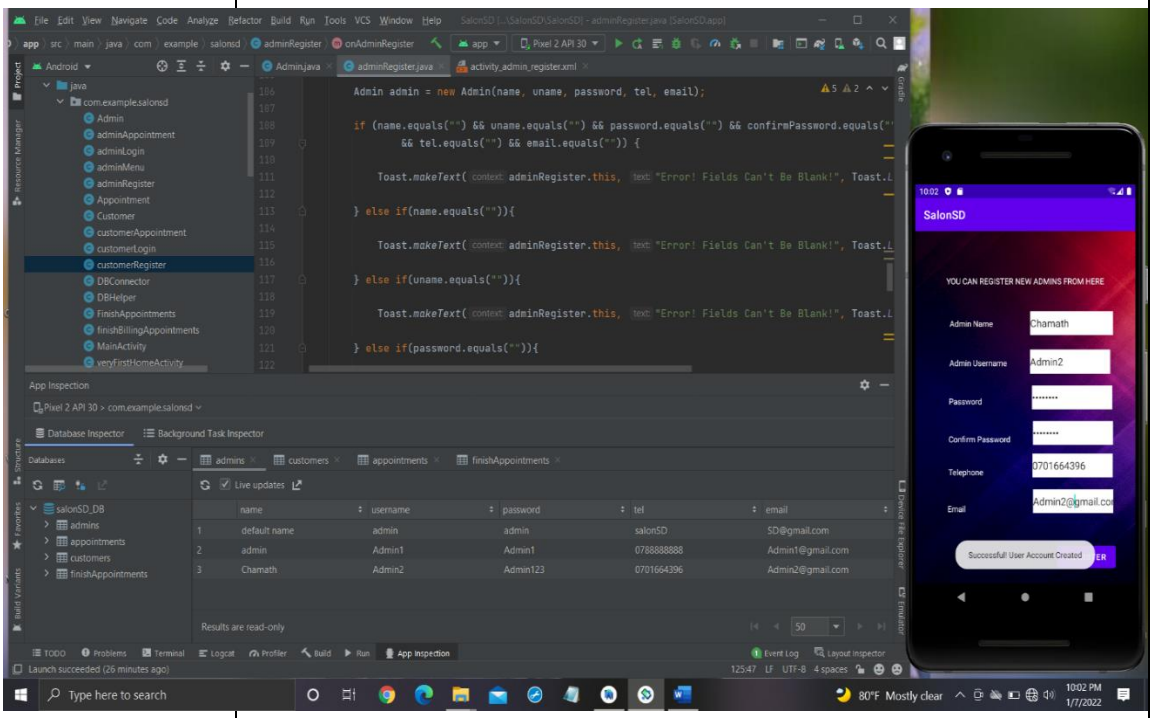
TEST CASE ID	TC2.2.6
TEST CASE NAME	Check by entering only email
TEST SUMMARY	Checking the empty fields validation in Admin Register by filling only email
TEST STEPS	1. Fill only email field 2. Press on register button
TEST DATA	Email: Admin2@gmail.com
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC2.3
TEST CASE NAME	Check by entering already saved username into the username field
TEST SUMMARY	Checking the already saved username validation in Admin Register by filling all other fields with a existing username in the username field
TEST STEPS	1. Fill all the fields 2. Fill username field with already existing username 3. Press on register button
TEST DATA	Admin Name: Chamath Username: Admin1 Password: Admin123 Confirm Password: Admin123 Telephone: 0701664396 Email: Admin2@gmail.com
EXPECTED RESULT	Should display an Error Message as Entered username already exists
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

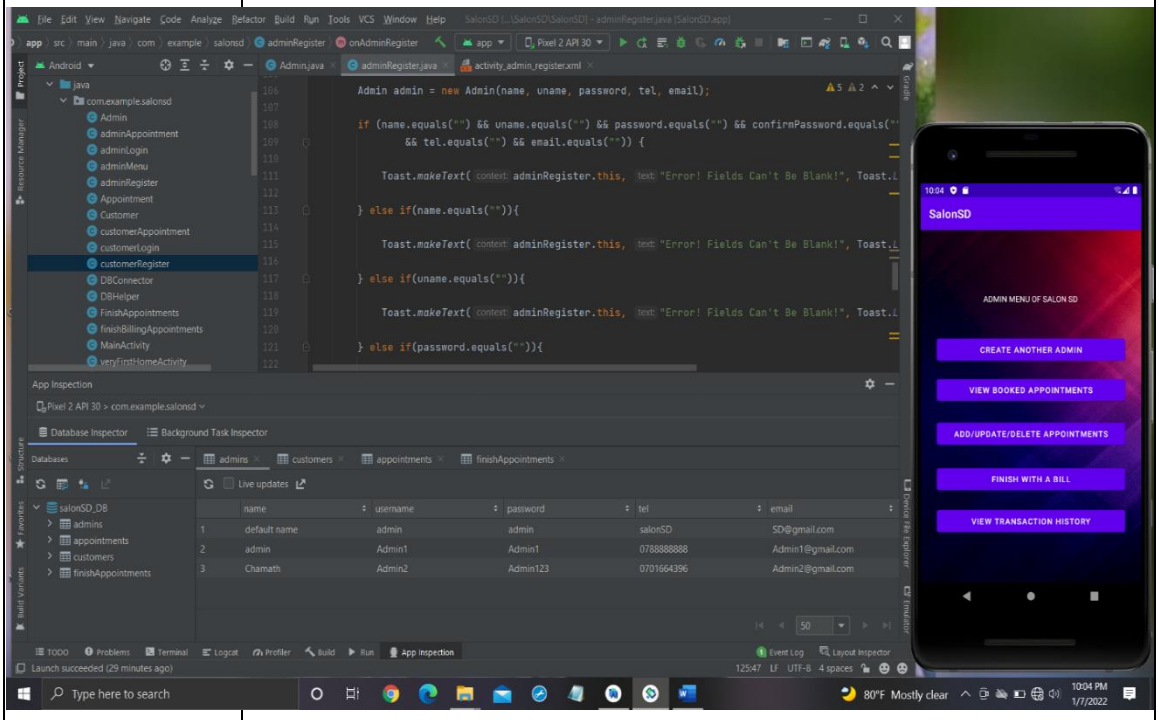


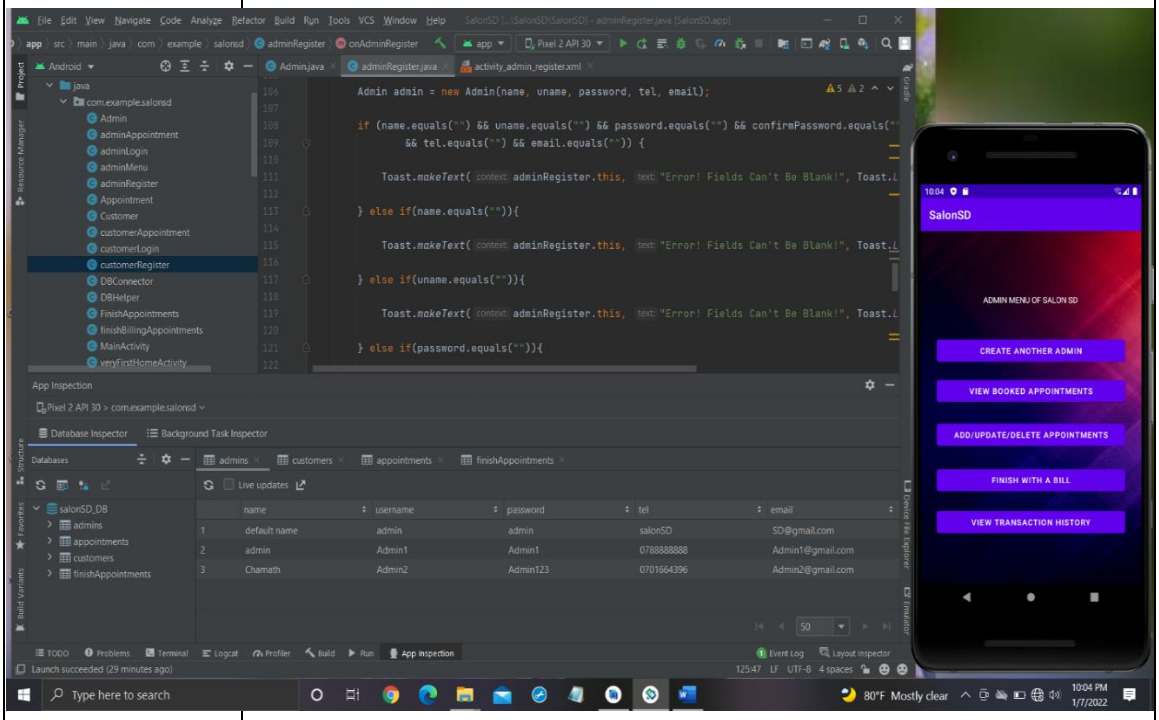
TEST CASE ID	TC2.4
TEST CASE NAME	Check password and confirm password matching
TEST SUMMARY	Checking the password and confirm password matching validation in Admin Register
TEST STEPS	1. Fill all the fields 2. Fill username field with a new username 3. Enter a password 4. Enter a different confirm password 5. Press on register button
TEST DATA	Admin Name: Chamath Username: Admin2 Password: Admin123 Confirm Password: Admin Telephone: 0701664396 Email: Admin2@gmail.com
EXPECTED RESULT	Should display an Error Message as Password & confirm password should match
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC2.5
TEST CASE NAME	Check email format
TEST SUMMARY	Checking the email format validation in Admin Register
TEST STEPS	1. Fill all the fields 2. Fill username field with a new username 3. Enter matching password and confirm password 4. Enter email without the @ symbol 5. Press on register button
TEST DATA	Admin Name: Chamath Username: Admin2 Password: Admin123 Confirm Password: Admin123 Telephone: 0701664396 Email: Admin2gmail.com
EXPECTED RESULT	Should display an Error Message as Invalid Email format
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC2.6
TEST CASE NAME	Check Registration working without any empty field and with new correct data
TEST SUMMARY	Checking the registration with all correct values inside fields in Admin Register
TEST STEPS	1. Fill all the fields 2. Press on register button
TEST DATA	Admin Name: Chamath Username: Admin2 Password: Admin123 Confirm Password: Admin123 Telephone: 0701664396 Email: Admin2@gmail.com
EXPECTED RESULT	Should display a message as User account created successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

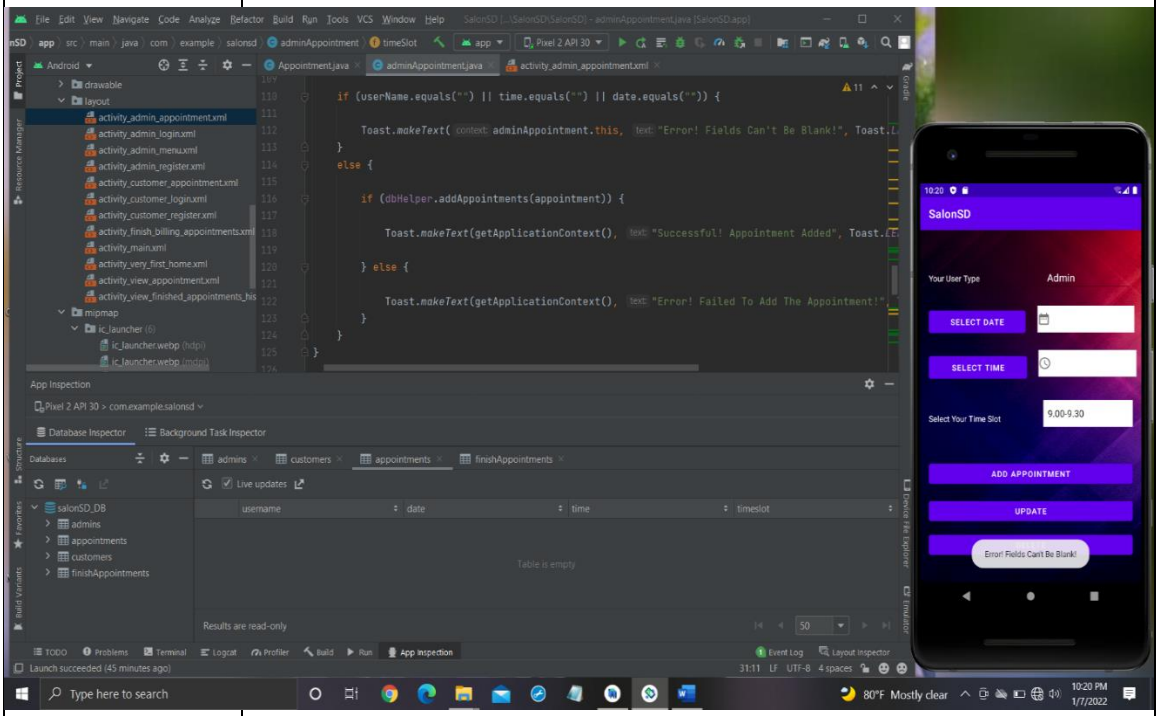
TC2.7 - Check on back when some fields are filled

