Property Management System

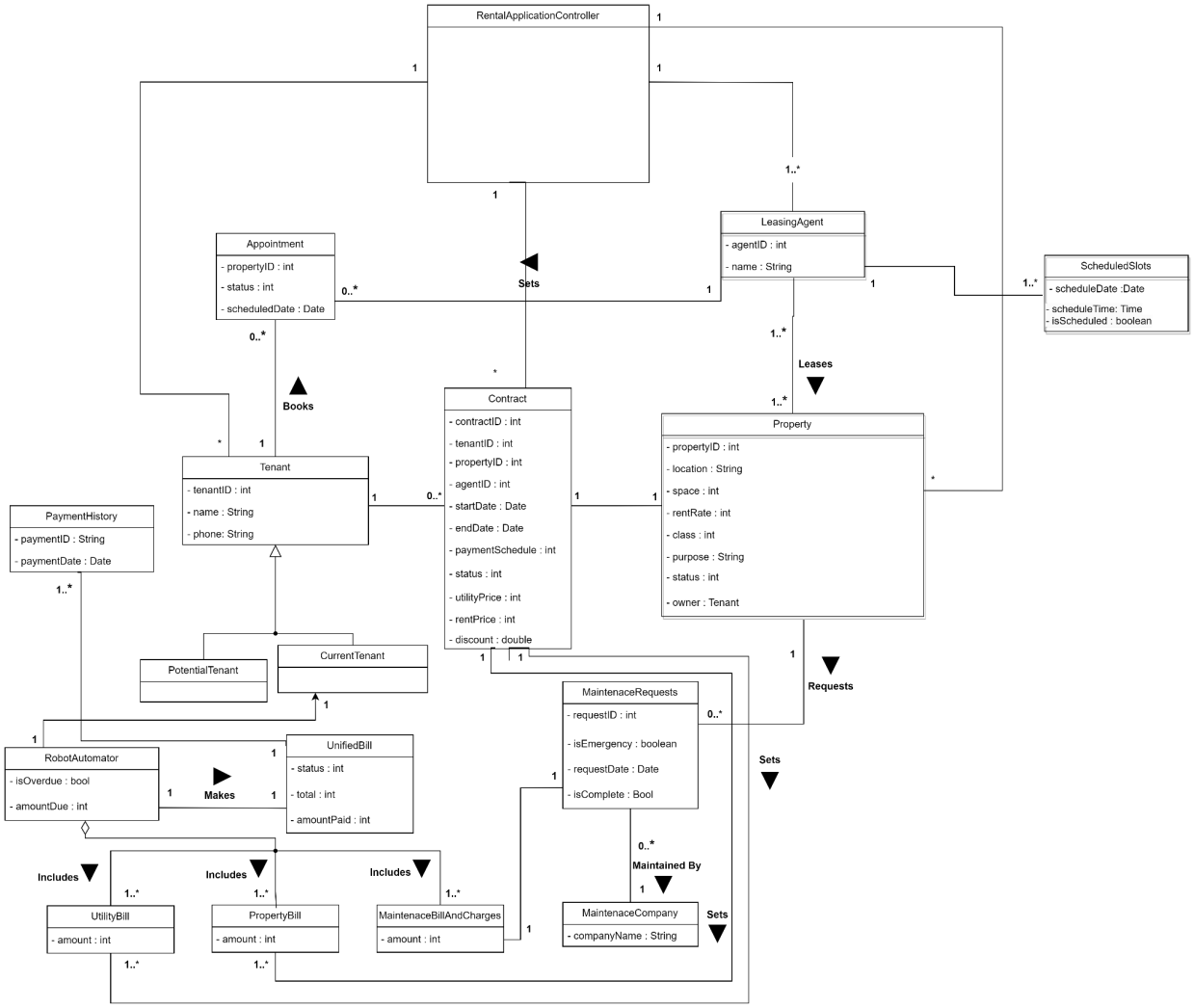
Deliverable 6: Testing Document

COE 420 Software Engineering
American University of Sharjah
College of Engineering
Computer Science and Engineering

