

SMART MEDICAL SOLUTION

BY

AMAN FERDOUS APU
ID: 161-15-6985

AND

FAHIMATUL ISLAM
ID: 161-15-7105

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Mr. Majidur Rahman
Lecturer
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

05 DECEMBER 2019

APPROVAL

This Project/internship titled “**Smart Medical Solution**”, submitted by Aman Ferdous Apu & Fahimatul Islam ,ID No: 161-15-6985 & 161-15-7105to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 05/12/2019

BOARD OF EXAMINERS



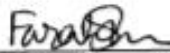
Dr. Syed Akhter Hossain
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



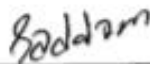
Abdus Sattar
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Farah Sharmin
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Saddam Hossain
Assistant Professor
Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Mr. Majidur Rahman , Lecturer , Department of CSE Daffodil International University.** We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Mr. Majidur Rahman
Lecturer
Department of CSE
Daffodil International University

Submitted by:



Aman Ferdous Apu
ID: 161-15-6985
Department of CSE
Daffodil International University



Fahimatul Islam
ID: 161-15-7105
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to Supervisor **Mr. Majidur Rahman** (Lecturer). Department of CSE Daffodil International University , Dhaka. Deep Knowledge & keen interest of our supervisor in the field of BSc in CSE Program to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain Head**, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents

ABSTRACT

This is an Android based project. There are many medical relatable applications but they are not as complete as ours. There are many types of applications that only sell medicines online and there are many who can go to the online site through the application. But there is no other application like us, because we are unique. We have two parts in the application and doctors and patients. With the help of our application, the patient can make an appointment to any doctor of his choice and the patient can see the doctor's profile. The patient will know the doctor's time and location. This allows the patient to know about any doctor he may need and to make an appointment without having to suffer. This will reduce patient time and suffering. There is another feature which is the ambulance service. With this, the patient can receive his services at the urgent moment. Again, the doctor can easily find the patient without any problem. In addition, if the doctor has his own ambulance, he can add it or if he has a known ambulance. Here the patient can also add an ambulance that is known to himself, which will benefit the people of the food as well. So we think that this is a very effective application that will benefit both the patient and the doctor. As a result, our medical sector will further improve.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
CHAPTER	
CHAPTER 01: INTRODUCTION	1-6
1.1 Project Introduction	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Expected Outcome	3
CHAPTER 02: BACKGROUND	4-5
2.1 Introduction	4
2.2 Related Work	4
2.3 My Project Work	4
2.4 Comparative Studies	5
2.5 Scope of Problems	5
2.6 Challenges	

CHAPTER 03: REQUIREMENT SPECIFICATION	6-17
3.1 Requirement Collection and Analysis	6
3.2 Functional Requirement	6
3.3 Non-Functional Requirement	6
3.4 Login process	7
3.5 Use Case	8
3.6 Logical Data Model	9
3.7 Design Requirements	9
3.8 Application Photo	10
CHAPTER 04: DESIGN SPECIFICATION	18-30
4.1 Implementation Requirement	18
4.2 Android Studio	18
4.3 Android SDK	18
4.4 Java Development Kit (JDK)	18
4.5 Android Virtual Device (AVD)	19
4.6 Interaction Design and UX	19
4.7 Code and Data base	20-30
CHAPTER 05: IMPLEMENTATION AND TESTING	31-32
5.1 Implementation of Database	
5.2 Testing	
5.3 Implementation of Front-end Design	
5.4 Implementation of java and XML code	

5.5 Testing Implementation

CHAPTER 06: CONCLUSION AND FUTURE SCOP

33

6.1 Conclusion

6.2 Scope for Future Developments

APPENDIX

REFERENCES

LIST OF FIGURES

FIGURES	PAGE
Figure: 3.4.1: patients login	7
Figure: 3.4.2: Doctors login	8
Figure: 3.5.1: Use case diagram	9
Figure: 3.8.4: Patients login	11-14
Figure: 3.8.5: Doctors Login	15-17
Figure 4.7.1: Ambulance & Appointment	20
Figure 4.7.2: Categories & Users	21
Figure 4.7.3: Android XML file & Android .Java file	22
Figure 4.7.4: Firestore Repository & Ambulance.Java	23-26
Figure 4.7.5: Appointment & Multiple Item	27-28
Figure 4.7.6: Resources & Specialty	29
Figure 4.7.7 Status & User.java	30
Figure 5.4: ER Diagram.	32

CHAPTER 1

INTRODUCTION

1.1 Project Introduction

This is an Android based project. There are many medical relatable applications but they are not as complete as ours. There are many types of applications that only sell medicines online and there are many who can go to the online site through the application. But there is no other application like us, because we are unique. We have two parts in the application and doctors and patients. With the help of our application, the patient can make an appointment to any doctor of his choice and the patient can see the doctor's profile. The patient will know the doctor's time and location. This allows the patient to know about any doctor he may need and to make an appointment without having to suffer. This will reduce patient time and suffering. There is another feature which is the ambulance service. With this, the patient can receive his services at the urgent moment. Again, the doctor can easily find the patient without any problem. In addition, if the doctor has his own ambulance, he can add it or if he has a known ambulance. Here the patient can also add an ambulance that is known to himself, which will benefit the people of the food as well. So we think that this is a very effective application that will benefit both the patient and the doctor. As a result, our medical sector will further improve.

1.2 Motivation

There is no application like us because we are unique. Most of the applications are using medicine and other problem but in our application we are working with patient and doctors. Here doctors can create their own profile where doctors will update their regular activities like where is the chamber and visiting time etc. Patients will also create their own profile and get any specialist doctor's appointment. There is also another feature which is ambulance service. Here doctors and patients will able to add their known ambulance name and contacts. So patients will able to know any information about any famous doctor and get their appointment without any problem and they can save their time and money. By this ambulance service doctors will able to know about their patient and if they have any ambulance then they can give service to the people. This reduced the suffering of the common people.

1.3 Objectives

- We have used some android tools. They are:-
 1. Play service.
 2. Google map.
 3. Firebase authentication.
 4. SDK authentication.

- For database we are used Firebase store and OTP authentication.

1.4 Expected Outcome

- You will be able to know about any famous doctor and their appointment and also their visiting times and visiting places.
- Patient will get ambulance service, where emergency services will be available at the moment of emergency
- Doctor will get information about their patients.
- New doctors, in particular, will be able to serve their patients
- Using this application they can save their time and money.

CHAPTER 02

BACKGROUND

2.1 Introduction

Mobile phone is now available and affordable for everyone. Most hospitals are getting digital so, this application will be more effective. Most of people in our country know the use of android system applications, that's why we should use this android application. It is easy to use and get the services.

2.2 Related Work

There are many medical based android applications that has been implemented yet. But what about perfection? If your work has no perfection then there is no work to do with it. Example: Sehaty, Google Fit, Patient Aid, Health solutions, Diseases Treatment Dictionary. But as I told you our project has some advance and faster technique than others. We have the attractive UI design too. We give some of the Screenshot images below:

2.3 My Project Work

There are many app in the play store, but all application are not completed. Some of them are complicated and some of them are restricted etc. But our application is best because we have given to our best in this application. Our user interface is more attractive than any other application. For this we used **JAVA, XML**. In this application we also used registration process. For the registration we used Firebase and authentication we used **GoogleAuth**. We used some featured in feature we will give more features.

2.4 Comparative Studies

In the Play Store, you can find many apps that incorporate some features into our app. Some applications are dynamic. The phone number allows the user to sign in to this application. There is another function - an ambulance. The physician and patient can enter the number and name of the ambulance.

2.5 Scope of Problems

First of all, this application is designed to be user friendly. For other game store apps, but you have some issues. But our app is useful for saving time. But our app is useful for saving time. We have many opportunities to improve our applications. We want to improve our apps every day and try to add more features. Currently, there are 154 million mobile phone users and most are using Android phones. We've tried to build something that is useful and useful to everyone. All types of Android devices support this application and each user has a unique identifier. This app is protected for everyone. I use a Google Account to sign in.

2.6 Challenges

If you do something good, there will be obstacles. We also have to face many obstacles. We are not high level experts so we have to work hard and face many problems. First our project was not running then we fixed it.

CHAPTER 03

REQUIREMENT SPECIFICATION

3.1 Requirement Analysis

Involves all the tasks that are conducted to identify the needs of different stakeholders . It must be documented, computable and tested. It can be functional and non-functional.

There are two parts

1. Android studio and codes.
2. Console Firebase.

3.2 Functional Requirements

Functional requirement is related to the technical function of the system. Our system have some functional requirement such as User phone number, their information, doctors information update, ambulance add option, Google account and maintaining database.

3.3 Non-Functional Requirements

Non Functional requirement is related to the security, viability, versatility, quality and ease of use. Our application have a great user interface for easy to use. Our application quality is very good to compare with other related application. In our application security system is very good. Our application will supported all kind of android operating system. Doctors can give input and update their information. Our application is android operating system based device with internet connection.

3.4 LogIn Process

As a Patient:

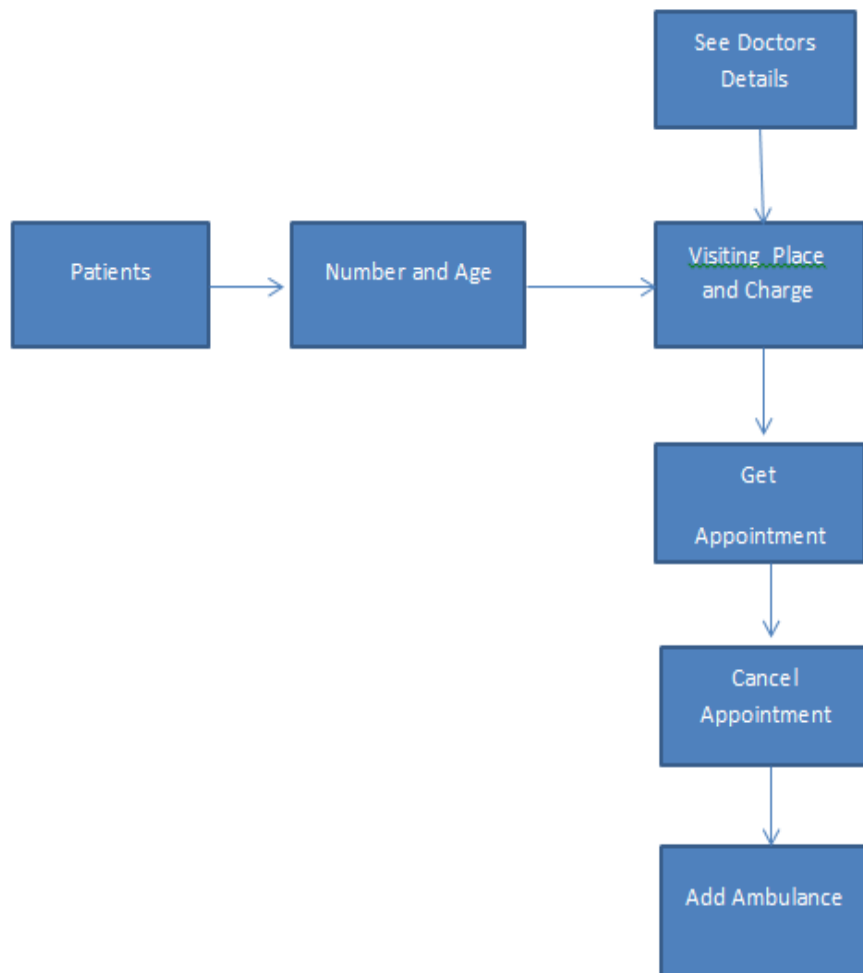


Figure: 3.4.1: patients login

3.4.1 As a doctor

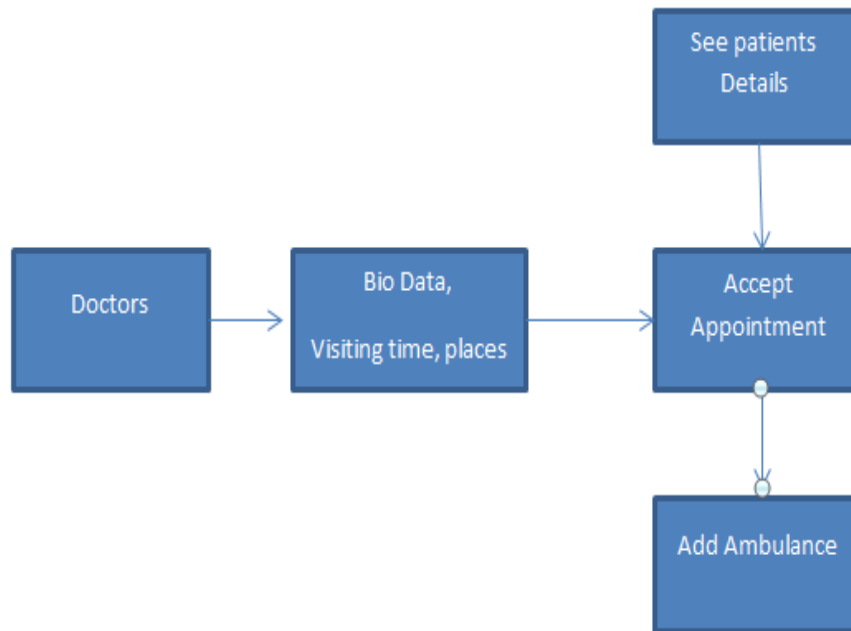


Figure: 3.4.2: Doctors login

3.5 Use case

The use case is a diagram of how the user is using the system. The diagram used includes the actors, the use cases and their relationships. The usage diagram reflects some of the features of the system. It is used to understand the user. The figure shows a flow chart of the enclosure used below.

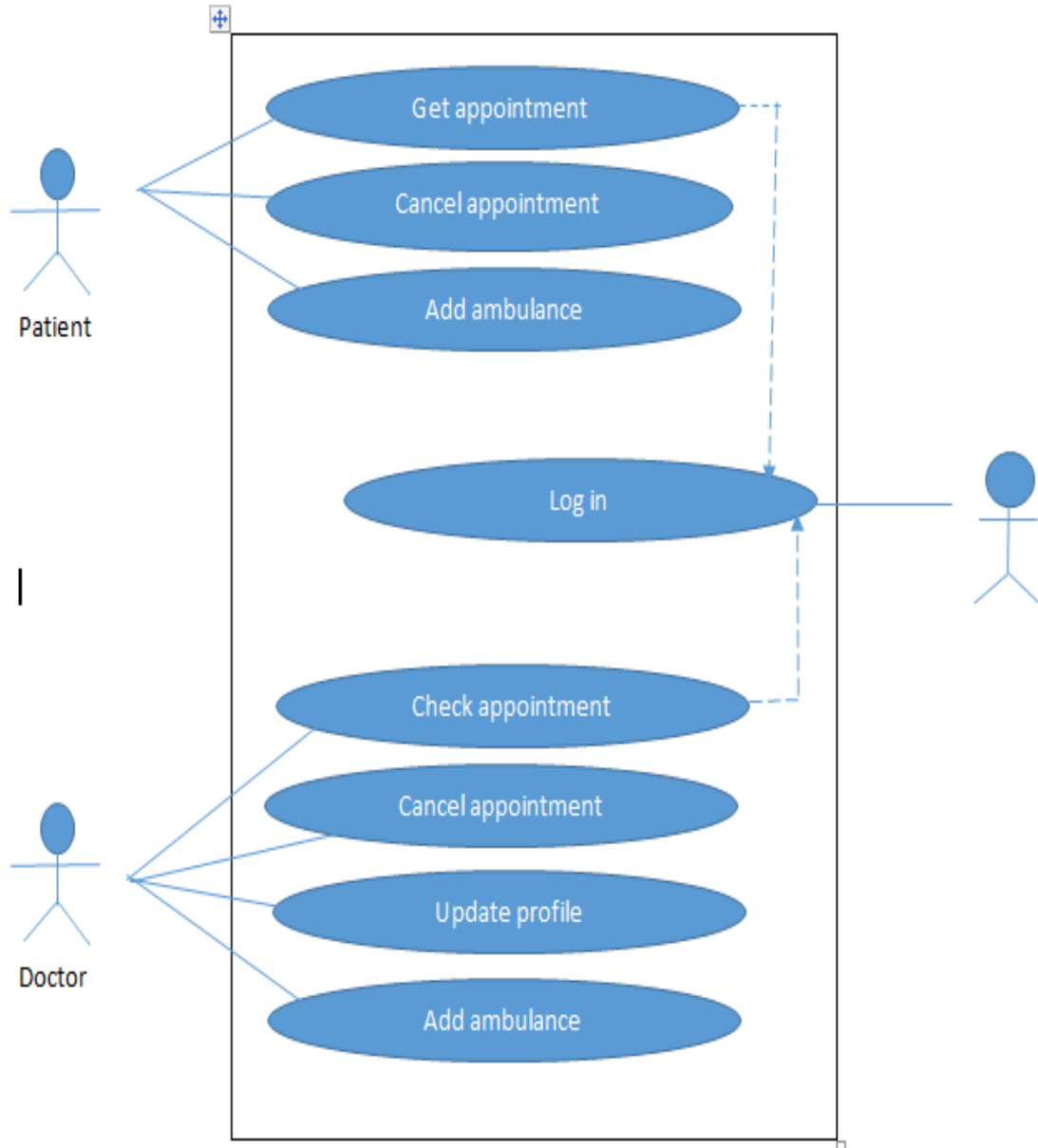


Figure:3.5.1: Use case diagram

3.6 Logical Data Model

The logical data presented is data templates for a specific domain that are freely stored in a specific part of the database as innovative capabilities, but data structures such as a list for students. We use object-oriented XML databases and XML user interfaces as background. My application is a database server and a fire database. Our application is designed to search for a student certificate. There is a database component.

3.7 Design Requirement

Project design requirements are different from other projects because my project is based on my specific problem statement and design experience. I developed the project for a specific segment.

Here some requirement are:

- In this system, we used a simple authentication for my user because of this application is public application.
- In my application I provide a simple user interface for easy to use.