TEST CASE ID	TC2.7.1
TEST CASE NAME	Check back button notification when some fields are filled
TEST SUMMARY	Checking Admin Registration back button with some empty fields
TEST STEPS	1. Fill only few fields 2. Keep some empty fields 3. Press on back button of the phone
TEST DATA	Admin Name: Empty Username: Empty Password: Admin123 Confirm Password: Admin123 Telephone: 0701664396 Email: Admin2@gmail.com
EXPECTED RESULT	A notification should pop up and ask whether user wants to move back and get user's response.
ACTUAL RESULT	
CONCLUSION	The expected notification displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass
TEST CASE ID	TC2.7.2

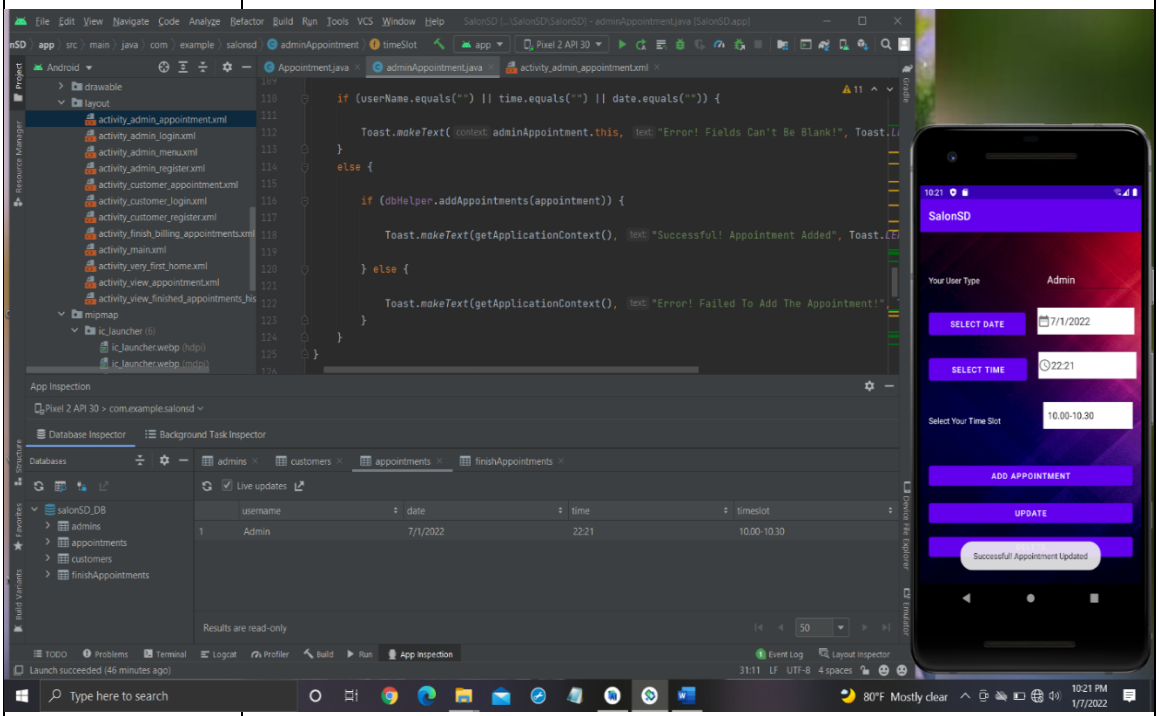
TEST CASE NAME	Check back button notification when user's response is YES
TEST SUMMARY	Checking YES response
TEST STEPS	Press on YES
TEST DATA	YES response
EXPECTED RESULT	If user provided YES, the previous interface will be displayed.
ACTUAL RESULT	
CONCLUSION	The expected interface displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

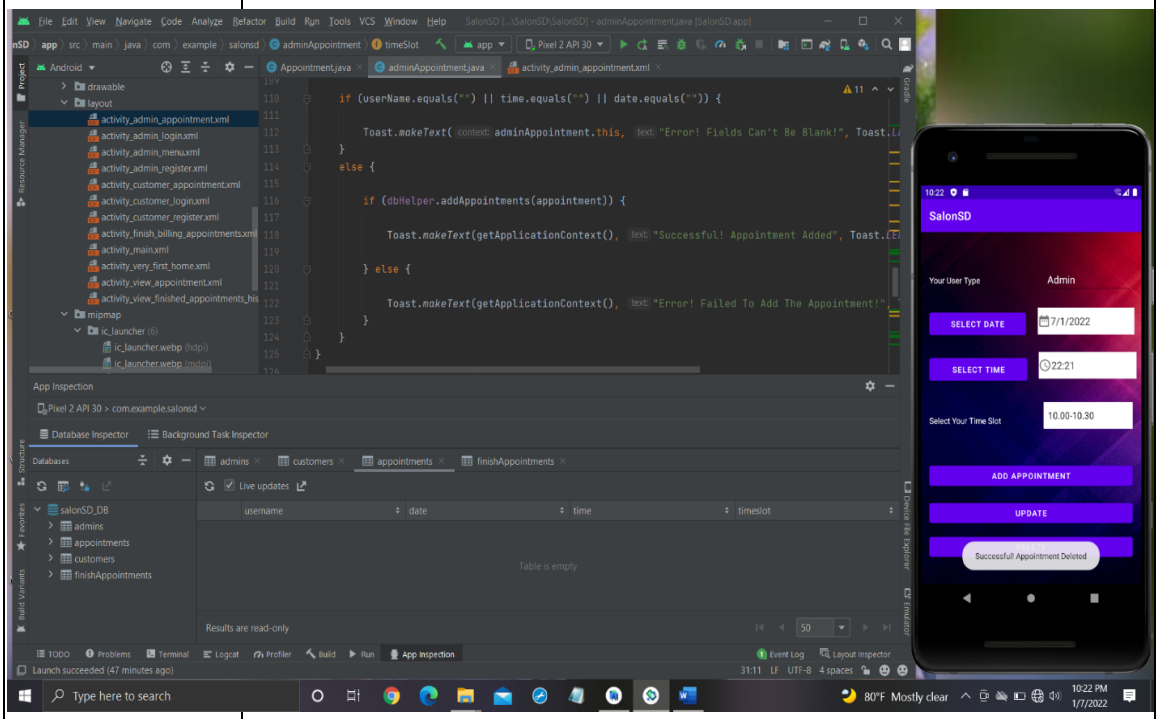
TEST CASE ID	TC2.7.2
TEST CASE NAME	Check back button notification when user's response is NO
TEST SUMMARY	Checking NO response
TEST STEPS	Press on NO
TEST DATA	NO response
EXPECTED RESULT	If user provided NO, that notification will fade away and a message will pop up mentioning that Continue Editing
ACTUAL RESULT	
CONCLUSION	The expected toast message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TC3 - Admin Appointment

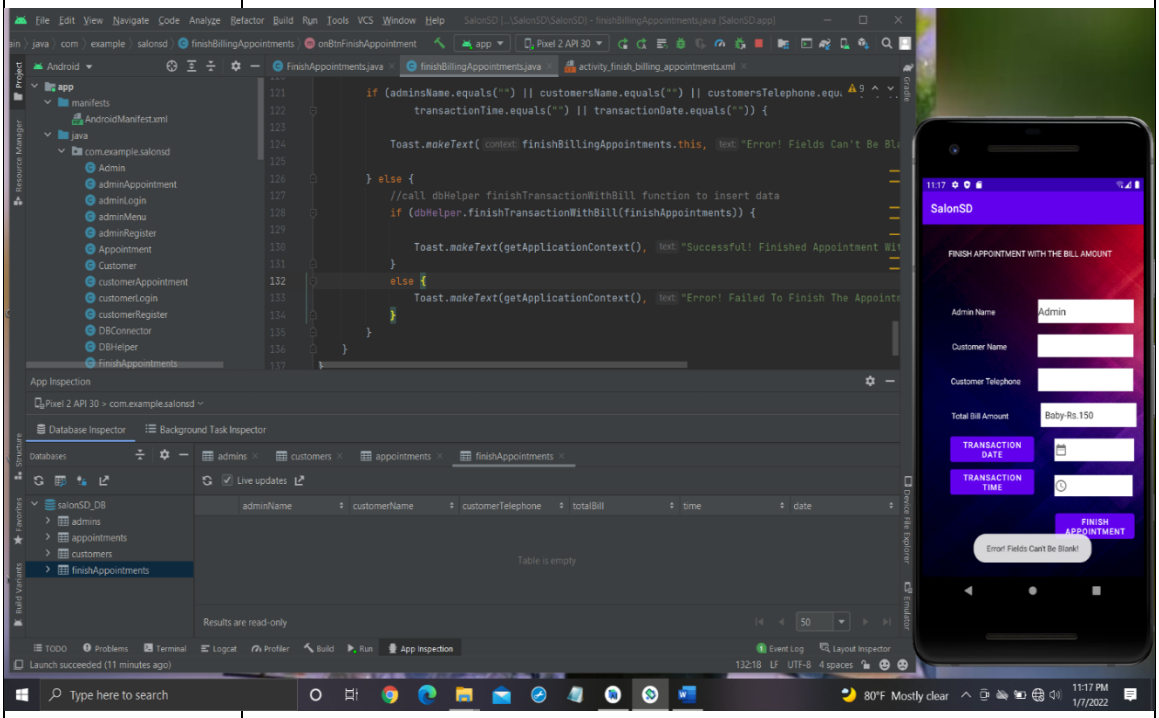
TEST CASE ID	TC3.1
TEST CASE NAME	Check Add Appointment with empty fields
TEST SUMMARY	Checking empty fields validation in Admin Appointment interface
TEST STEPS	Press on Add Appointment button (without filling any field)
TEST DATA	Empty fields
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

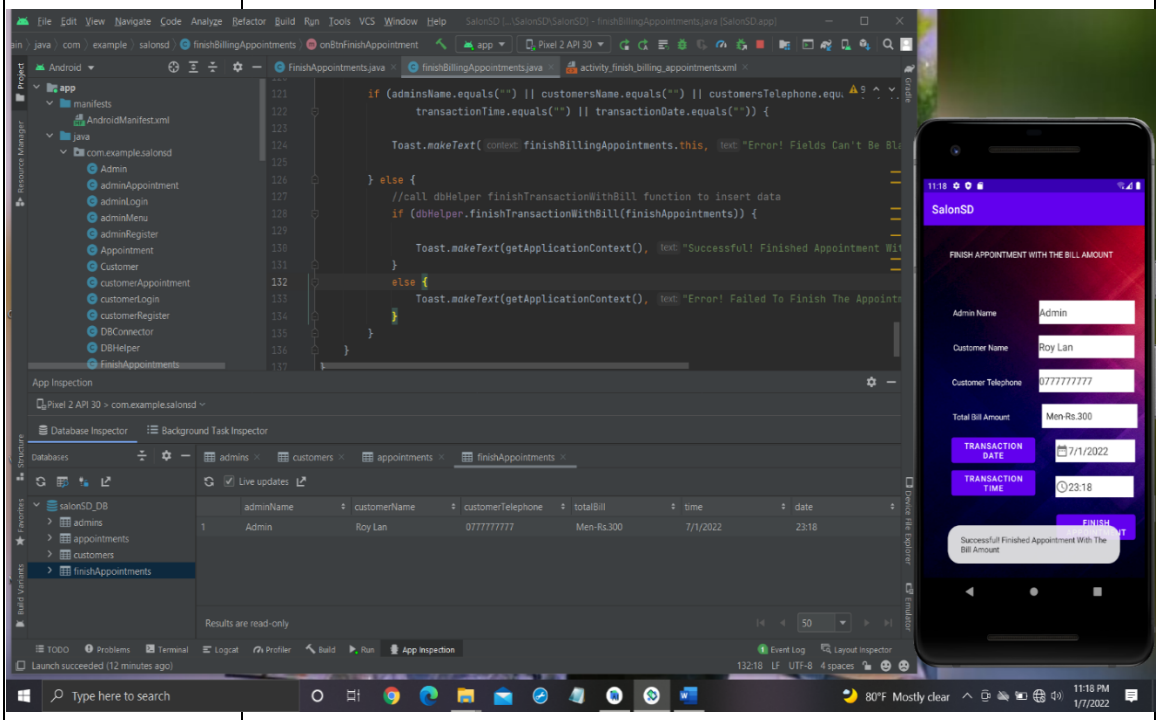
TEST CASE ID	TC3.2
TEST CASE NAME	Check Add Appointment working without any empty field
TEST SUMMARY	Checking correctly working of Add Appointment in Admin Appointment interface
TEST STEPS	1. Have to fill all fields 2. Press on Add Appointment button
TEST DATA	User Type: Admin Date: Current Date Time: Current Time Time Slot: Select Time Slot (9.00-9.30) from the Spinner
EXPECTED RESULT	Should display a message as Appointment added successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC3.3
TEST CASE NAME	Check Update Appointment working without any empty and updated fields
TEST SUMMARY	Checking correctly working of Update Appointment in Admin Appointment interface
TEST STEPS	1. Have to fill all fields 2. Change the fields that are want to be changed 3. Press on Update Appointment button
TEST DATA	User Type: Admin Date: Change the date Time: Change the time Time Slot: Change the Time Slot(9.00-9.30) to the 10.00-10.30 Time Slot from the Spinner
EXPECTED RESULT	Should display a message as Appointment updated successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