Team DevOps	
Abdosalam Azmi	b00088275
Bingxuan Li	b00088619
Harish Menon	b00087415
Muhammad Ahmer	b00087698
Muhammad Zuhair Uddin	b00090000

Table of Contents

Updated Domain Model



Blackbox Testing

Equivalence Partioning With Boundary Value Analysis (BVA)

Conditions	Valid Partitions	Tags	Invalid Partitions	Tags	Valid boundaries	Tags	Invalid boundaries	Tags
Time Slot	Any available time slots	V1	No timeslot ——— Timeslot is not available	X1 X2	Earliest Timing ——— Last Timing	B1 B2	Slot timing that has already passed ——— 1 year + 1 day in the future	D1 D2
Leasing Agent	Any available agents	V2	No agent ——— agent is not availables	X3 X4	First Agent ——— Last Agent	B3 B4	———	
Confirmation of Appointment	Appointment details are valid	V3	Invalid Information ——— Cancel Appointment Confirmation	X5 X6	———		———	

BlackBox Test Cases

Test Cases	Description	Expected Outcome	Errors	Tags Covered
1	Time Slot: 17/5/2023 15:20 Agent Selected: Zuhair	Appointment has been confirmed	N/A	V1, V2, V3, B1, B2
2	Time Slot: 14/5/2023 2:20 Agent Selected: Harish	Appointment has been confirmed	N/A	V1, V2, V3, B3, B4
3	Time Slot: 18/5/2023 8:00 Agent Selected: Ahmer	Cannot select	Invalid Timeslot	X1, X3, X5

4	Time Slot: 11/5/2023 1:30 Agent Selected: Bing	Cannot select	Already Booked	V2, X2
5	Time Slot: 21/5/2023 2:30" Agent Selected: Harish	Cannot select	Valid Time, but not for the given Agent	V1, X4
6	Time Slot: 13/5/2023 4:20 Agent Selected: Harish	Cannot Select	Slot Time has Already Passed	V2, D1
7	Time Slot: 23/5/2024 4:20 Agent Selected: Harish	Cannot Select	Too far into the future, overflow	V2, D2
8	Time Slot: 1/5/2023 15:20	Cancelled Appointment	User cancelled confirmation of Appointment	X6

Whitebox Testing

Multiple Condition Coverages (MCC):

Criterion:

1. Time Slot
2. Leasing Agent
3. Confirmation

Test Case Number	Time Slot	Leasing Agent	Confirmation	Appointment Is Booked	Error States
1	Yes	Yes	Yes	Yes	No
2	Yes	Yes	No	No	No
3	Yes	No	Yes	No	Yes
4	Yes	No	No	No	No
5	No	Yes	Yes	No	Yes
6	No	Yes	No	No	No
7	No	No	Yes	No	Yes
8	No	No	No	No	No

Definitions

Valid Data:

- If the test data has a date and time that is earlier than today's date and time, then the time slot is not shown in the drop-down box.
- If the time slot under a specific leasing agent is booked by another tenant, then the time slot is not shown in the drop-down box.
- The timings of bookings must be specific to the selected leasing agent.

Entry/Exit Criterion

Entry	**Exit**
User selects “Book Appointment”	System displays the book appointment menu
User selects a given timing from the “Leasing Agents” drop down box	Gives a list of available Leasing Agents the user can select
User selects a given timing from the “Timings” drop down box	Gives a list of available Timings the user can select
User selects a preferred Leasing Agent	Leasing Agents is set in the Appointment
User selects a preferred Timing	TIming is set in the Appointment
User selects “Confirm” Button	Confirms the Appointment, and sends notification to the user and to the tenant