3.8 APPLICATION PHOTO:

Patient Login

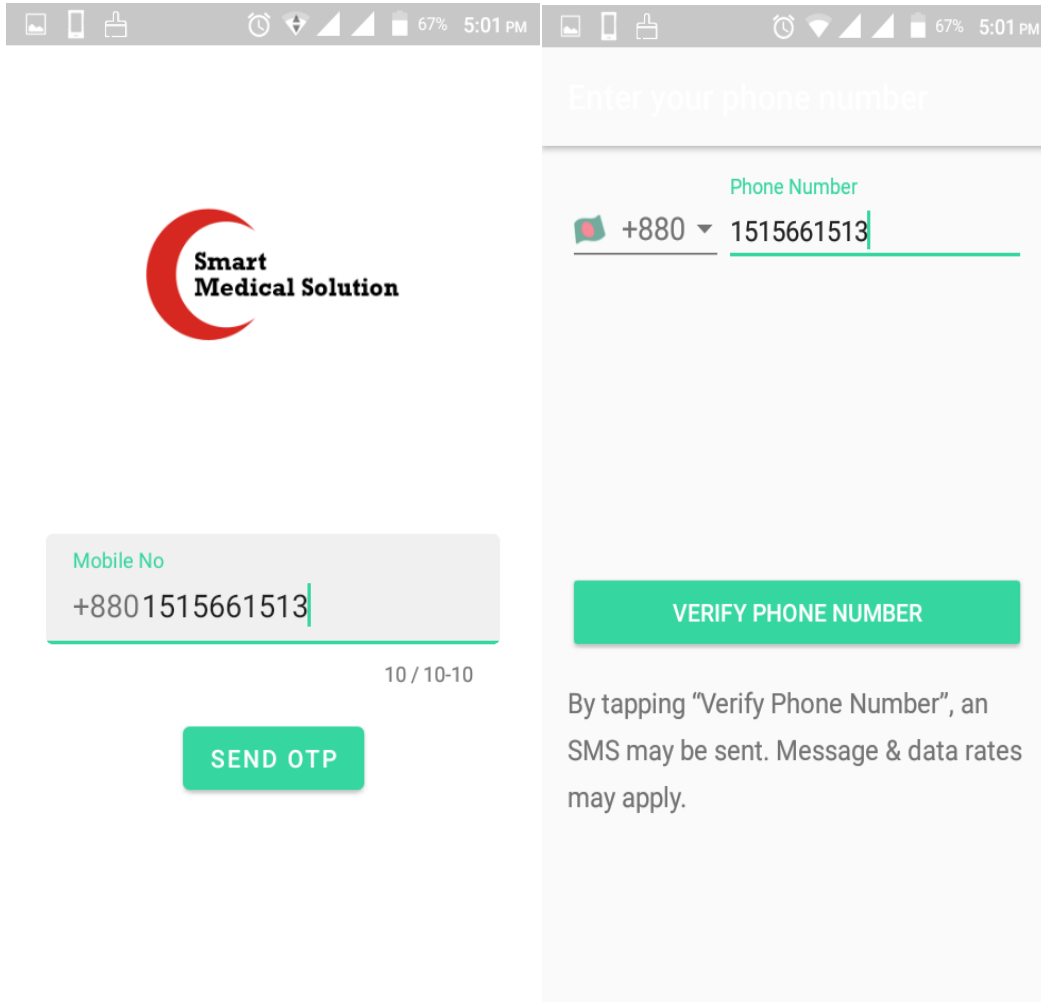


Fig 3.8.1: Patient login

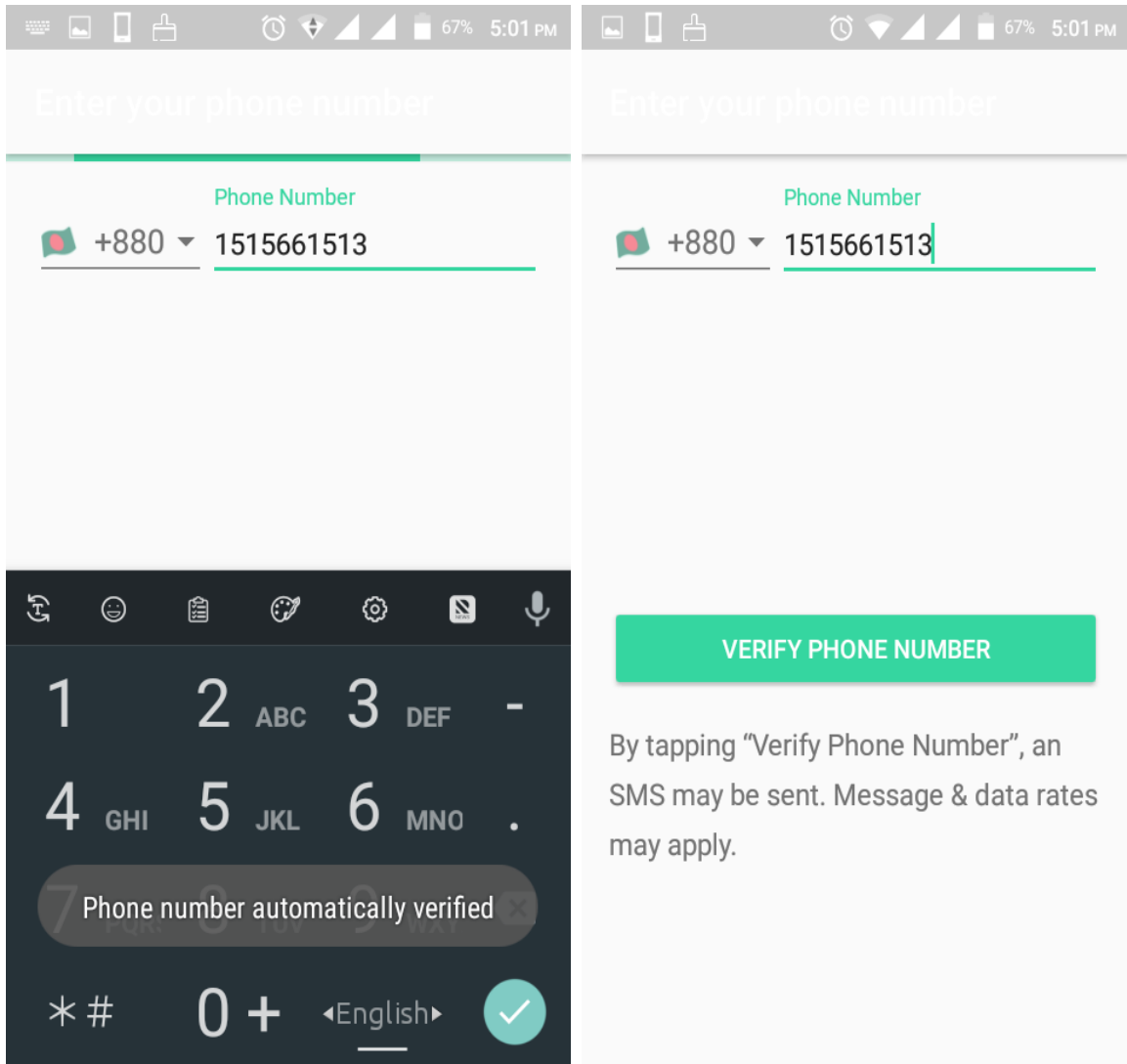


Fig 3.8.2: Patient login

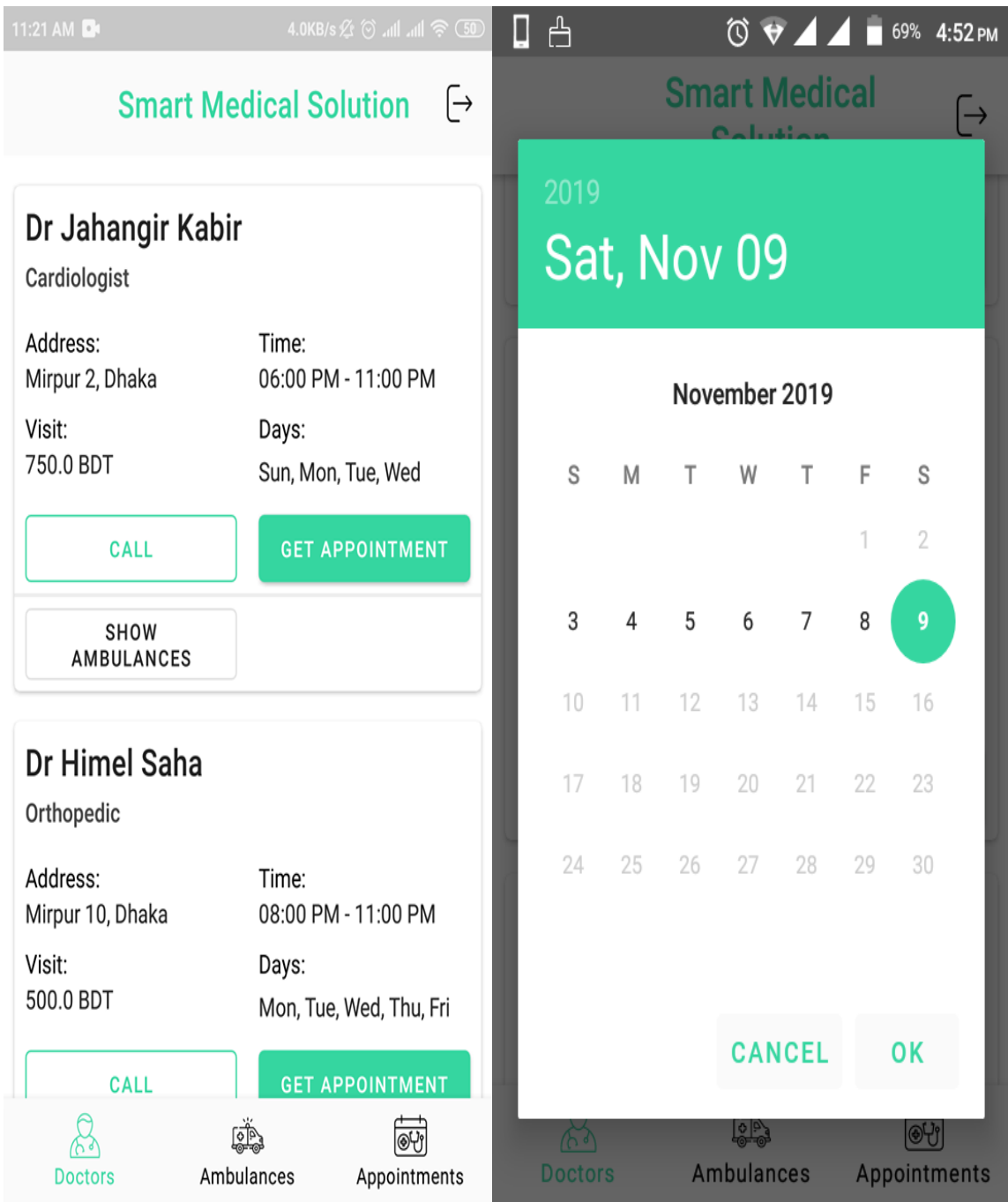


Fig 3.8.3: Patient login

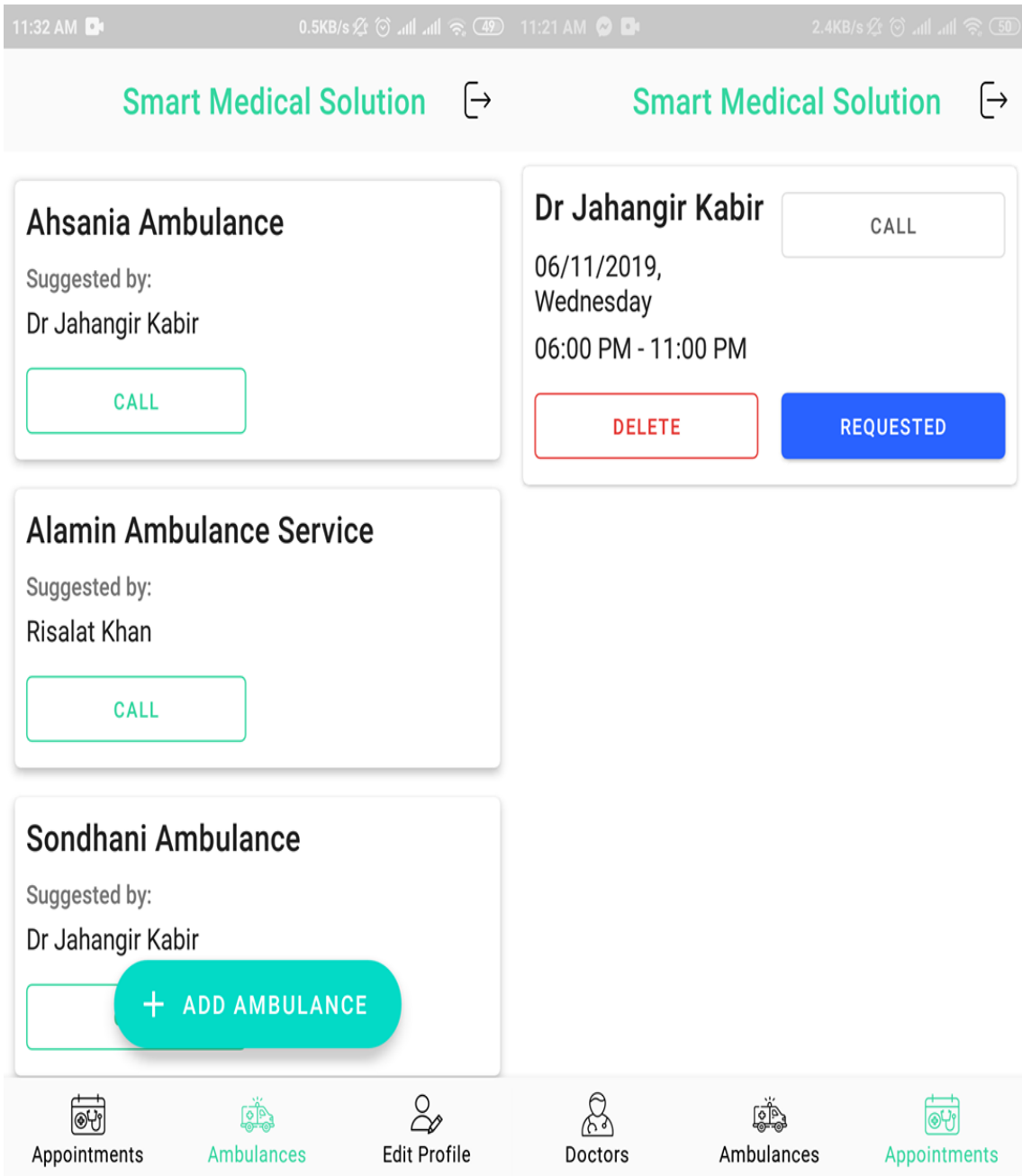



Fig 3.8.4: Patient login

Doctors Login :