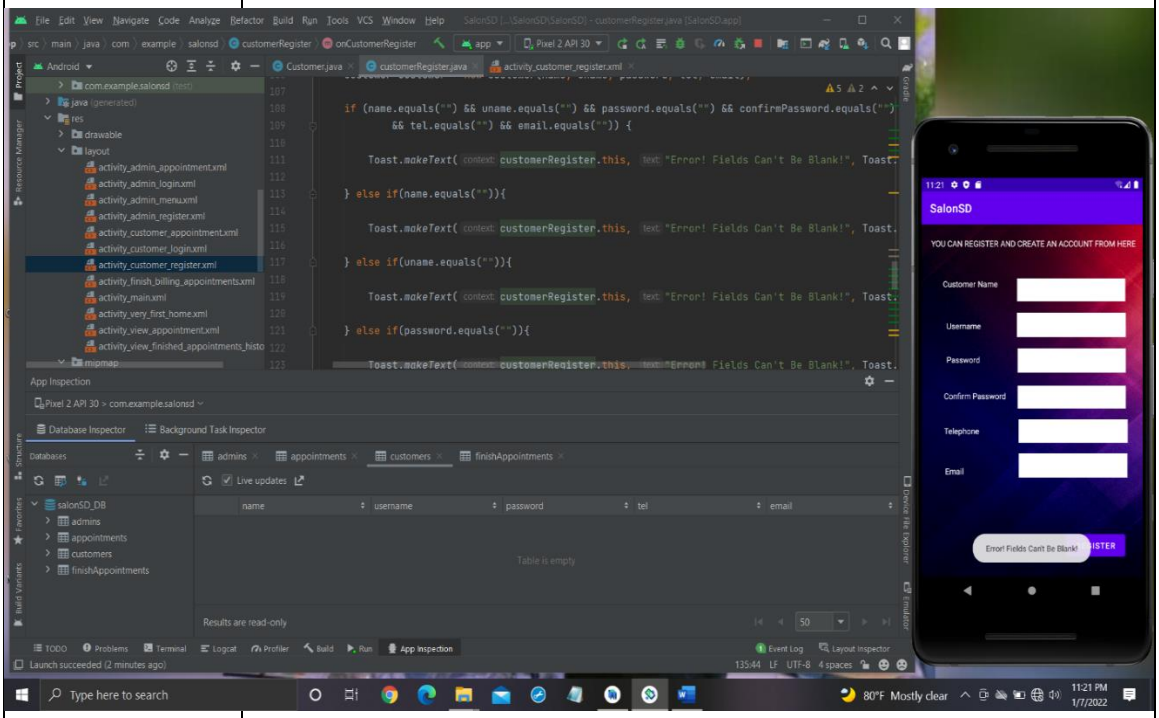
TEST CASE ID	TC3.4
TEST CASE NAME	Check Delete Appointment working
TEST SUMMARY	Checking correctly working of Delete Appointment in Admin Appointment interface
TEST STEPS	1. Have to fill all fields (Specially the username field) 2. Press on Delete Appointment button
TEST DATA	Username: Admin
EXPECTED RESULT	Should display a message as Appointment deleted successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TC4 - Finish Billing Appointments

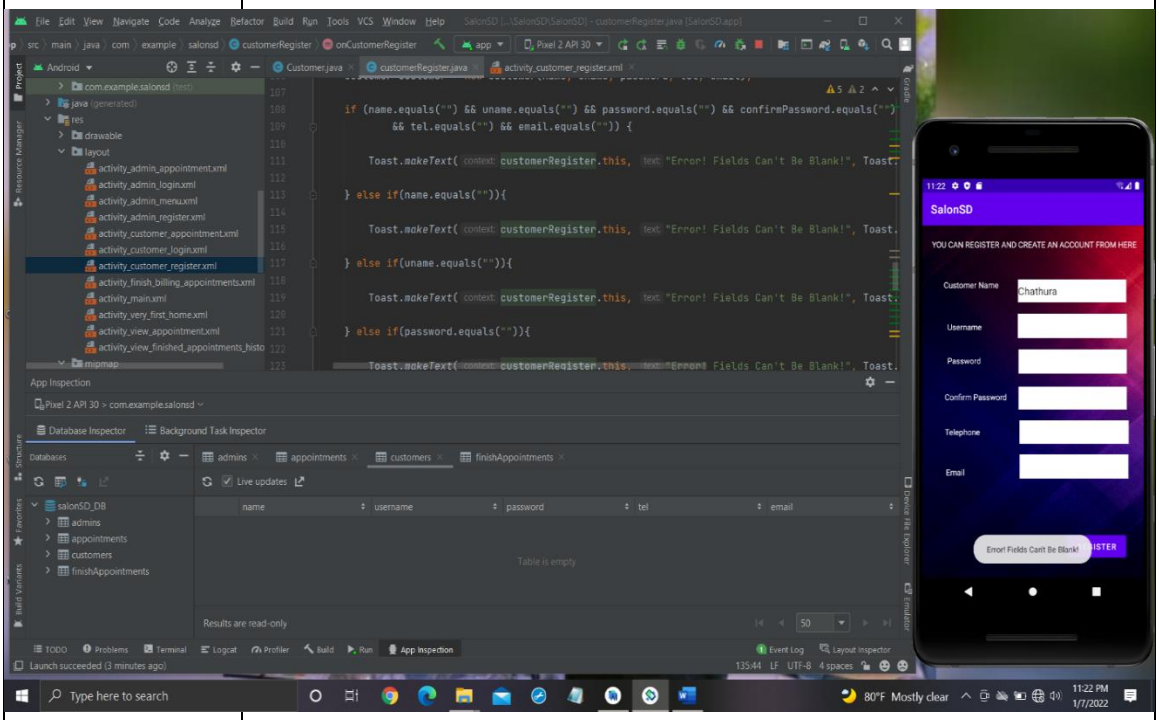
TEST CASE ID	TC4.1
TEST CASE NAME	Check Finish Appointment with empty fields
TEST SUMMARY	Checking empty fields validation in Finish Appointment interface
TEST STEPS	Press on Finish Appointment button (without filling any fields)
TEST DATA	Empty fields
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC4.2
TEST CASE NAME	Check Finish Appointment working without any empty field
TEST SUMMARY	Checking correctly working of Finish Appointment in Admin Finish Appointment interface
TEST STEPS	1. Have to fill all fields 2. Press on Finish Appointment button
TEST DATA	Admin Name: Admin Customer Name: Roy Lan Customer Telephone: 0777777777 Total Bill Amount: Men – Rs.300 from the spinner Transaction Date: Current Date Transaction Time: Current Time
EXPECTED RESULT	Should display a message as Appointment Finished successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

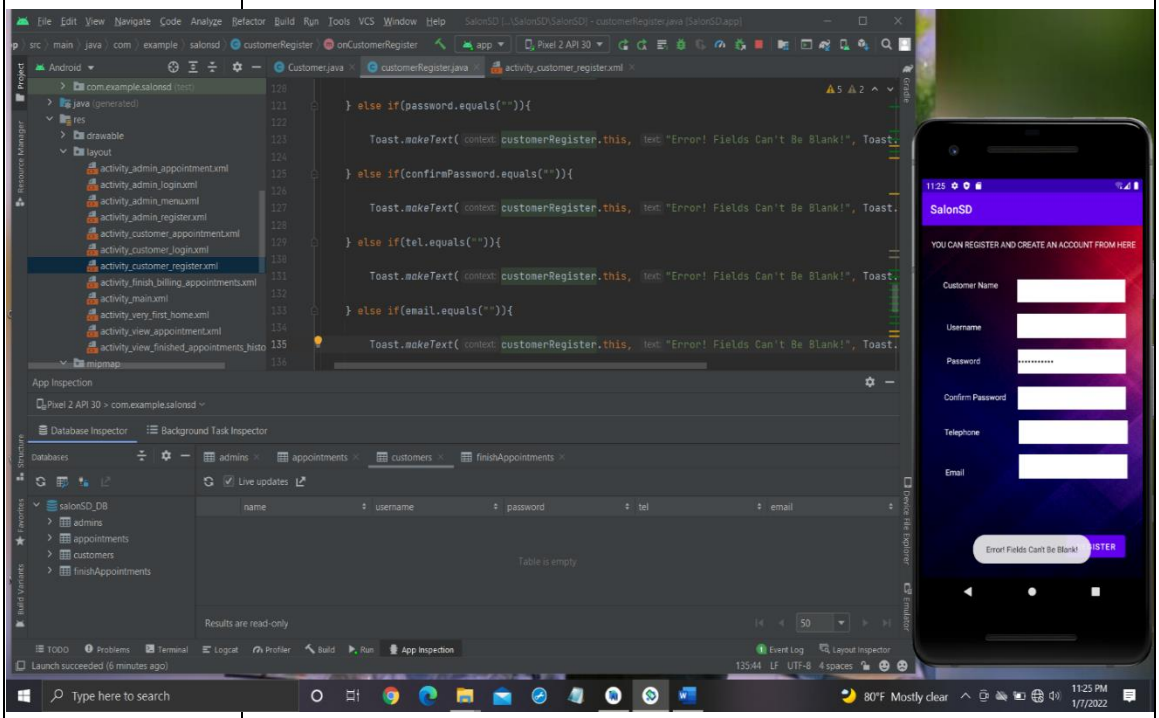
TC5 - Customer Registration

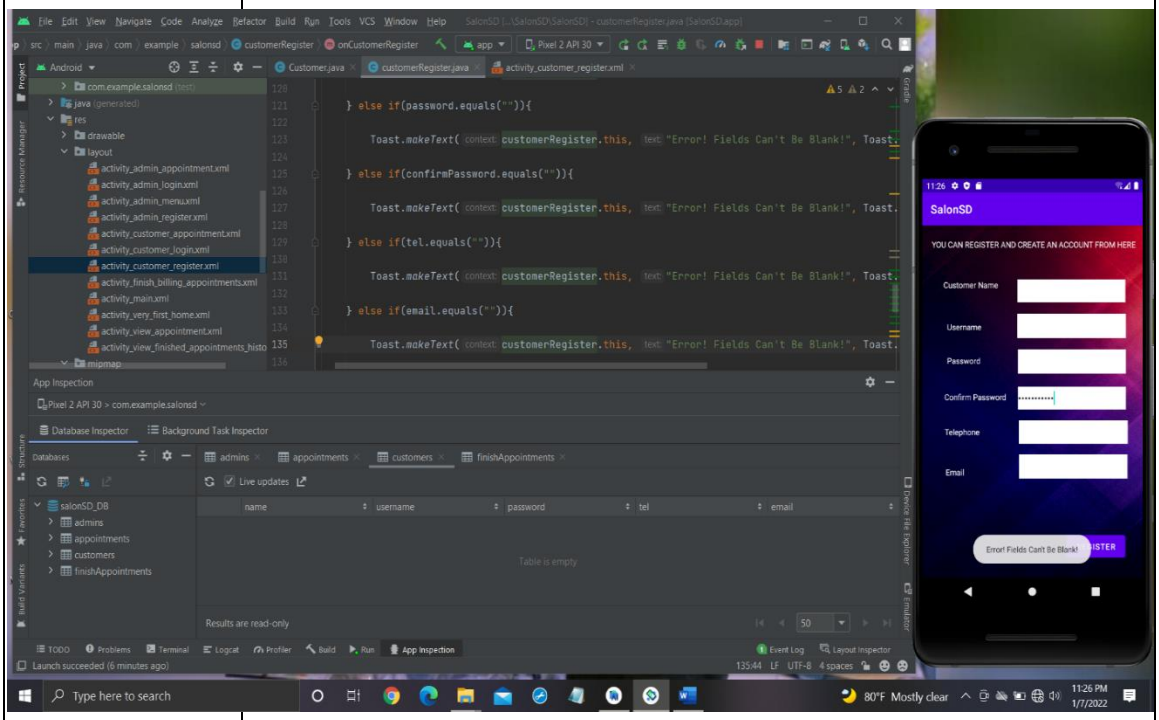
TEST CASE ID	TC5.1
TEST CASE NAME	Check Register empty fields
TEST SUMMARY	Checking the empty fields validation in Customer Register interface
TEST STEPS	Press on register button (without filling any field)
TEST DATA	Empty fields
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