Component Testing

Component #1: LeasingAgent Drop-Down Box

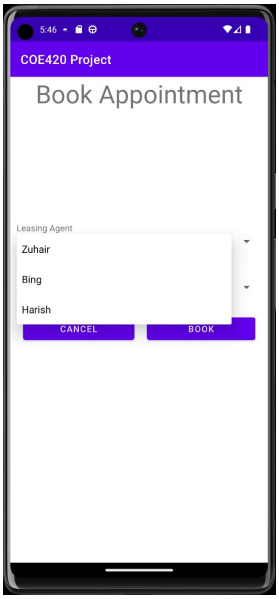
The code retrieves the leasing agents from the given data specified and shows it in the app.

```
leasingAgents = retrieveListOfLeasingAgents();
// Sets drop-down box
ArrayAdapter<LeasingAgent> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item, leasingAgents);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
leasingAgentSpinner.setAdapter(adapter);


private ArrayList<LeasingAgent> retrieveListOfLeasingAgents()
{
   ArrayList<LeasingAgent> agents = new ArrayList<>();
   agents.add(app.getZuhair());
   agents.add(app.getBing());
   agents.add(app.getHarish());
   return agents;
}
```

Test data code:

```
zuhair = new LeasingAgent(0, "Zuhair", zuhairTimings);
bing = new LeasingAgent(7, "Bing", bingTimings);
harish = new LeasingAgent(21, "Harish", harishTimings);
```

Screenshot of the successful retrieval of the Leasing Agents:

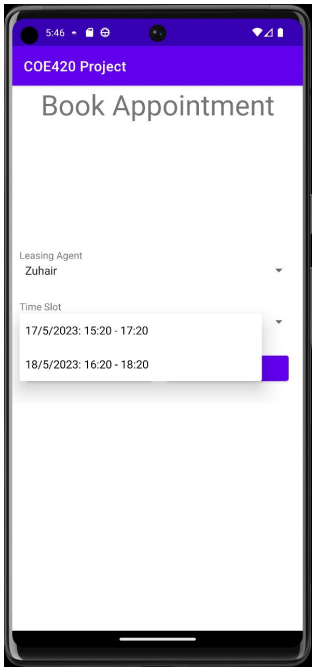


Component #2: ScheduledSlot Drop-Down Box

The code will retrieve the scheduled slots available for the selected leasing agent and shows it in the app.

```
LeasingAgent agent = (LeasingAgent) leasingAgentSpinner.getSelectedItem();
ArrayList<ScheduledSlot> slots = agent.retrieveScheduleSlot();
// Sets drop-down box
ArrayAdapter<ScheduledSlot> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item, slots);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
timeSlotSpinner.setAdapter(adapter);
```

Screenshot of the successful retrieval of the Scheduled slots:



Component #3: Book Button and notification

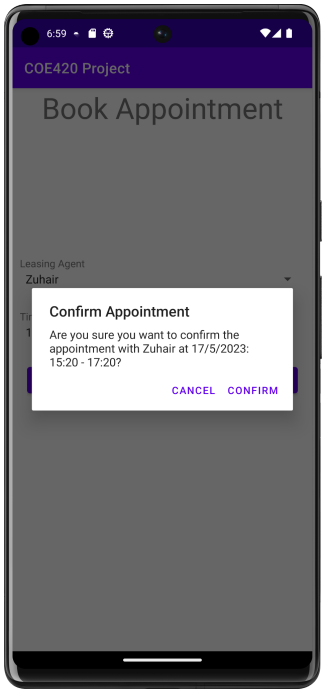
The code must retrieve the information that is selected and first ask for confirmation on the selection, then once it is confirmed, the code should send notification to the mobile regarding the appointment.

```
 @Override
    public void onClick(View v)
    {
        if(v.getId() == R.id.bookButton)
        {
            ScheduledSlot slot = (ScheduledSlot)
timeSlotSpinner.getSelectedItem();
            LeasingAgent agent = (LeasingAgent)
leasingAgentSpinner.getSelectedItem();
            if(agent != null && slot != null)
```

```
            {
                AlertDialog.Builder builder = new
AlertDialog.Builder(BookAppointment.this);
                builder.setTitle("Confirm Appointment");
                builder.setMessage("Are you sure you want to confirm the
appointment with " + agent.getName() + " at " + slot.toString() + "?");
                builder.setPositiveButton("Confirm", new
DialogInterface.OnClickListener()
                {
                    @Override
                    public void onClick(DialogInterface dialog, int id)
                    {
                        agent.createAppointment(app.getPropertyID(),
slot.getScheduleDateTimeStart());
                        agent.updateSchedule(slot);
                        sendNotification(agent.getName() + " - " +
slot.toString());
                        globalIntent.putExtra("agent", agent.getName());
                        globalIntent.putExtra("date", slot.toString());
                        globalIntent.putExtra("ValidRequest", true);
                        setResult(0, globalIntent);
                        finish();
                    }
                });
                builder.setNegativeButton("Cancel", null);
                builder.setCancelable(true);
                builder.create();
                builder.show();
            }
        } else if(v.getId() == R.id.cancelButton)
        {
            finish();
        }
    }

    private void sendNotification(String text)
    {
        int icon = R.drawable.ic_launcher_foreground;
        CharSequence tickerText = "Book Appointment";
        CharSequence contentTitle = "Appointment Created";
        CharSequence contentText = text;
        NotificationChannel notificationChannel =
                new NotificationChannel("Channel_ID", "My Notifications",
NotificationManager.IMPORTANCE_DEFAULT);
        NotificationManager manager = (NotificationManager)
getSystemService(this.NOTIFICATION_SERVICE);
        manager.createNotificationChannel(notificationChannel);
        Notification notification = new NotificationCompat
                .Builder(this, "Channel_ID")
                .setSmallIcon(icon)
                .setTicker(tickerText)
                .setContentTitle(contentTitle)
                .setContentText(contentText)
                .setAutoCancel(false)
                .setChannelId("Channel_ID")
                .build();
        manager.notify(1, notification);
```

```
}
```

Screenshot of the successful retrieval of the selected information to be booked:



Integration/Functional Testing

Functional #1: ScheduledSlot Drop-Down Box showing slots that are not booked yet

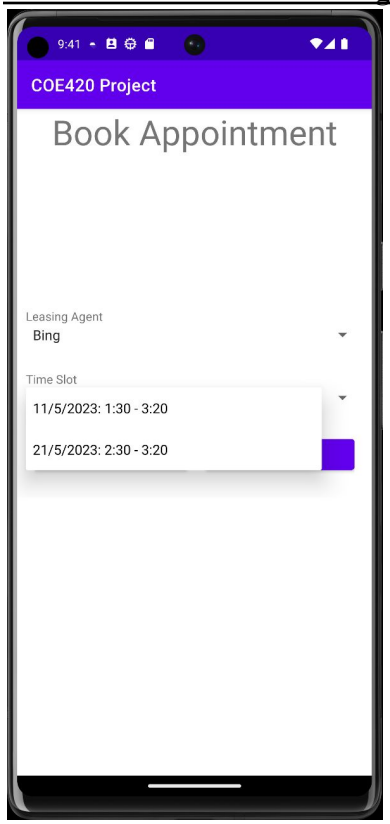
When the schedule slot is retrieved, the program code checks whether the status of the appointment is not booked and only shows those time slots in the app. Those slots which are booked are not shown in the app.