11:21 AM 0.0KB/s 11:32 AM 1.4KB/s



Smart
Medical Solution

Mobile No

+8801679879688

10 / 10-10

SEND OTP

← Edit Profile

Full Name

Aman Ferdous

Age

27

Address

Shukrabad,Dhanmondi32

Speciality

Cardiologist

Starting start

07:00 PM

Ending start

09:00 PM

Weekdays

M T W T F S S

Fig 3.8.5: Doctors login

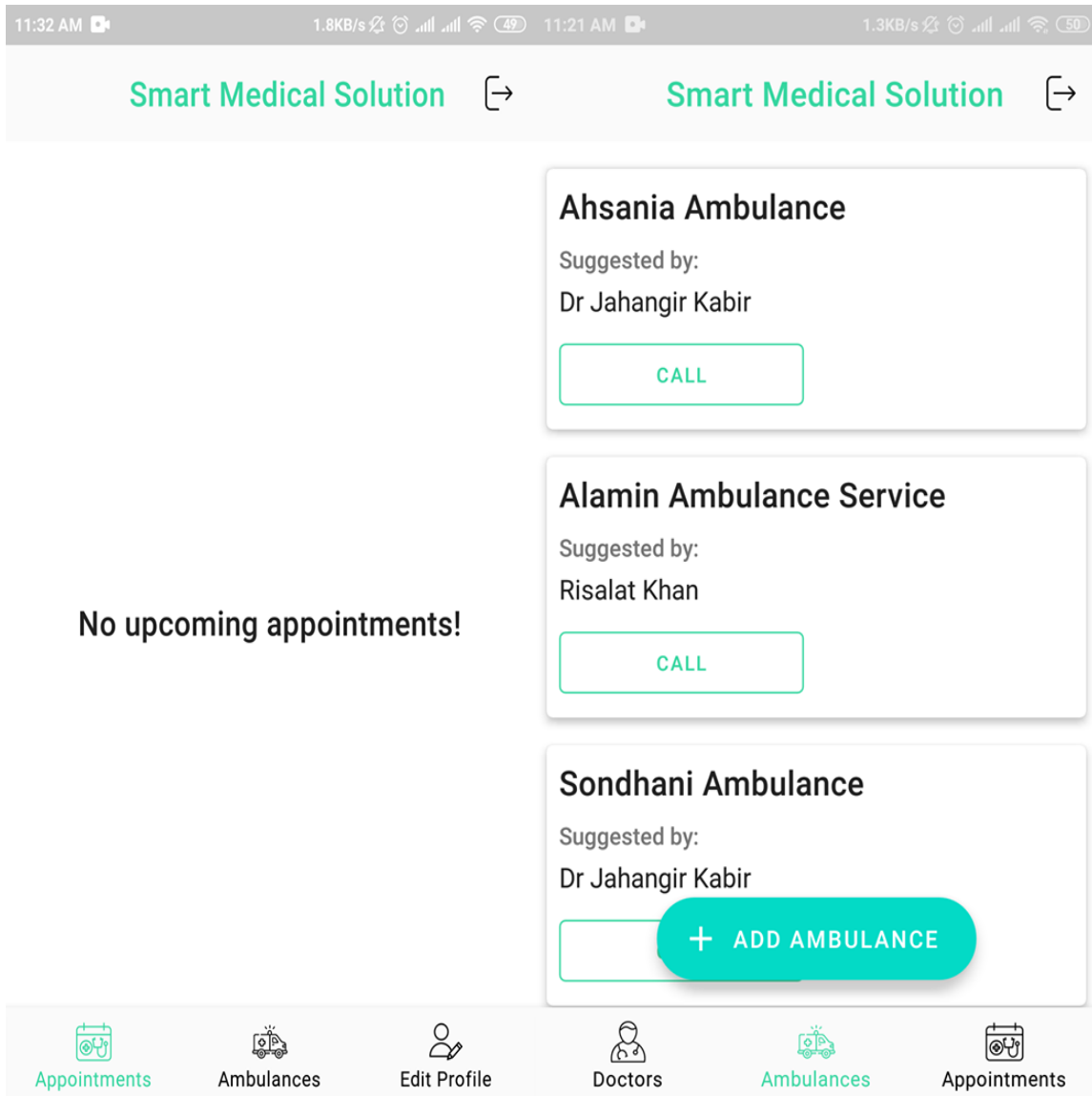


Fig 3.8.6: Doctors login

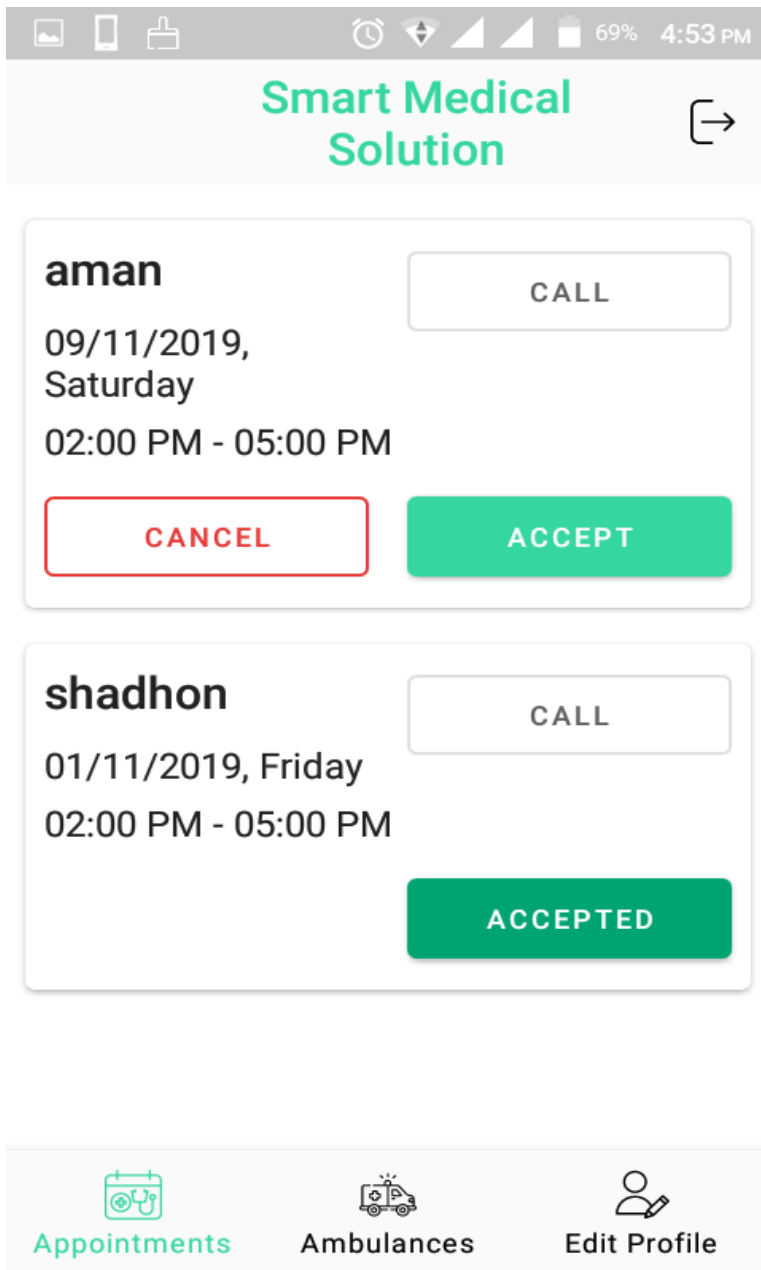


Fig 3.8.7: Doctors login

CHAPTER 04

DESIGN SPECIFICATION

A definition of design is an explanation for the development of design. In the "Design Requirements" section, we tried to look at the design of our mobile application. I have also discussed many of the tools I use to develop my application.

4.1 Implementation Requirement

I use different types of implementation components, features, and tools to create an application. These tools make the application easy to use and useful to the application. Let me know the basic code, Java and XML, for my project. The functionality of my application is written in English.

To execute our project, you must:

4.2 Android Studio

Android Studio is an Integrated Development Environment (IDE) for upgrading to the Android operating system. Specifically designed for Android application development. I developed my application in Android Studio.

4.3 Android SDK

The Android SDK provides API libraries and development tools for testing, creating, and debugging an Android application. Developing Android applications requires the Android SDK. This SDK is a powerful tool that includes not only a development library but also a simulator for running a test application.

4.4 Java Development kit (JDK)

JDK is an implementation of either one of the java SE, Java EE or Java ME platforms.

4.5 Android virtual Device (AVD)

The creator can use the Android emulator as a target platform to run and test his Android application on his computer. The emulator in AVD Manager provides a graphical interface that uses the Android application. An emulation tool used to debug and test applications. An emulation tool used to debug and test applications. You need to install an Android system image that your emulator can use before creating an Android emulator.

4.6 Interaction Design and UX

Interaction design is the subject that facilitates the interaction of this process. In fact, contact design is a system that understands a user's problem. I find my app very easy, but also easy to improve my experience and productivity. I tried to give my UX application a great experience. I find my system simple to improve user performance and experience.

4.7 CODE and DATABASE :

The figure consists of two screenshots of the Firebase Database console. The top screenshot shows the 'ambulances' collection with a document containing the following data:

```

name: "Ahsania Ambulance"
phone: "+88097776666"
  
```

The bottom screenshot shows the 'appointments' collection with a document containing the following data:

```

doctor: /users/fhy0sP0dVwVTsWGLzr4XXkbTCIO2
end: October 23, 2019 at 12:00:02 PM UTC+6
patient: /users/hMNrgEhRjefD8UOvceyb6RUmg82
start: October 23, 2019 at 7:00:02 AM UTC+6
status: 1
  
```

Figure 4.7.1 : Ambulance & Appointment

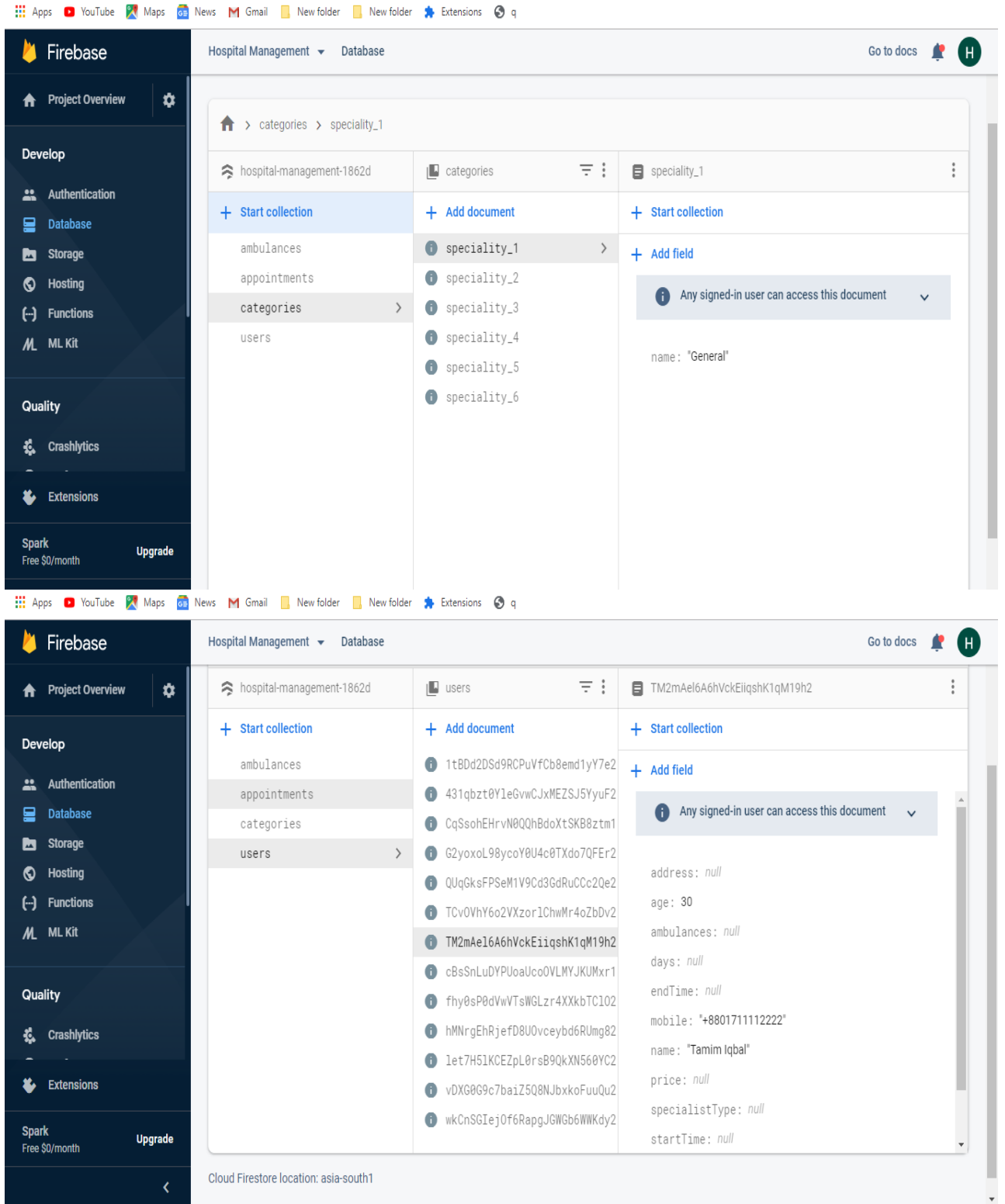


Figure 4.7.2: Categories & Users

Java Codes:

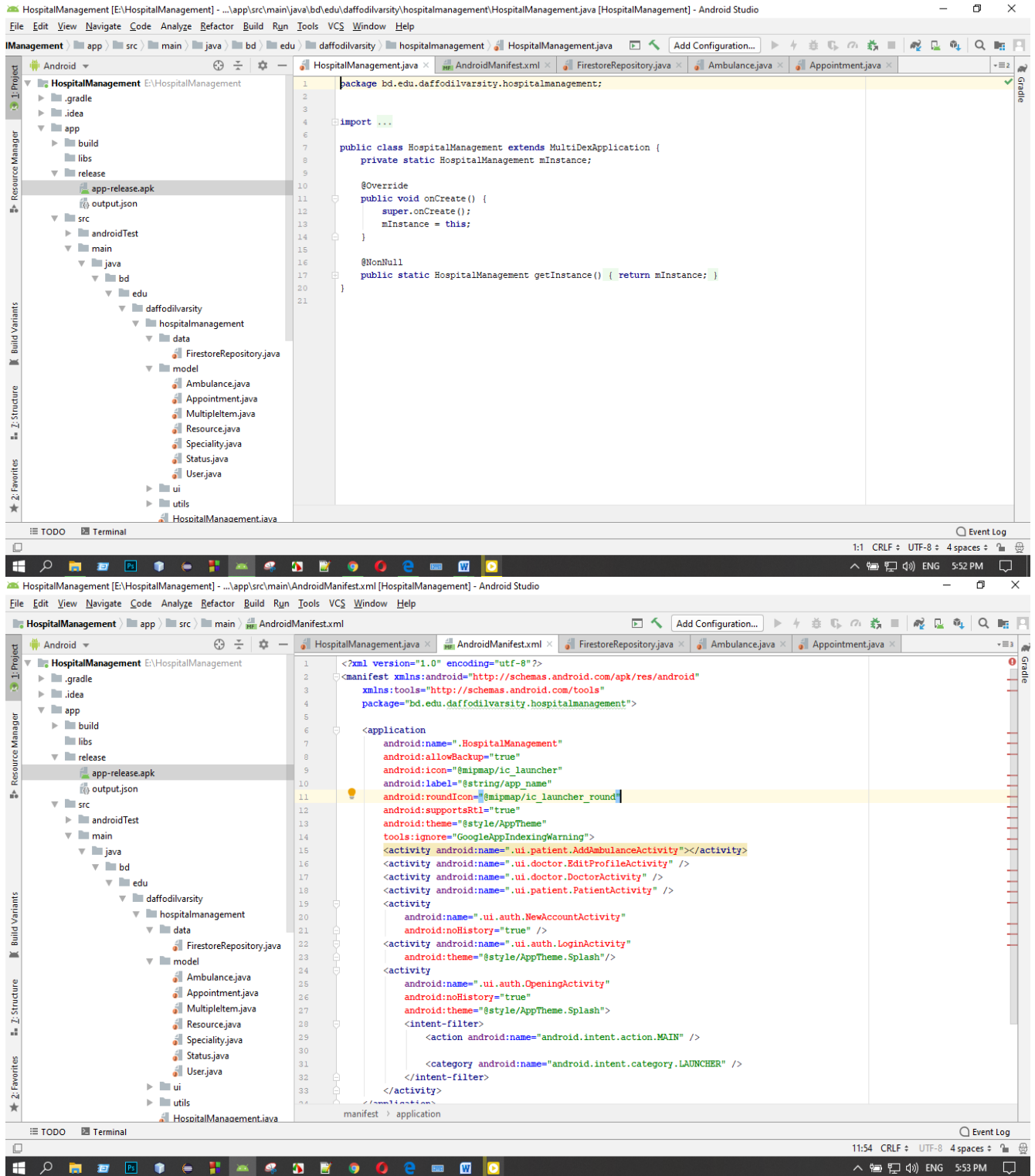


Figure 4.7.3 :Android XML file & Android .Java file

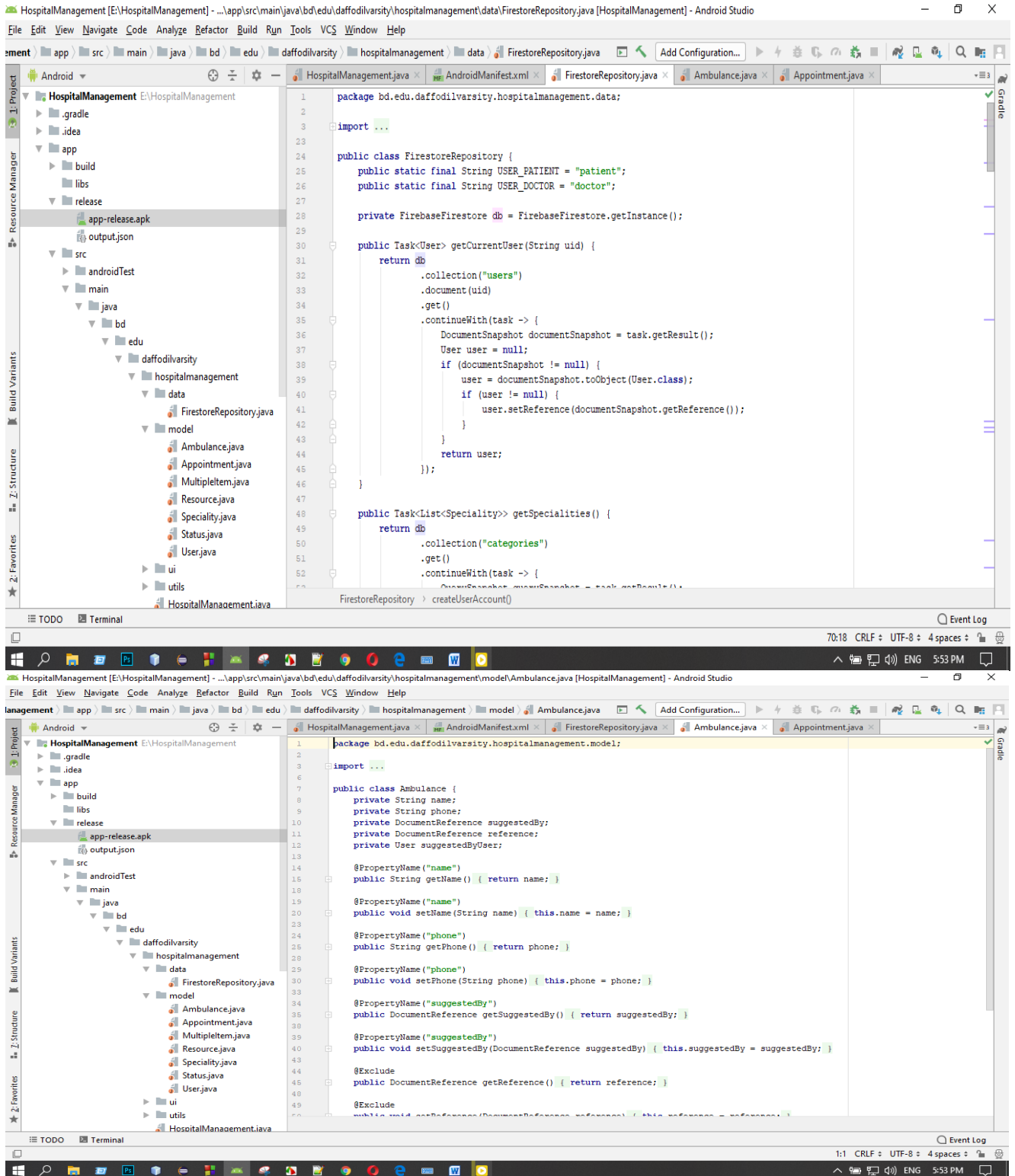


Figure 4.7.4 : FireStoreRepository & Ambulance.Java

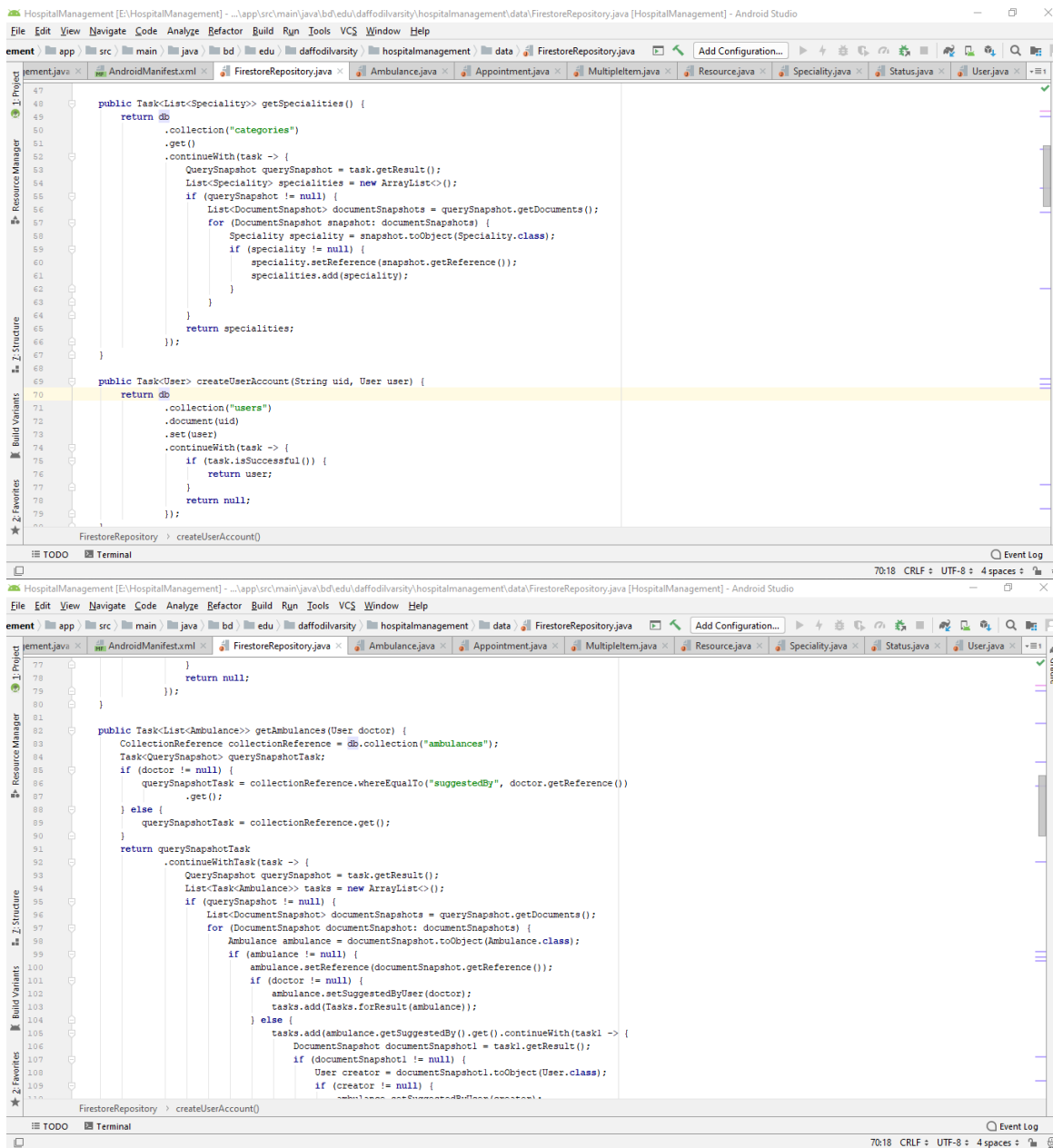


Figure 4.7.5 : FirestoreRepository & Ambulance.java

```

113         return ambulance;
114     }
115     }
116     }
117     }
118     }
119     }
120     }
121     }
122     }
123     public Task<List<User>> getDoctorsList() {
124         return db
125             .collection("users")
126             .whereEqualTo("userType", USER_DOCTOR)
127             .get()
128             .continueWithTask(task -> {
129                 QuerySnapshot querySnapshot = task.getResult();
130                 List<Task<User>> userTasks = new ArrayList<>();
131                 if (querySnapshot != null) {
132                     List<DocumentSnapshot> documentSnapshots = querySnapshot.getDocuments();
133                     for (DocumentSnapshot documentSnapshot : documentSnapshots) {
134                         User doctor = documentSnapshot.toObject(User.class);
135                         if (doctor != null) {
136                             doctor.setReference(documentSnapshot.getReference());
137                             userTasks.add(doctor.getSpecialistType());
138                         }
139                     }
140                     .continueWith(task1 -> {
141                         DocumentSnapshot documentSnapshot1 = task1.getResult();
142                         if (documentSnapshot1 != null) {
143                             Speciality speciality = documentSnapshot1.toObject(Speciality.class);
144                             if (speciality != null) {
145                                 doctor.setSpecialist(speciality.getName());
146                             }
147                         }
148                     });
149                 }
150             });
151     }
152     }
153     }
154     }
155     }
156     }
157     }
158     }
159     }
160     }
161     }
162     }
163     }
164     }
165     }
166     }
167     }
168     }
169     }
170     }
171     }
172     public Task<List<Appointment>> getAppointments(String uid, @NonNull String userType) {
173         CollectionReference collectionReference = db.collection("appointments");
174         Query query;
175         if (userType.equals(USER_PATIENT)) {
176             query = collectionReference.whereEqualTo("patient", db.collection("users").document(uid));
177         } else {
178             query = collectionReference.whereEqualTo("doctor", db.collection("users").document(uid));
179         }
180         return query
181             .get()
182             .continueWith(task -> {
183                 QuerySnapshot querySnapshot = task.getResult();
184                 List<Task<Appointment>> appointmentTasks = new ArrayList<>();
185                 if (querySnapshot != null) {
186                     List<DocumentSnapshot> documentSnapshots = querySnapshot.getDocuments();
187                     for (DocumentSnapshot documentSnapshot : documentSnapshots) {
188                         Appointment appointment = documentSnapshot.toObject(Appointment.class);
189                         if (appointment != null) {
190                             appointment.setReference(documentSnapshot.getReference());
191                             appointmentTasks.add(appointment);
192                         }
193                     }
194                     .continueWith(task1 -> {
195                         DocumentSnapshot documentSnapshot1 = task1.getResult();
196                         if (documentSnapshot1 != null) {
197                             User doctor1 = documentSnapshot1.toObject(User.class);
198                             if (doctor1 != null) {
199                                 List<Ambulance> ambulances = task121.getResult();
200                                 if (ambulances != null && ambulances.size() > 0) {
201                                     doctor1.setAmbulances(ambulances);
202                                 }
203                             }
204                         }
205                     });
206                 }
207             });
208         return Tasks.forResult(null);
209     }
210     }
211     }
212     }
213     }
214     }
215     }
216     }
217     }
218     }
219     }
220     }
221     }
222     }
223     }
224     }
225     }
226     }
227     }
228     }
229     }
230     }
231     }
232     }
233     }
234     }
235     }
236     }
237     }
238     }
239     }
240     }
241     }
242     }
243     }
244     }
245     }
246     }
247     }
248     }
249     }
250     }
251     }
252     }
253     }
254     }
255     }
256     }
257     }
258     }
259     }
260     }
261     }
262     }
263     }
264     }
265     }
266     }
267     }
268     }
269     }
270     }
271     }
272     }
273     }
274     }
275     }
276     }
277     }
278     }
279     }
280     }
281     }
282     }
283     }
284     }
285     }
286     }
287     }
288     }
289     }
290     }
291     }
292     }
293     }
294     }
295     }
296     }
297     }
298     }
299     }
300     }
301     }
302     }
303     }
304     }
305     }
306     }
307     }
308     }
309     }
310     }
311     }
312     }
313     }
314     }
315     }
316     }
317     }
318     }
319     }
320     }
321     }
322     }
323     }
324     }
325     }
326     }
327     }
328     }
329     }
330     }
331     }
332     }
333     }
334     }
335     }
336     }
337     }
338     }
339     }
340     }
341     }
342     }
343     }
344     }
345     }
346     }
347     }
348     }
349     }
350     }
351     }
352     }
353     }
354     }
355     }
356     }
357     }
358     }
359     }
360     }
361     }
362     }
363     }
364     }
365     }
366     }
367     }
368     }
369     }
370     }
371     }
372     }
373     }
374     }
375     }
376     }
377     }
378     }
379     }
380     }
381     }
382     }
383     }
384     }
385     }
386     }
387     }
388     }
389     }
390     }