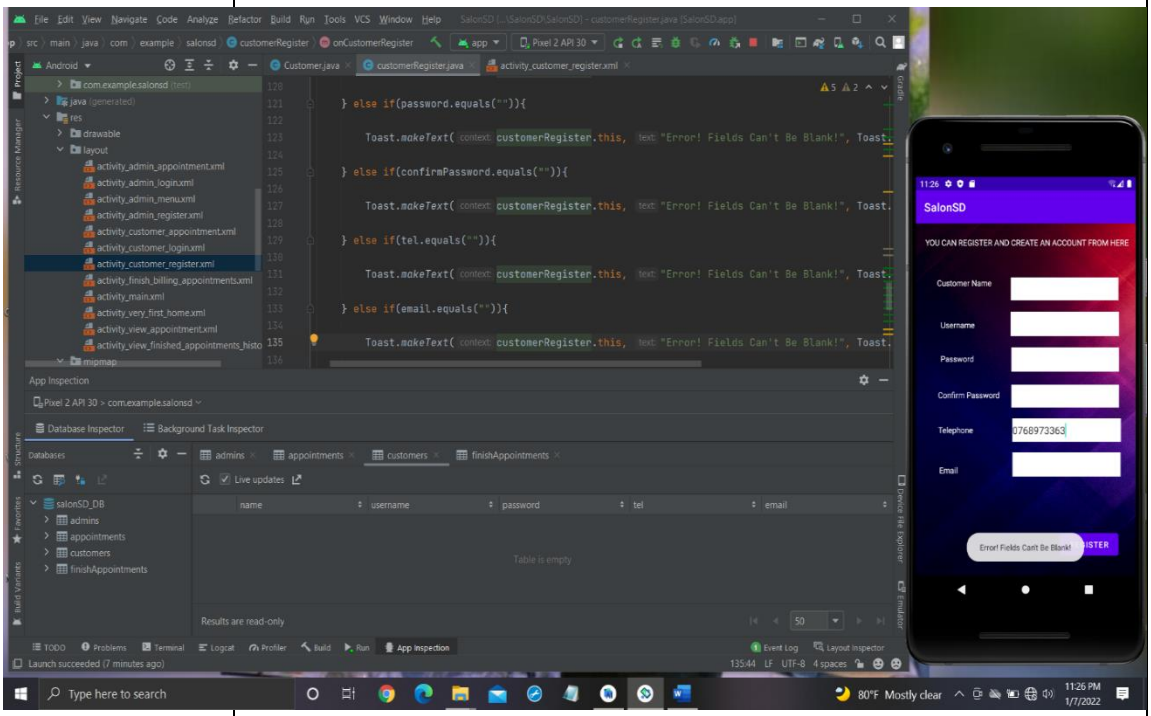
TC5.2 - Check by entering values one by one into fields

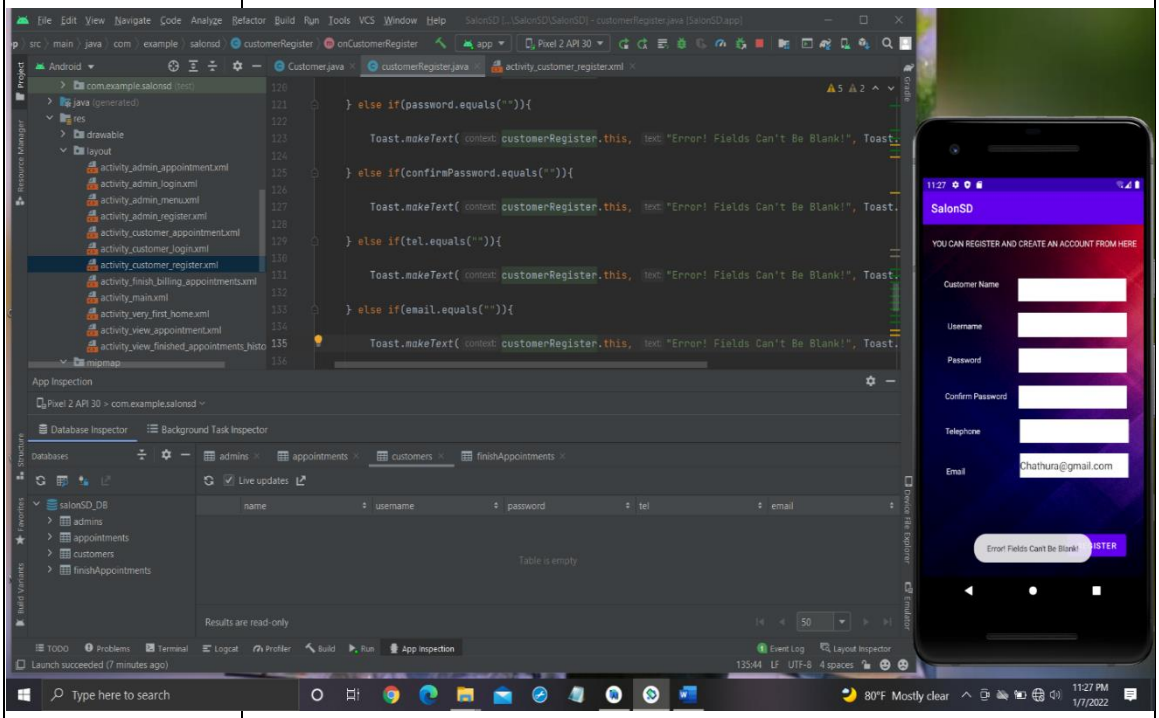
TEST CASE ID	TC5.2.1
TEST CASE NAME	Check by entering only customer name
TEST SUMMARY	Checking the empty fields validation in Customer Register by filling only customer name
TEST STEPS	3. Fill only customer name field 4. Press on register button
TEST DATA	Customer name: Chathura
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC5.2.2
TEST CASE NAME	Check by entering only username
TEST SUMMARY	Checking the empty fields validation in Customer Register by filling only username
TEST STEPS	3. Fill only username field 4. Press on register button
TEST DATA	Username: Mr.Chathura
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

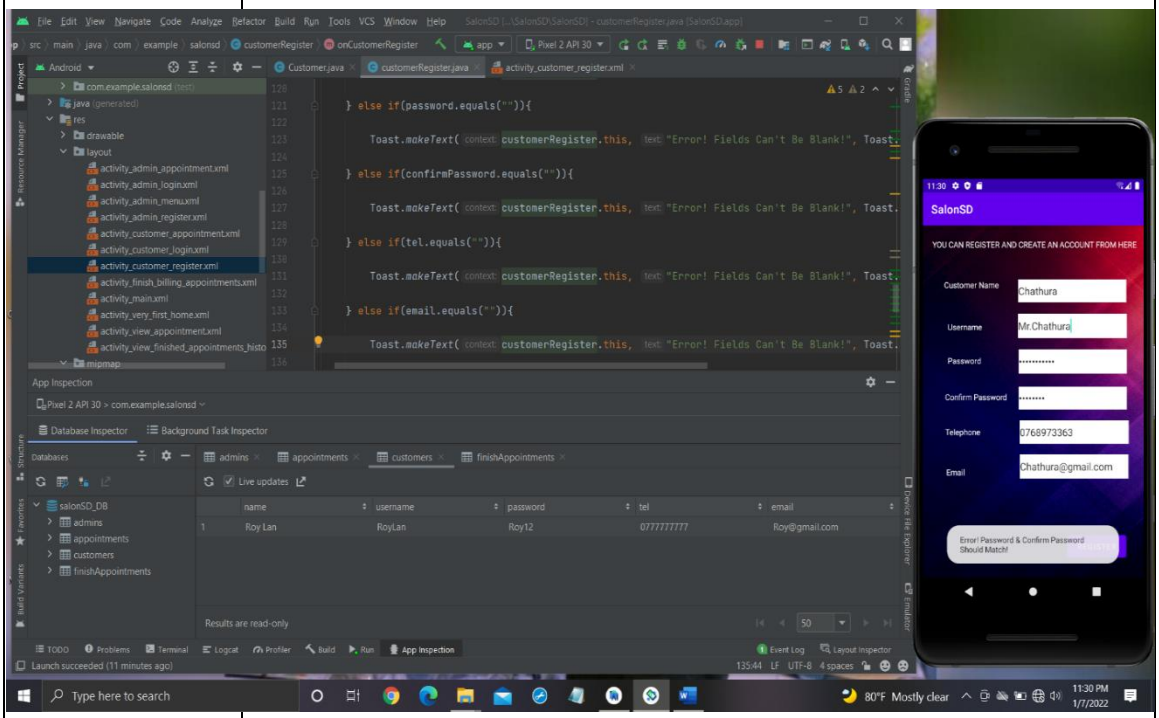
TEST CASE ID	TC5.2.3
TEST CASE NAME	Check by entering only password
TEST SUMMARY	Checking the empty fields validation in Customer Register by filling only password
TEST STEPS	3. Fill only password field 4. Press on register button
TEST DATA	Password: Chathura123
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

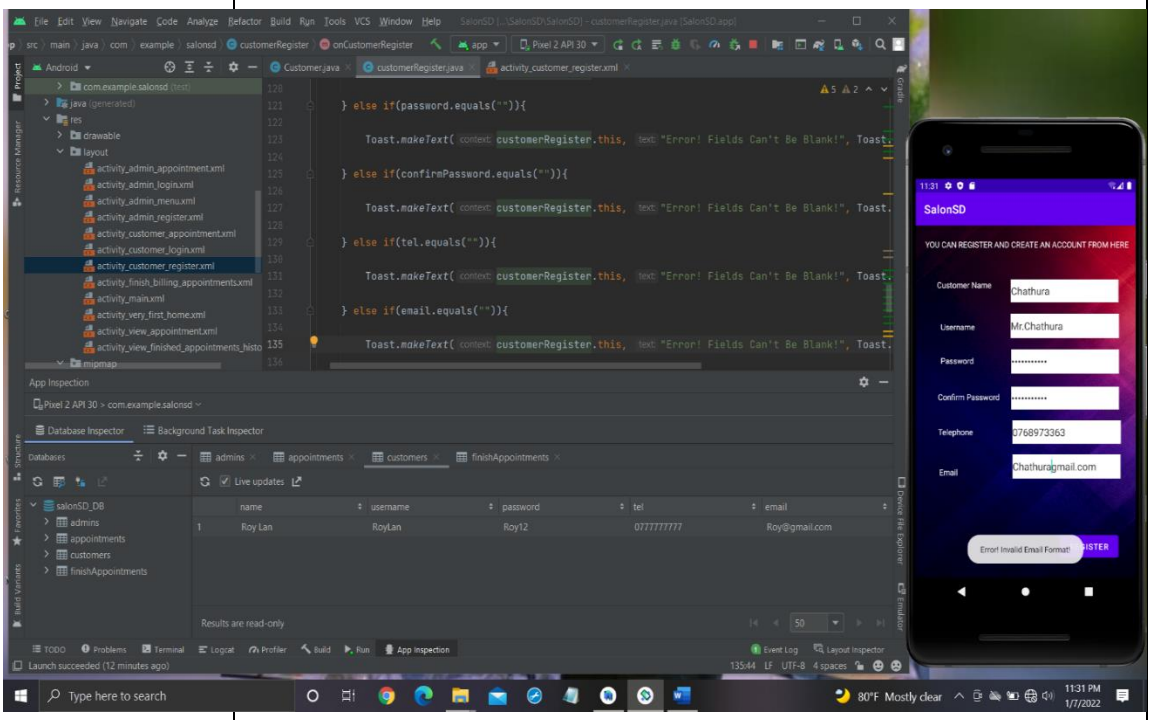
TEST CASE ID	TC5.2.4
TEST CASE NAME	Check by entering only confirm password
TEST SUMMARY	Checking the empty fields validation in Customer Register by filling only confirm password
TEST STEPS	3. Fill only confirm password field 4. Press on register button
TEST DATA	Confirm password: Chathura123
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