Test Data :

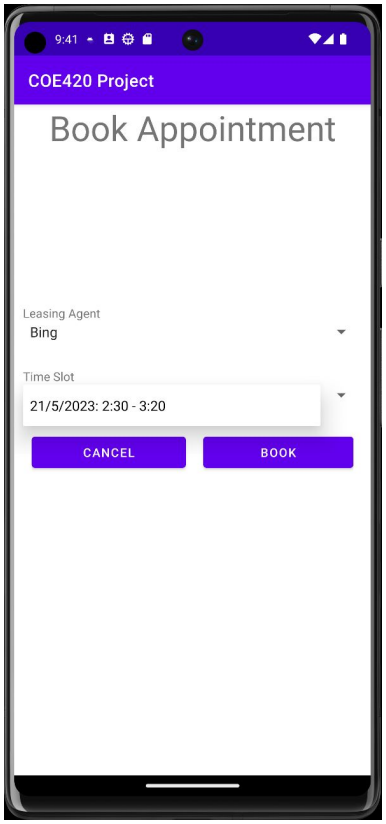
```
@Override
    public void onCreate()
    {
        super.onCreate();
        ArrayList<ScheduledSlot> zuhairTimings = new ArrayList<>();
        ArrayList<ScheduledSlot> bingTimings = new ArrayList<>();
        ArrayList<ScheduledSlot> harishTimings = new ArrayList<>();
        try
        {
            SimpleDateFormat dateTimeFormat = new
SimpleDateFormat("d/M/yyyy H:mm");
            zuhairTimings.add(new
ScheduledSlot(dateTimeFormat.parse("17/5/2023 15:20"),
dateTimeFormat.parse("17/5/2023 17:20")));
            zuhairTimings.add(new
ScheduledSlot(dateTimeFormat.parse("18/5/2023 16:20"),
dateTimeFormat.parse("18/5/2023 18:20")));
            zuhairTimings.add(new
ScheduledSlot(dateTimeFormat.parse("1/5/2023 15:20"),
dateTimeFormat.parse("1/5/2023 17:20")));
            bingTimings.add(new
ScheduledSlot(dateTimeFormat.parse("11/5/2023 1:30"),
dateTimeFormat.parse("11/5/2023 3:20")));
            bingTimings.add(new
ScheduledSlot(dateTimeFormat.parse("21/5/2023 2:30"),
dateTimeFormat.parse("21/5/2023 3:20")));
            harishTimings.add(new
ScheduledSlot(dateTimeFormat.parse("13/5/2023 1:20"),
dateTimeFormat.parse("13/5/2023 5:20")));
            harishTimings.add(new
ScheduledSlot(dateTimeFormat.parse("14/5/2023 2:20"),
dateTimeFormat.parse("14/5/2023 5:20")));
        } catch(ParseException e)
        {

        }
        zuhair = new LeasingAgent(0, "Zuhair", zuhairTimings);
        bing = new LeasingAgent(7, "Bing", bingTimings);
        harish = new LeasingAgent(21, "Harish", harishTimings);
```

```
}
```

Screenshot of the code running with the above test data before the booking:



Screenshot of the code running with the above test data after the booking of the first slot:



Functional #2: ScheduledSlot Drop-Down Box showing only future slots

When the scheduled slot is retrieved, the code below compares the start time with the current date from the data provided. If the date and time is in the past, the scheduled slot will not show that time slot, otherwise it will be shown in the drop-down box.

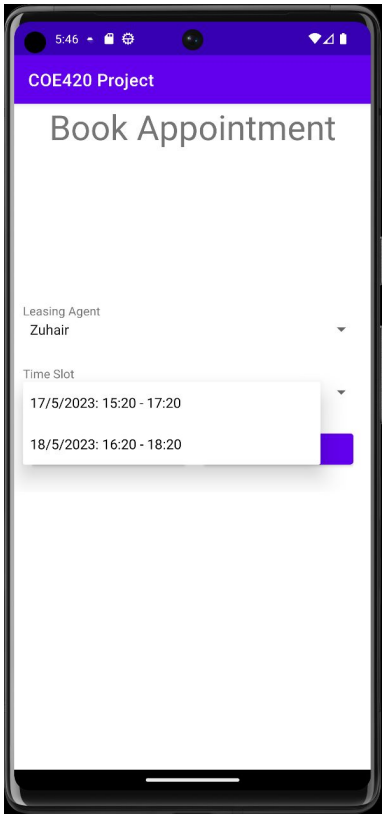
```
public ArrayList<ScheduledSlot> retrieveScheduleSlot()
{
   ArrayList<ScheduledSlot> availableSlots = new ArrayList<>();
   LocalDateTime localDateTime = LocalDateTime.now();
   ZonedDateTime zdt = ZonedDateTime.of(localDateTime, ZoneId.systemDefault());
   long currentDateTime = zdt.toInstant().toEpochMilli();
   for(ScheduledSlot slot : slots)
   {
       if(slot.getScheduleDateTimeStart().getTime() >= currentDateTime &&
!slot.isScheduled())
       {
           availableSlots.add(slot);
       }
   }
   return availableSlots;


}
```