391     }
392     }
393     }
394     }
395     }
396     }
397     }
398     }
399     }
400     }
401     }
402     }
403     }
404     }
405     }
406     }
407     }
408     }
409     }
410     }
411     }
412     }
413     }
414     }
415     }
416     }
417     }
418     }
419     }
420     }
421     }
422     }
423     }
424     }
425     }
426     }
427     }
428     }
429     }
430     }
431     }
432     }
433     }
434     }
435     }
436     }
437     }
438     }
439     }
440     }
441     }
442     }
443     }
444     }
445     }
446     }
447     }
448     }
449     }
450     }
451     }
452     }
453     }
454     }
455     }
456     }
457     }
458     }
459     }
460     }
461     }
462     }
463     }
464     }
465     }
466     }
467     }
468     }
469     }
470     }
471     }
472     }
473     }
474     }
475     }
476     }
477     }
478     }
479     }
480     }
481     }
482     }
483     }
484     }
485     }
486     }
487     }
488     }
489     }
490     }
491     }
492     }
493     }
494     }
495     }
496     }
497     }
498     }
499     }
500     }
501     }
502     }
503     }
504     }
505     }
506     }
507     }
508     }
509     }
510     }
511     }
512     }
513     }
514     }
515     }
516     }
517     }
518     }
519     }
520     }
521     }
522     }
523     }
524     }
525     }
526     }
527     }
528     }
529     }
530     }
531     }
532     }
533     }
534     }
535     }
536     }
537     }
538     }
539     }
540     }
541     }
542     }
543     }
544     }
545     }
546     }
547     }
548     }
549     }
550     }
551     }
552     }
553     }
554     }
555     }
556     }
557     }
558     }
559     }
560     }
561     }
562     }
563     }
564     }
565     }
566     }
567     }
568     }
569     }
570     }
571     }
572     }
573     }
574     }
575     }
576     }
577     }
578     }
579     }
580     }
581     }
582     }
583     }
584     }
585     }
586     }
587     }
588     }
589     }
590     }
591     }
592     }
593     }
594     }
595     }
596     }
597     }
598     }
599     }
600     }
601     }
602     }
603     }
604     }
605     }
606     }
607     }
608     }
609     }
610     }
611     }
612     }
613     }
614     }
615     }
616     }
617     }
618     }
619     }
620     }
621     }
622     }
623     }
624     }
625     }
626     }
627     }
628     }
629     }
630     }
631     }
632     }
633     }
634     }
635     }
636     }
637     }
638     }
639     }
640     }
641     }
642     }
643     }
644     }
645     }
646     }
647     }
648     }
649     }
650     }
651     }
652     }
653     }
654     }
655     }
656     }
657     }
658     }
659     }
660     }
661     }
662     }
663     }
664     }
665     }
666     }
667     }
668     }
669     }
670     }
671     }
672     }
673     }
674     }
675     }
676     }
677     }
678     }
679     }
680     }
681     }
682     }
683     }
684     }
685     }
686     }
687     }
688     }
689     }
690     }
691     }
692     }
693     }
694     }
695     }
696     }
697     }
698     }
699     }
700     }
701     }
702     }
703     }
704     }
705     }
706     }
707     }
708     }
709     }
710     }
711     }
712     }
713     }
714     }
715     }
716     }
717     }
718     }
719     }
720     }
721     }
722     }
723     }
724     }
725     }
726     }
727     }
728     }
729     }
730     }
731     }
732     }
733     }
734     }
735     }
736     }
737     }
738     }
739     }
740     }
741     }
742     }
743     }
744     }
745     }
746     }
747     }
748     }
749     }
750     }
751     }
752     }
753     }
754     }
755     }
756     }
757     }
758     }
759     }
760     }
761     }
762     }
763     }
764     }
765     }
766     }
767     }
768     }
769     }
770     }
771     }
772     }
773     }
774     }
775     }
776     }
777     }
778     }
779     }
780     }
781     }
782     }
783     }
784     }
785     }
786     }
787     }
788     }
789     }
790     }
791     }
792     }
793     }
794     }
795     }
796     }
797     }
798     }
799     }
800     }
801     }
802     }
803     }
804     }
805     }
806     }
807     }
808     }
809     }
810     }
811     }
812     }
813     }
814     }
815     }
816     }
817     }
818     }
819     }
820     }
821     }
822     }
823     }
824     }
825     }
826     }
827     }
828     }
829     }
830     }
831     }
832     }
833     }
834     }
835     }
836     }
837     }
838     }
839     }
840     }
841     }
842     }
843     }
844     }
845     }
846     }
847     }
848     }
849     }
850     }
851     }
852     }
853     }
854     }
855     }
856     }
857     }
858     }
859     }
860     }
861     }
862     }
863     }
864     }
865     }
866     }
867     }
868     }
869     }
870     }
871     }
872     }
873     }
874     }
875     }
876     }
877     }
878     }
879     }
880     }
881     }
882     }
883     }
884     }
885     }
886     }
887     }
888     }
889     }
890     }
891     }
892     }
893     }
894     }
895     }
896     }
897     }
898     }
899     }
900     }
901     }
902     }
903     }
904     }
905     }
906     }
907     }
908     }
909     }
910     }
911     }
912     }
913     }
914     }
915     }
916     }
917     }
918     }
919     }
920     }
921     }
922     }
923     }
924     }
925     }
926     }
927     }
928     }
929     }
930     }
931     }
932     }
933     }
934     }
935     }
936     }
937     }
938     }
939     }
940     }
941     }
942     }
943     }
944     }
945     }
946     }
947     }
948     }
949     }
950     }
951     }
952     }
953     }
954     }
955     }
956     }
957     }
958     }
959     }
960     }
961     }
962     }
963     }
964     }
965     }
966     }
967     }
968     }
969     }
970     }
971     }
972     }
973     }
974     }
975     }
976     }
977     }
978     }
979     }
980     }
981     }
982     }
983     }
984     }
985     }
986     }
987     }
988     }
989     }
990     }
991     }
992     }
993     }
994     }
995     }
996     }
997     }
998     }
999     }
1000    }