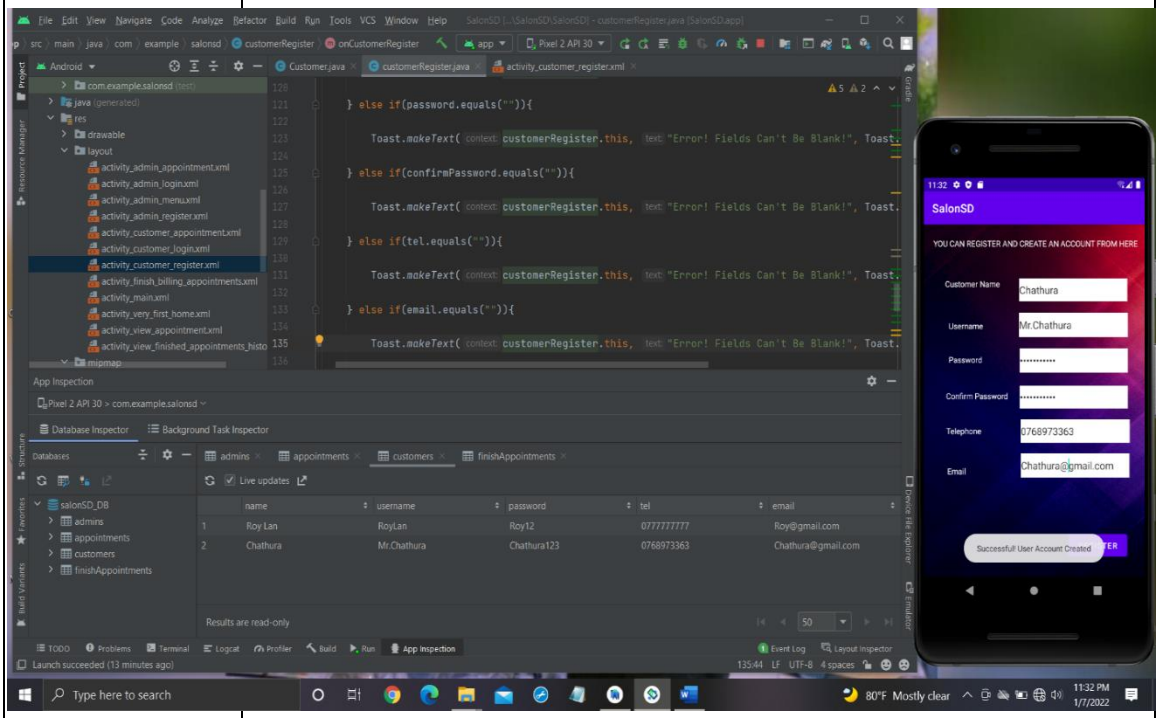
TEST CASE ID	TC5.2.5
TEST CASE NAME	Check by entering only telephone
TEST SUMMARY	Checking the empty fields validation in Customer Register by filling only telephone
TEST STEPS	3. Fill only telephone field 4. Press on register button
TEST DATA	Telephone: 0768973363
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC5.2.6
TEST CASE NAME	Check by entering only email
TEST SUMMARY	Checking the empty fields validation in Customer Register by filling only email
TEST STEPS	3. Fill only email field 4. Press on register button
TEST DATA	Email: Chathura@gmail.com
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

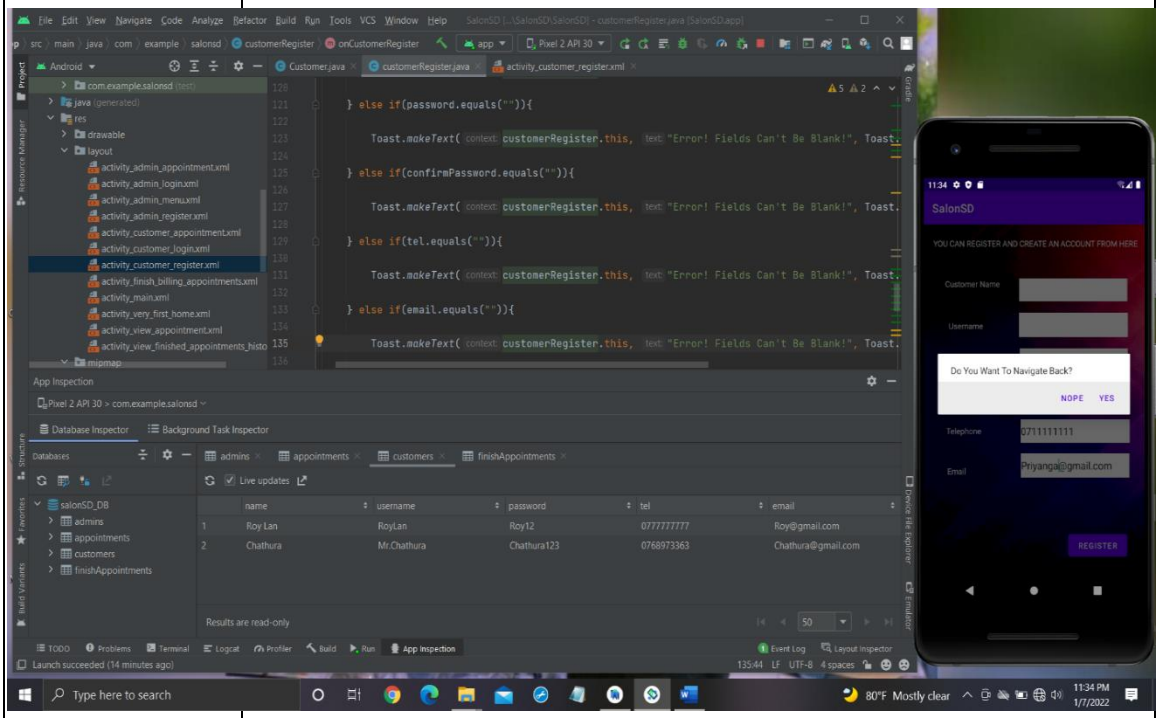
TEST CASE ID	TC2.3
TEST CASE NAME	Check by entering already saved username into the username field
TEST SUMMARY	Checking the already saved username validation in Customer Register by filling all other fields with an existing username in the username field
TEST STEPS	4. Fill all the fields 5. Fill username field with already existing username 6. Press on register button
TEST DATA	Admin Name: Chathura Username: RoyLan Password: Chathura123 Confirm Password: Chathura123 Telephone: 0768973363 Email: Chathura@gmail.com
EXPECTED RESULT	Should display an Error Message as Entered username already exists
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

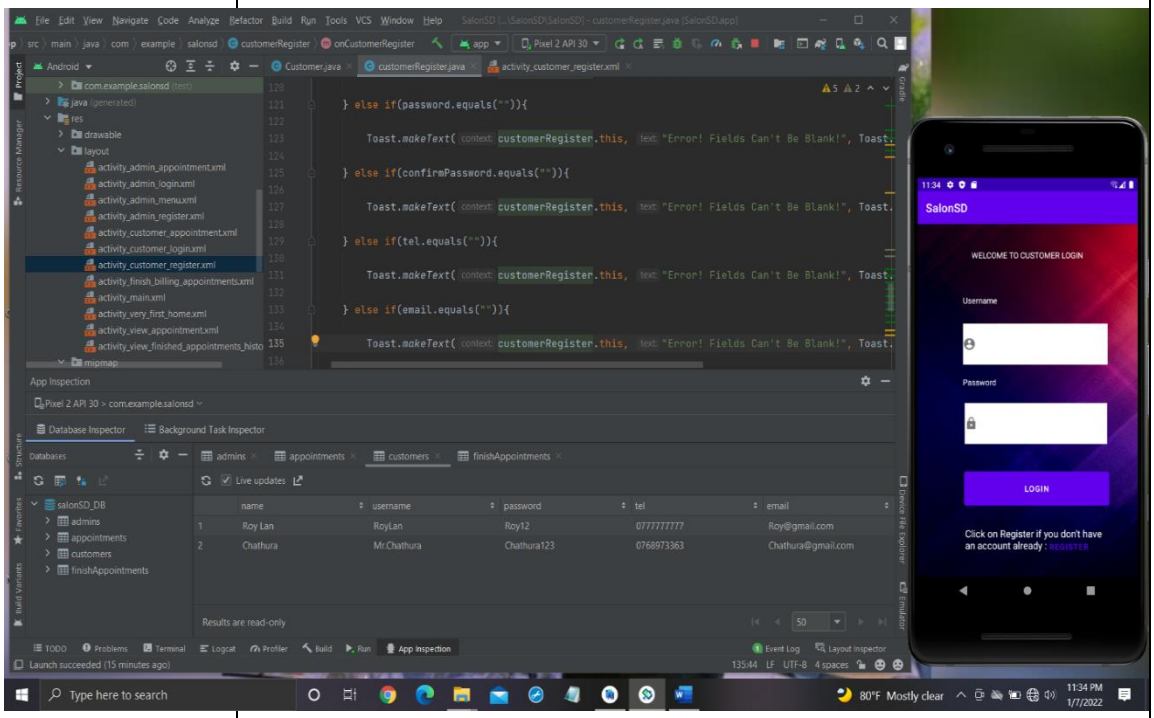
TEST CASE ID	TC5.4
TEST CASE NAME	Check password and confirm password matching
TEST SUMMARY	Checking the password and confirm password matching validation in Customer Register
TEST STEPS	6. Fill all the fields 7. Fill username field with a new username 8. Enter a password 9. Enter a different confirm password 10. Press on register button
TEST DATA	Customer Name: Chathura Username: Mr.Chathura Password: Chathura123 Confirm Password: Chathura Telephone: 0768973363 Email: Chathura@gmail.com
EXPECTED RESULT	Should display an Error Message as Password & confirm password should match
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

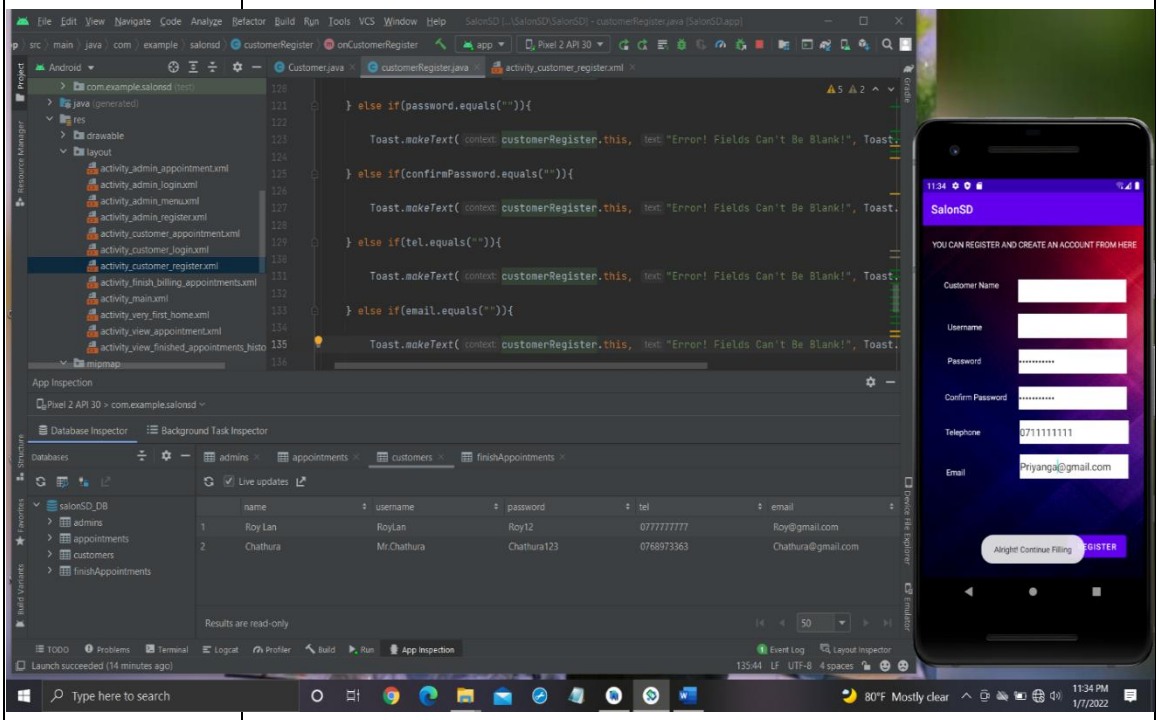
TEST CASE ID	TC5.5
TEST CASE NAME	Check email format
TEST SUMMARY	Checking the email format validation in Customer Register
TEST STEPS	6. Fill all the fields 7. Fill username field with a new username 8. Enter matching password and confirm password 9. Enter email without the @ symbol 10. Press on register button
TEST DATA	Customer Name: Chathura Username: Mr.Chathura Password: Chathura123 Confirm Password: Chathura123 Telephone: 0768973363 Email: Chathuragmail.com
EXPECTED RESULT	Should display an Error Message as Invalid Email format
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC5.6
TEST CASE NAME	Check Registration working without any empty field and with new correct data
TEST SUMMARY	Checking the registration with all correct values inside fields in Customer Register
TEST STEPS	3. Fill all the fields 4. Press on register button
TEST DATA	Customer Name: Chathura Username: Mr.Chathura Password: Chathura123 Confirm Password: Chathura123 Telephone: 0768973363 Email: Chathura@gmail.com
EXPECTED RESULT	Should display a message as User account created successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