Test Data :

```
 @Override
    public void onCreate()
    {
        super.onCreate();
        ArrayList<ScheduledSlot> zuhairTimings = new ArrayList<>();
        ArrayList<ScheduledSlot> bingTimings = new ArrayList<>();
        ArrayList<ScheduledSlot> harishTimings = new ArrayList<>();
        try
        {
            SimpleDateFormat dateTimeFormat = new
SimpleDateFormat("d/M/yyyy H:mm");
            zuhairTimings.add(new
ScheduledSlot(dateTimeFormat.parse("17/5/2023 15:20"),
dateTimeFormat.parse("17/5/2023 17:20")));
            zuhairTimings.add(new
ScheduledSlot(dateTimeFormat.parse("18/5/2023 16:20"),
dateTimeFormat.parse("18/5/2023 18:20")));
            zuhairTimings.add(new
ScheduledSlot(dateTimeFormat.parse("1/5/2023 15:20"),
dateTimeFormat.parse("1/5/2023 17:20")));
            bingTimings.add(new
ScheduledSlot(dateTimeFormat.parse("11/5/2023 1:30"),
dateTimeFormat.parse("11/5/2023 3:20")));
            bingTimings.add(new
ScheduledSlot(dateTimeFormat.parse("21/5/2023 2:30"),
dateTimeFormat.parse("21/5/2023 3:20")));
            harishTimings.add(new
ScheduledSlot(dateTimeFormat.parse("13/5/2023 1:20"),
dateTimeFormat.parse("13/5/2023 5:20")));
            harishTimings.add(new
ScheduledSlot(dateTimeFormat.parse("14/5/2023 2:20"),
dateTimeFormat.parse("14/5/2023 5:20")));
        } catch(ParseException e)
        {

        }
        zuhair = new LeasingAgent(0, "Zuhair", zuhairTimings);
        bing = new LeasingAgent(7, "Bing", bingTimings);
        harish = new LeasingAgent(21, "Harish", harishTimings);
    }
```

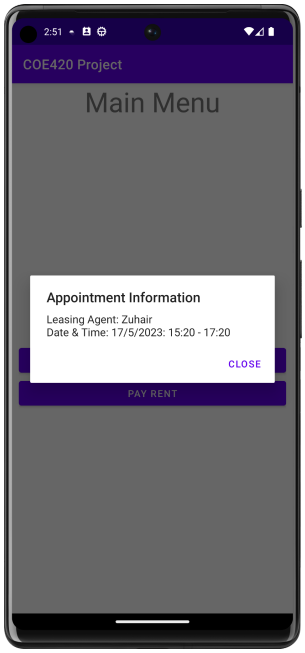
Screenshot of the code running with the above test data:

This shows that the code is not taking any time slots that are before the current date.


5:46
COE420 Project
Book Appointment
Leasing Agent
Zuhair
Time Slot
17/5/2023: 15:20 - 17:20
18/5/2023: 16:20 - 18:20

System Testing

Functional: The system is able to book the appointment by creating an Appointment class and storing the Appointment details. This means that the appointment details can be stored and retrieved.



Non-functional: The system is accessible, so that it blocks out any user error.

Type	**Expected Outcome**
Functional	Booked Appointment Succesfully
Non-Functional	Prevents selection of invalid options