```

Fig: 4.7.6 FirestoreRepository & Ambulance.java

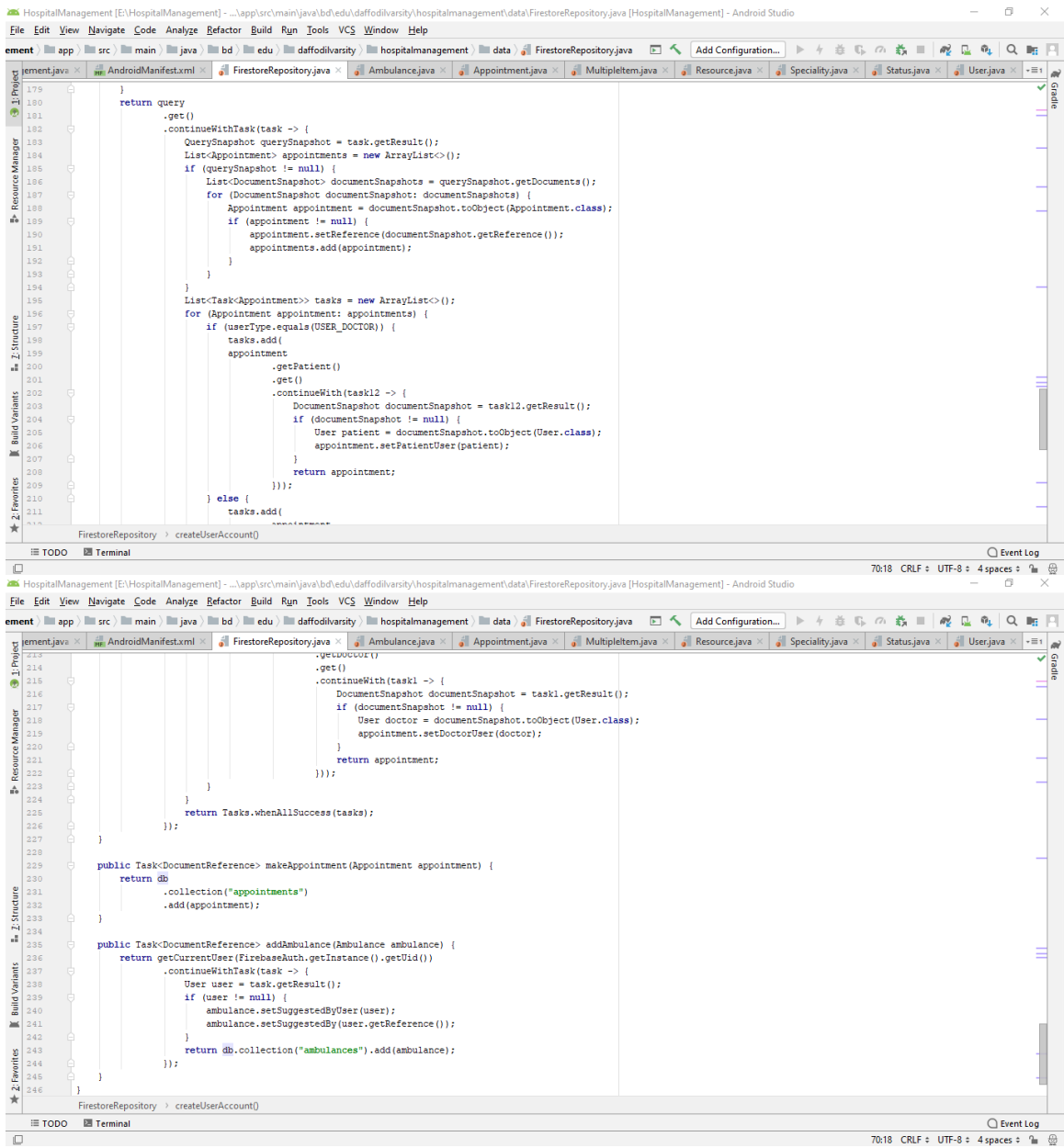


Figure 4.7.7 : FirestoreRepository & Ambulance.java

```

package bd.edu.daffodilvarsity.hospitalmanagement.model;

import ...

public class Appointment {
    @Exclude
    public static final int ACCEPTED = 1;
    @Exclude
    public static final int CANCELLED = 0;
    @Exclude
    public static final int REQUESTED = -1;

    private int status;
    private DocumentReference patient;
    private DocumentReference doctor;
    private Timestamp start;
    private Timestamp end;
    private DocumentReference reference;
    private User patientUser;
    private User doctorUser;

    @PropertyName("status")
    public int getStatus() { return status; }

    @PropertyName("status")
    public void setStatus(int status) { this.status = status; }

    @PropertyName("patient")
    public DocumentReference getPatient() { return patient; }

    @PropertyName("patient")
    public void setPatient(DocumentReference patient) { this.patient = patient; }
}

```

```

package bd.edu.daffodilvarsity.hospitalmanagement.model;

import ...

public class MultipleItem extends AbstractExpandableItem<MultipleItem> implements MultiItemEntity {
    public static final int DOCTOR_LIST = 1030;
    public static final int AMBULANCE_LIST = 1031;
    public static final int AMBULANCE_EXPANDED_LIST = 1032;
    public static final int APPOINTMENT_LIST = 1033;

    private int itemType;
    private User user;
    private Ambulance ambulance;
    private Appointment appointment;

    public MultipleItem(User user) {
        this.itemType = DOCTOR_LIST;
        this.user = user;
    }

    public MultipleItem(int itemType, Ambulance ambulance) {
        this.itemType = itemType;
        this.ambulance = ambulance;
    }

    public MultipleItem(Appointment appointment) {
        this.itemType = APPOINTMENT_LIST;
        this.appointment = appointment;
    }

    @Override
    public int getItemType() { return itemType; }
}