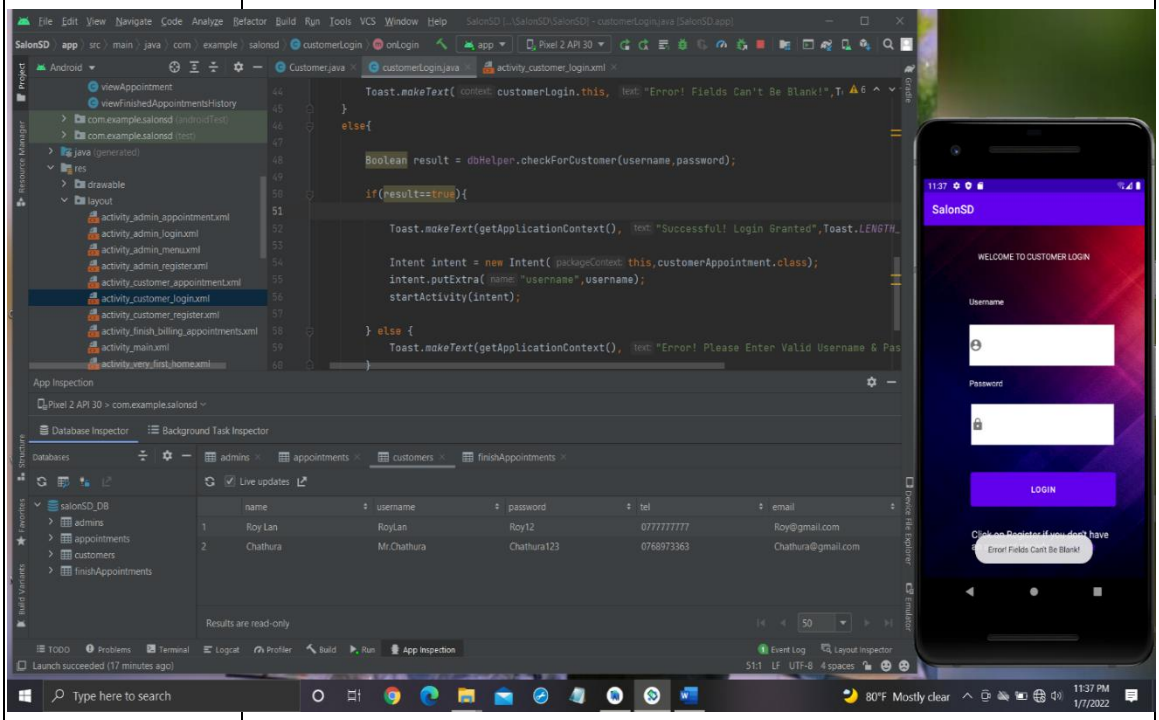
TC5.7 - Check on back when some fields are filled

TEST CASE ID	TC5.7.1
TEST CASE NAME	Check back button notification when some fields are filled
TEST SUMMARY	Checking Customer Registration back button with some empty fields
TEST STEPS	4. Fill only few fields 5. Keep some empty fields 6. Press on back button of the phone
TEST DATA	Customer Name: Empty Username: Empty Password: Priyanga456 Confirm Password: Priyanga456 Telephone: 0711111111 Email: Priyanga @gmail.com
EXPECTED RESULT	A notification should pop up and ask whether user wants to move back and get user's response.
ACTUAL RESULT	
CONCLUSION	The expected notification displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

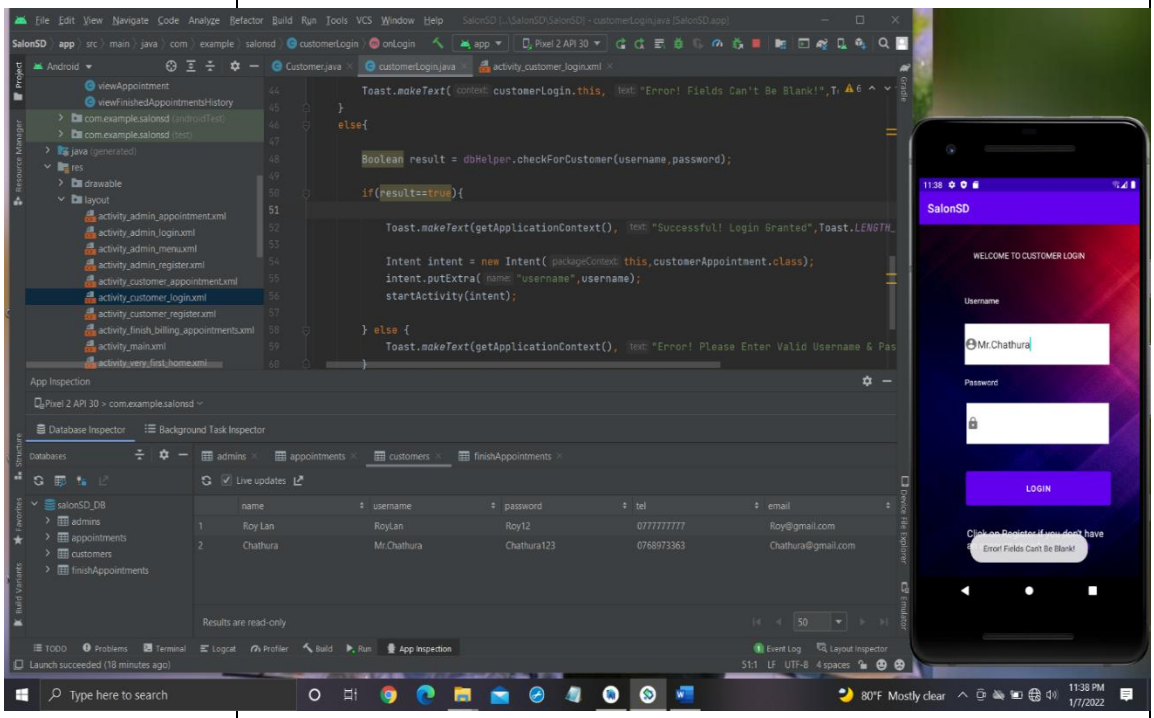
TEST CASE ID	TC5.7.2
TEST CASE NAME	Check back button notification when user's response is YES
TEST SUMMARY	Checking YES response
TEST STEPS	Press on YES
TEST DATA	YES response
EXPECTED RESULT	If user provided YES, the previous interface will be displayed.
ACTUAL RESULT	
CONCLUSION	The expected interface displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

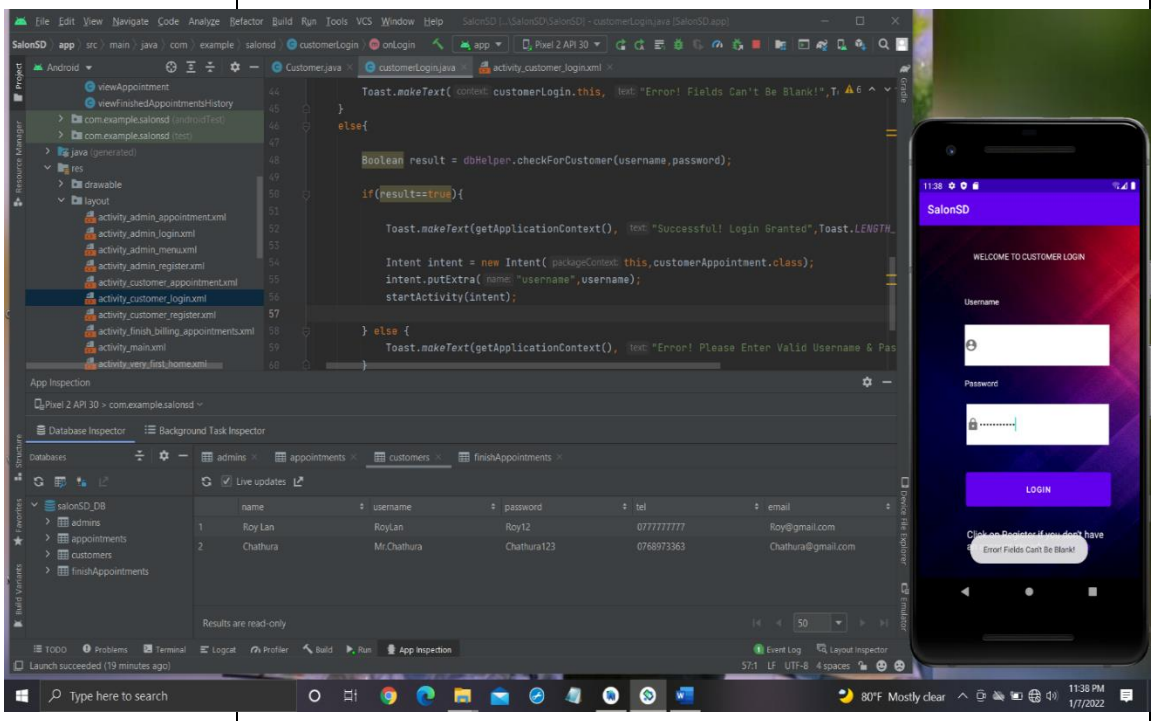
TEST CASE ID	TC5.7.2
TEST CASE NAME	Check back button notification when user's response is NO
TEST SUMMARY	Checking NO response
TEST STEPS	Press on NO
TEST DATA	NO response
EXPECTED RESULT	If user provided NO, that notification will fade away and a message will pop up mentioning that Continue Editing
ACTUAL RESULT	
CONCLUSION	The expected toast message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TC6 - Customer Login

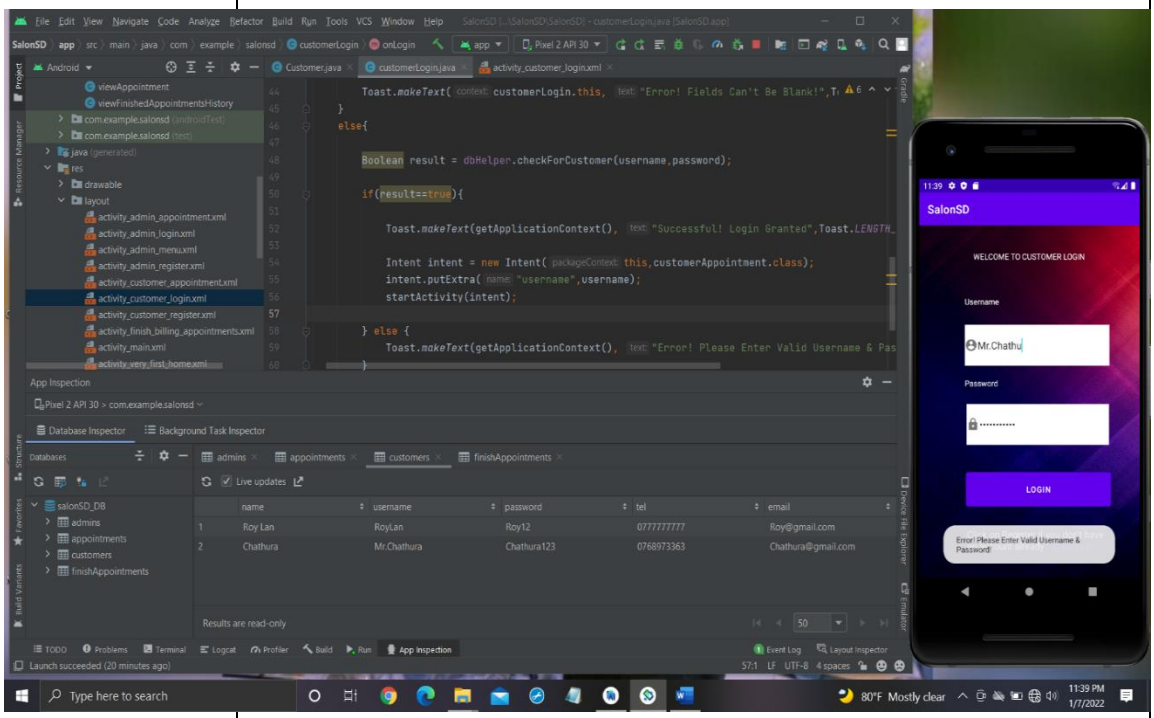
TEST CASE ID	TC6.1
TEST CASE NAME	Check Login empty fields
TEST SUMMARY	Checking the empty fields validation in Customer Login interface
TEST STEPS	Press on login button (without filling any field)
TEST DATA	Empty fields
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

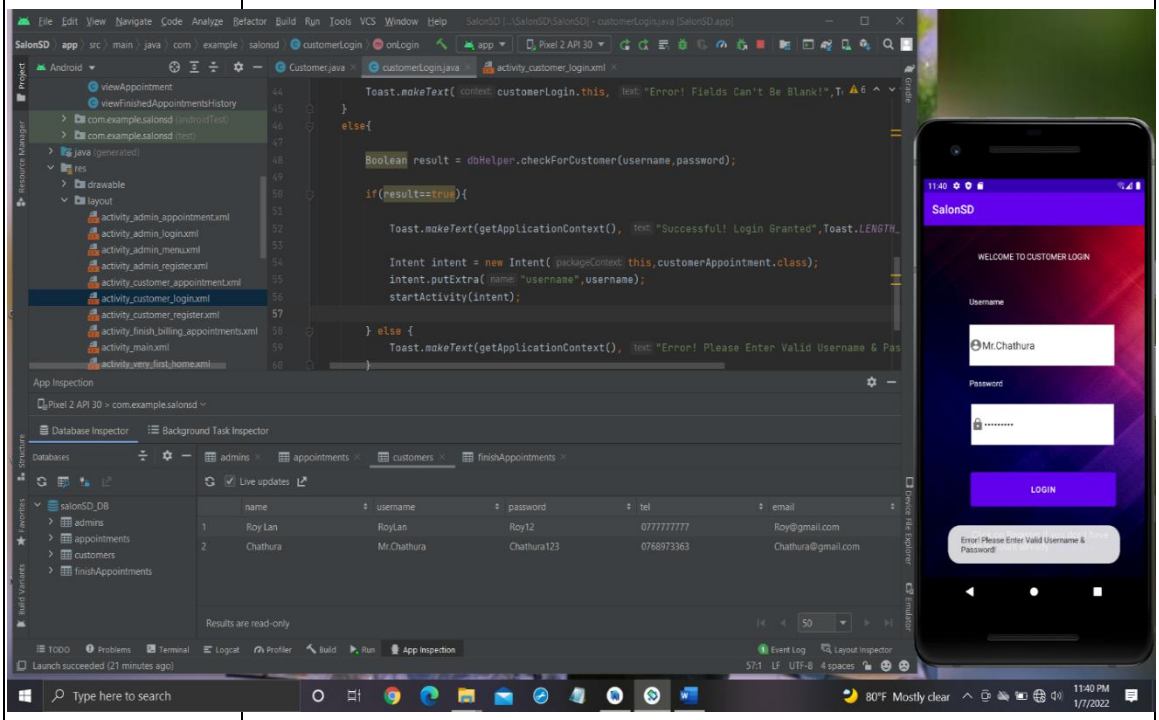
TC6.2 - Check fields by entering values one by one into fields

TEST CASE ID	TC6.2.1
TEST CASE NAME	Check by filling only username field
TEST SUMMARY	Checking empty field validation by filling only one field
TEST STEPS	3. Fill only username field with a valid customer username 4. Press on Login button
TEST DATA	Username: Mr.Chathura Password: empty
EXPECTED RESULT	Should display the same Error Message which displayed previously as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages displayed as a toast message with very clear outcomes and the same error displayed in the above test case has received here also.
STATUS (PASS/FAIL)	Pass

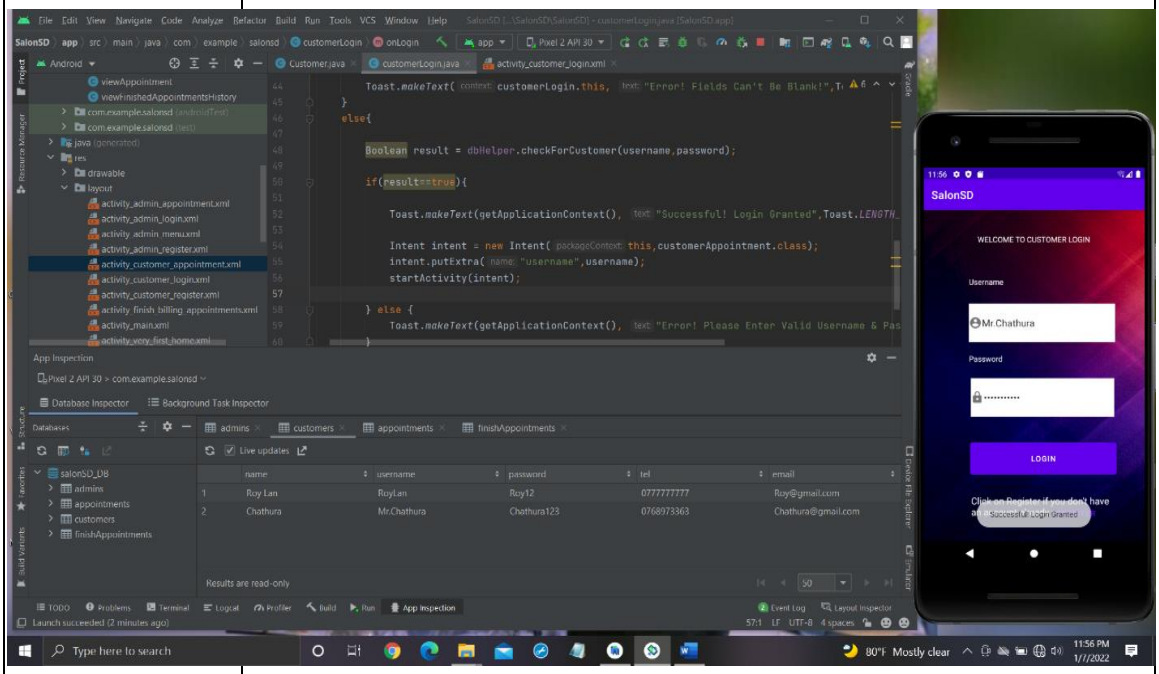
TEST CASE ID	TC6.2.2
TEST CASE NAME	Check by filling only password field
TEST SUMMARY	Checking empty field validation by filling only one field
TEST STEPS	3. Fill only password field with a valid customer password 4. Press on Login button
TEST DATA	Username: empty Password: Chathura123
EXPECTED RESULT	Should display the same Error Message which displayed previously as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages displayed as a toast message with very clear outcomes and the same error displayed in the above test case has received here also.
STATUS (PASS/FAIL)	Pass

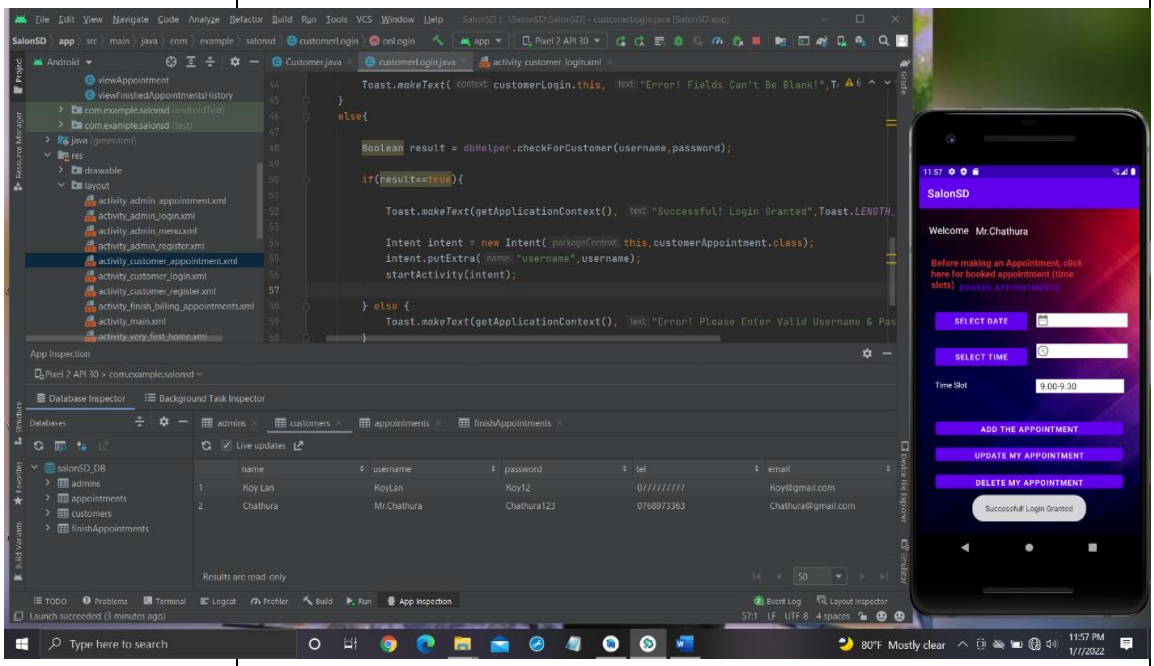
TC6.3 - Check by wrong username or password

TEST CASE ID	TC6.3.1
TEST CASE NAME	Check with wrong username and correct password
TEST SUMMARY	Checking correctly filled password field validation whereas username is wrong
TEST STEPS	4. Fill only password field with a valid customer password 5. Fill username field with a wrong username 6. Press on Login button
TEST DATA	Username: Mr.Chathu Password: Chathura123
EXPECTED RESULT	Should display an Error Message as Please enter valid username & password
ACTUAL RESULT	
CONCLUSION	The expected error message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

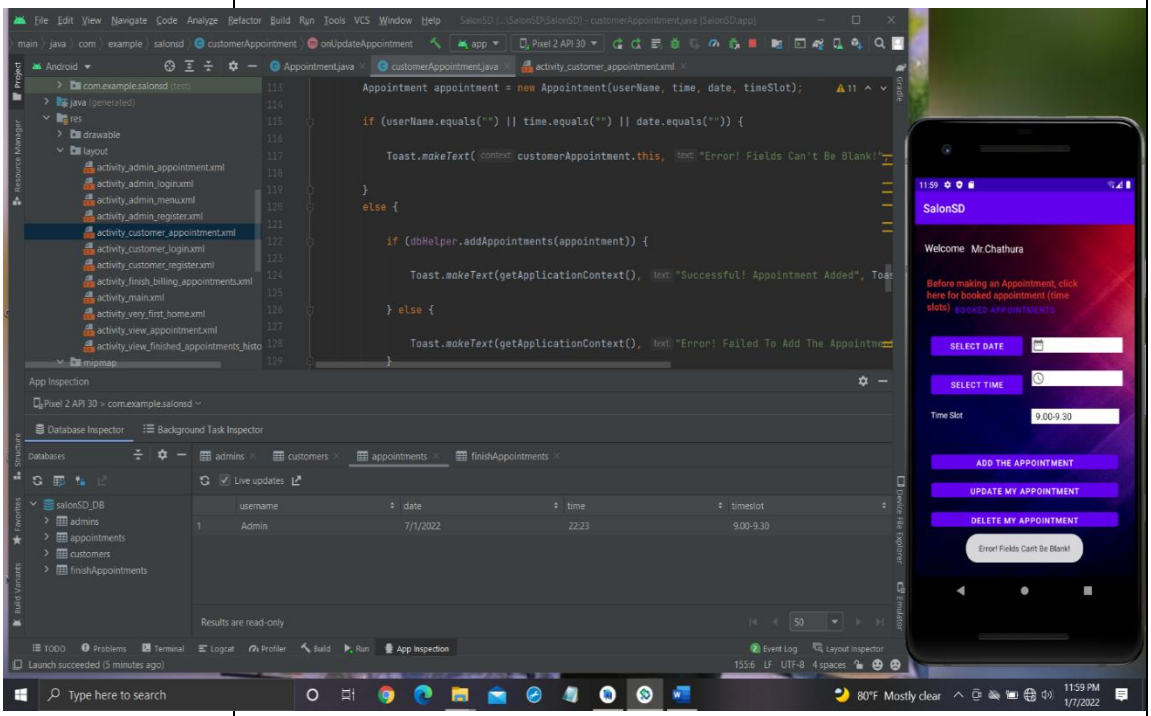
TEST CASE ID	TC6.3.2
TEST CASE NAME	Check with correct username and wrong password
TEST SUMMARY	Checking correctly filled username field validation whereas password is wrong
TEST STEPS	4. Fill only username field with a valid customer username 5. Fill password field with a wrong password 6. Press on Login button
TEST DATA	Username: Mr.Chathura Password: Chathura1
EXPECTED RESULT	Should display an Error Message as Please enter valid username & password
ACTUAL RESULT	
CONCLUSION	The expected error messages displayed as a toast message with very clear outcomes and the same error displayed in the above test case has received here also.
STATUS (PASS/FAIL)	Pass