```

Figure4.7.8 : Appoinment & Figure MultipleItem

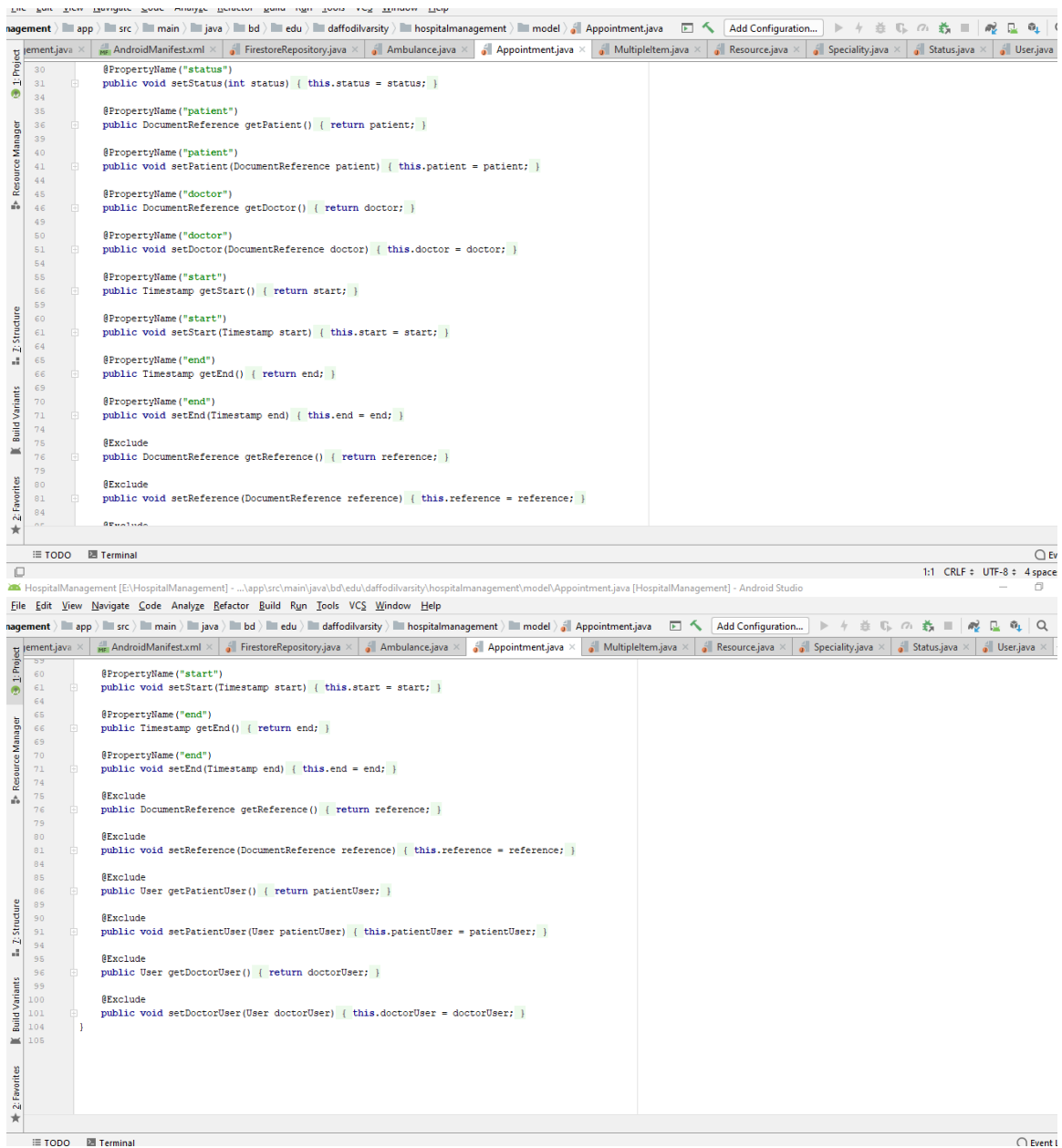


Figure 4.7.9 : Appoinment & Figure MultipleItem

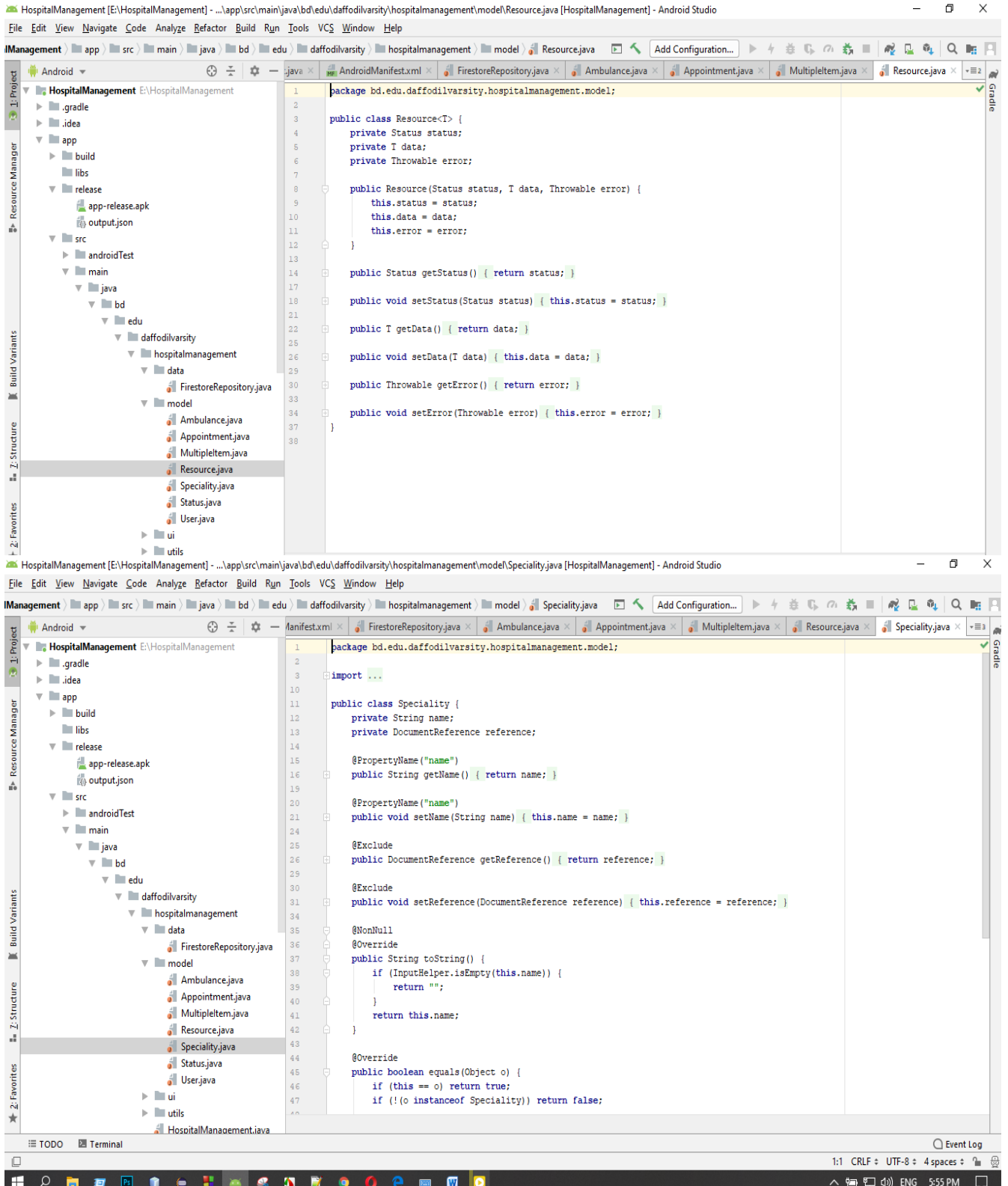


Figure 4.7.10: Resources & Specialty

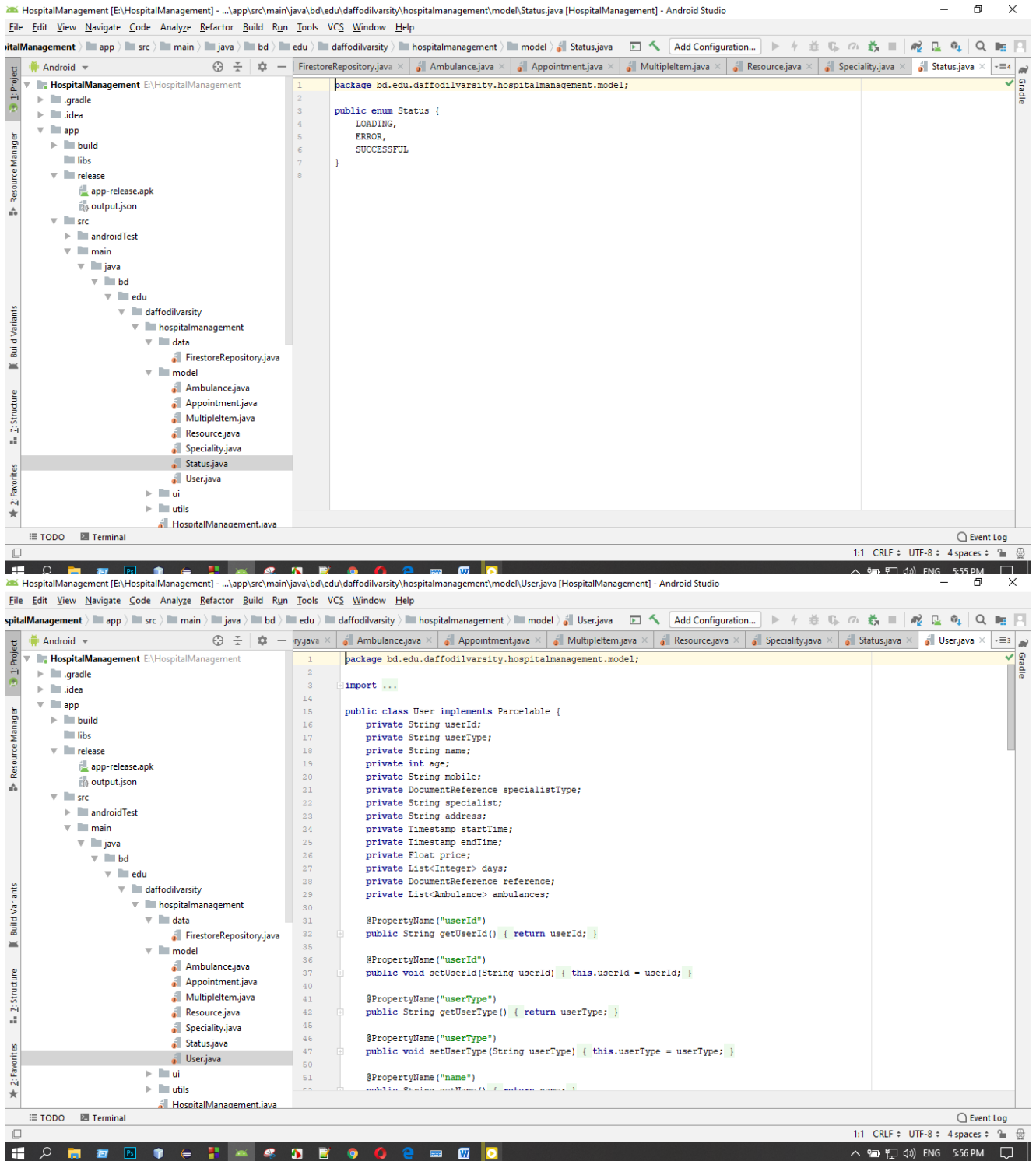


Figure : 4.7.11 Status & User.java

CHAPTER 05

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

“Smart Medical Solution” is an application for helping people make connected with doctors. In our application we use firebase for saving data in database. In our application user information like Phone number, name and id will be saved. Here include the all types of data which will be the record of data which will record of our background system also.

5.2 Testing

Testing is an integral part of any application or software. Create an application and the developer can go wrong. After the repair, the developer may find the error. The BD Hospital app works well on Android-powered Android devices version is equal or greater than 5.1. The interface is dynamic so it creates no error with several sizes of android devices all the features like search hospital, find location, find phone number, find hospital, find doctor’s information work good in almost every android device.

5.3 Implementation of Java and XML code

Use an Android studio using my UL File Designer application and Java kernel to connect an XML file to a Firebase database to store user data. I connect apps to the Internet and design Android Studio. Android Studio supports many different languages. Use object, class, and some Java code methods. This means that the work is done in Java code. In XML, I use a text editor in my application, a button. I developed the application with XML coding.

5.4 Testing Implementation

The Test Case for Project on Smart medical Solution.

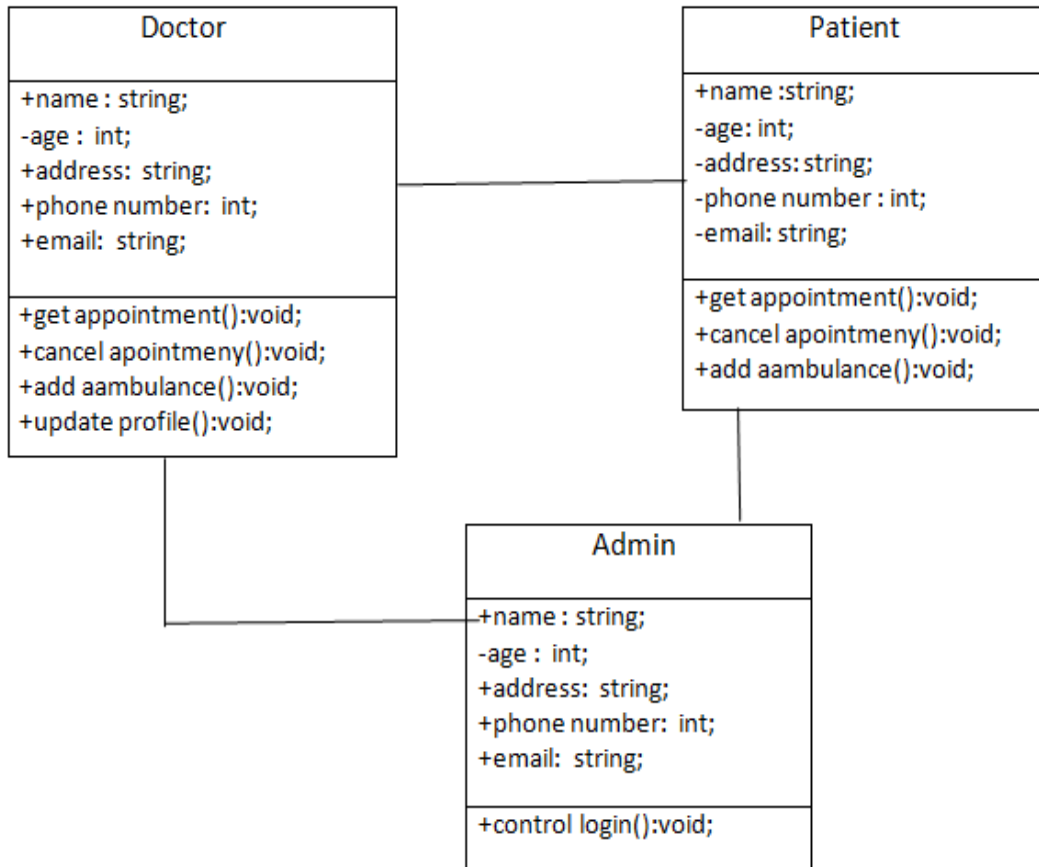


Fig 5.4: ER Diagram.

CHAPTER 06

CONCLUSION AND FUTURE SCOPE

Conclusion

Smart Medical Solution code has been successfully build and tested. We used this app in other phones and its run successfully. This is a smart application for this modern and smart world.

Future work

In future we will add payment option. By this option doctors will receive money in a certain time. We will also provide all the hospital to use this application. Then medical treatment will be easy way to anyone. They are:-

1. Payment Option.
2. Google Map Location.
3. Google Talking.

References

- [1] Learn about Use Case and java problems at << <https://developer.android.com/training/index.html> >>, Last accessed on 03-11-2019 at 09:00am.
- [2] Learn about Project Ideas, Available at << <https://nevonprojects.com/project-ideas/electronics-ideas/> >>, Last accessed on 03-10-2019 at 11:00am.
- [3] Learn about Android tutorials at << [https:// developer.android.com/training/index.html](https://developer.android.com/training/index.html) >>, last accessed on 03-10-2019 at 12:10pm.
- [4] Learn about Android and java problems at << <https://developer.android.com/training/index.html> >>, Last accessed on 03-11-2019 at 09:00am.
- [5] Learn about JAVASCRIPT, Available at << [https://developer.mozilla.org/en-US/docs/Web/JavaScript />](https://developer.mozilla.org/en-US/docs/Web/JavaScript/)>, Last accessed on 03-11-2019 at 10:00am.
- [6] Learn about Medical related application with android, Available at << <https://developer.android.com/training/index.html> >>, Last accessed on 03-12-2019 at 9:00pm.
- [7] Learn about Interaction Design, Available at << https://www.interaction-design.org/literature/topics/interaction-design_>>, last accessed on 03-16-2019 at 10:00am.
- [8] Learn about Database Server and Administration Tools, Available at <<<https://firebase.google.com/> >>, last accessed on 03-17-2019 at 02:00pm.
- [9] Learn about Database, Available at << <https://firebase.google.com/> >>, Last accessed on 03-19-2019 at 10:00pm.