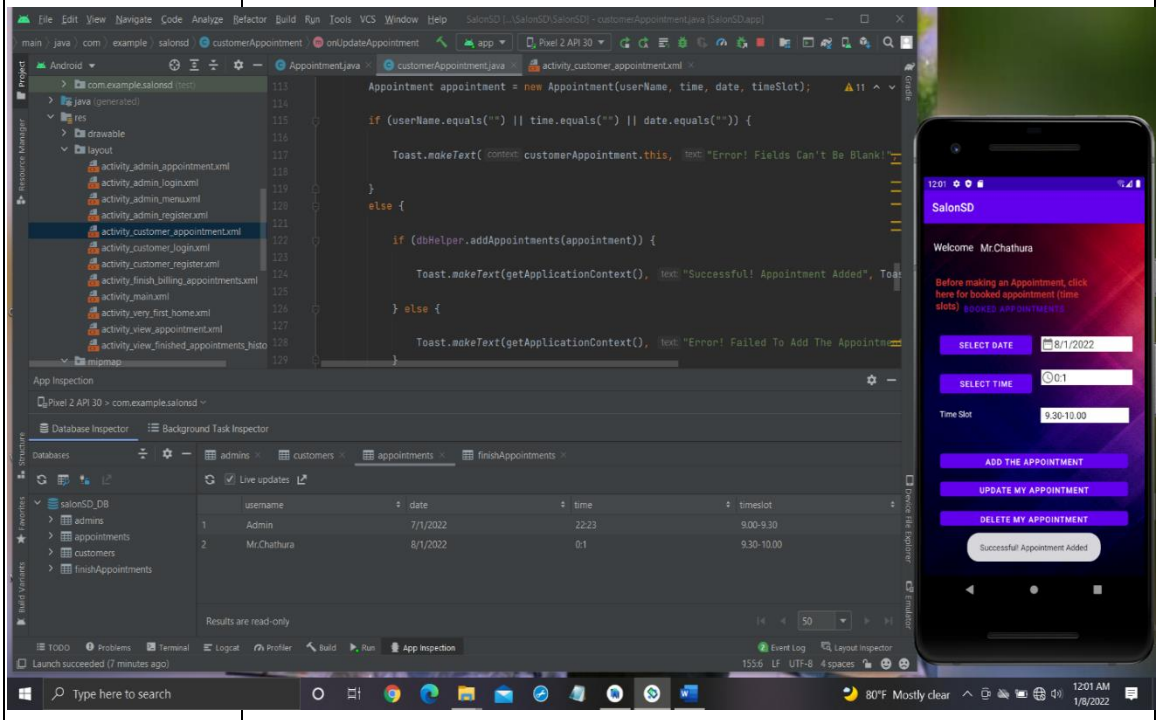
TEST CASE ID	TC1.4
TEST CASE NAME	Check Login working without any empty field and with saved & correct login credentials
TEST SUMMARY	Checking whether correctly filled username and password is working or not using registered username and password
TEST STEPS	4. Fill username field with a valid customer username 5. Fill password field with a valid customer password 6. Press on Login button
TEST DATA	Username: Mr.Chathura Password: Chathura123
EXPECTED RESULT	Customer Appointment interface should be loaded along with a message as Login granted successfully
ACTUAL RESULT	

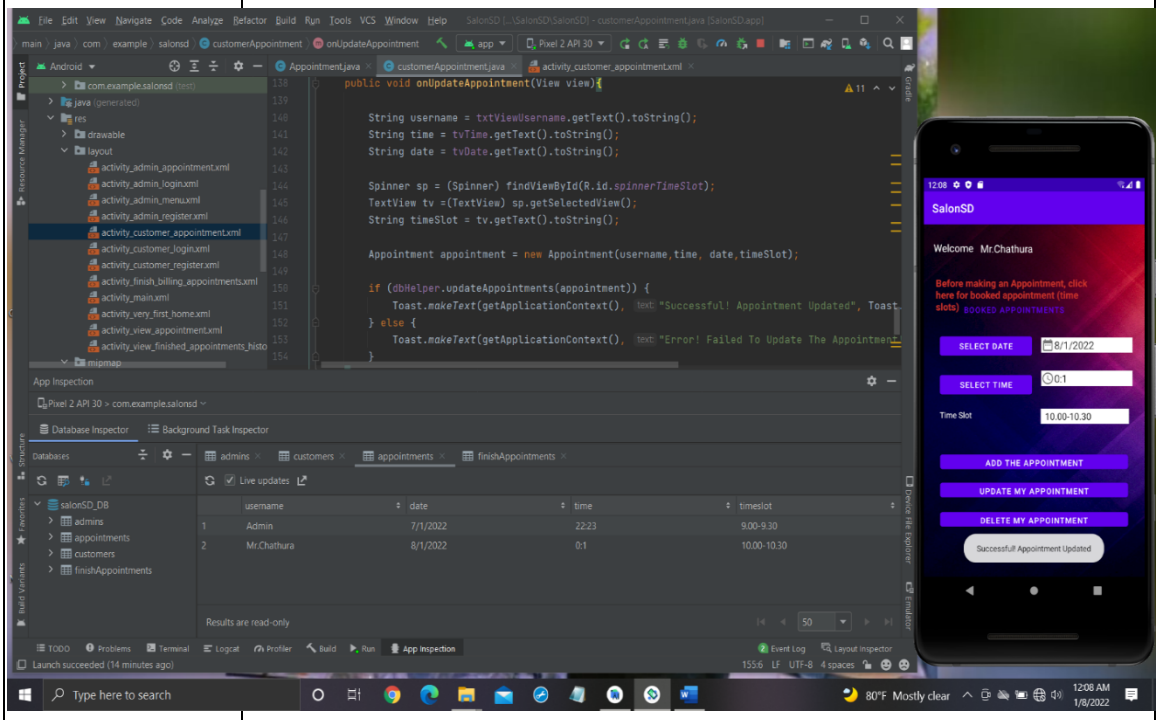


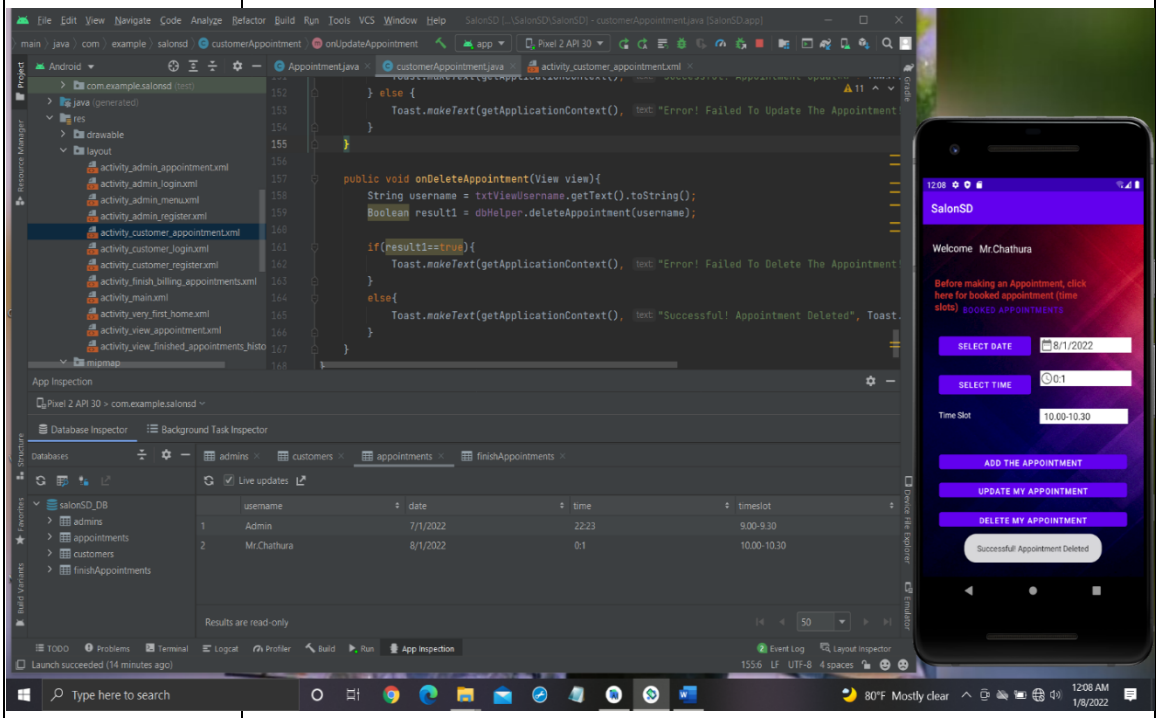
	
CONCLUSION	The expected successful message displayed as a toast message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TC7 - Customer Appointment

TEST CASE ID	TC7.1
TEST CASE NAME	Check Add Appointment with empty fields
TEST SUMMARY	Checking empty fields validation in Customer Appointment interface
TEST STEPS	Press on Add Appointment button (without filling any field)
TEST DATA	Empty fields
EXPECTED RESULT	Should display an Error Message as Fields can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC7.2
TEST CASE NAME	Check Add Appointment working without any empty field
TEST SUMMARY	Checking correctly working of Add Appointment in Customer Appointment interface
TEST STEPS	3. Have to fill all fields 4. Press on Add Appointment button
TEST DATA	Date: Current Date Time: Current Time Time Slot: Select 9.30-10.00 Time Slot from the Spinner
EXPECTED RESULT	Should display a message as Appointment added successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC7.3
TEST CASE NAME	Check Update Appointment working without any empty and updated fields
TEST SUMMARY	Checking correctly working of Update Appointment in Customer Appointment interface
TEST STEPS	4. Have to fill all fields 5. Change the fields that are want to be changed 6. Press on Update Appointment button
TEST DATA	Date: Change the date Time: Change the time Time Slot: Change the Time Slot(9.30-10.00) to the Time Slot(10.00-10.30) from the Spinner
EXPECTED RESULT	Should display a message as Appointment updated successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC7.4
TEST CASE NAME	Check Delete Appointment working
TEST SUMMARY	Checking correctly working of Delete Appointment in Customer Appointment interface
TEST STEPS	3. Have to fill all fields (Not Necessary) 4. Press on Delete Appointment button
TEST DATA	Delete button
EXPECTED RESULT	Should display a message as Appointment deleted successfully
ACTUAL RESULT	
CONCLUSION	The expected successful message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

References

Arnicans, G., 1998. *Application generation for the simple.* [Online]
Available at: https://www.researchgate.net/profile/Guntis-Arnicans-2/publication/268274019_Application_generation_for_the_simple_database_browser_based_on_the_ER_diagram/links/550469310cf2d60c0e67860d/Application-generation-for-the-simple-database-browser-based-on-the-
[Accessed 3 December 2021].

Bell, D., 2004. *UML basics: The sequence diagram.* [Online]
Available at: http://www.csun.edu/~twang/380/Slides/SequenceDiagram.pdf
[Accessed 10 December 2021].

Dondeti, S. N. &. J., 2012. *BLACK BOX AND WHITE BOX TESTING.* [Online]
Available at: https://www.researchgate.net/profile/S-Nidhra/publication/276198111_Black_Box_and_White_Box_Testing_Techniques_-_A_Literature_Review/links/570e313f08ae2b772e46aa40/Black-Box-and-White-Box-Testing-Techniques-A-Literature-Review.pdf
[Accessed 15 December 2021].

JavaTpoint., 2011-2021. *JavaTpoint - UML Class Diagram.* [Online]
Available at: https://www.javatpoint.com/uml-class-diagram
[Accessed 8 December 2021].
smartdraw, 1994-2022. *Smart Draw - Use Case Diagram.* [Online]
Available at: https://www.smartdraw.com/use-case-diagram/
[Accessed 5 December 2021].

Conclusion

However, I could get many advantages from this course work such as how to solve doubts and issues, how to gather requirements needed, how to do the analyzation properly, how to manage time effectively, how to overcome from errors which are arising when writing the code and so on. Finally…with all above mentioned details and me proper time management, I could prepare a successful document to complete this assignment. Thus, I hope this assignment been a great help for me to get learnt about Mobile Application Development and to prove that I have successfully completed my seventeenth assignment in my HD Program. Moreover, during this Mobile Application Development assessment, I learnt about many more things which are essential in designing and about coding while learning Java programming language more and many more other things. Developing of this mobile application for SD Saloon gave me the chance to try my new skills in practice. Specially this recalled my knowledge for drawing UML diagrams and Java coding. So, I have built the application in a user-friendly manner including admin registration and admin login, customer registration and login, admin menu, make appointment, view appointment, finish appointments, etc. While doing this project I also gained deep understanding on programming/coding and how it can be implemented in real life situations as now I have the experience in creating a Mobile Application for SD Salon. Thus, I believe that this was a great chance for me to improve my mobile application developing and computer programming skills.