

# **ANDROID-BASED APPLICATION FOR EVENT MANAGEMENT**

**BY**

**Md. Kamrul Hasan**  
**ID: 151-15-5498**

**Md. Shahadat Hossain**  
**ID: 151-15-4774**

**AND**

**Syed Md Samsudduha**  
**ID: 151-15-4773**

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

Supervised By

**Moushumi Zaman Bonny**  
Senior Lecturer  
Department of CSE  
Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**NOVEMBER 2018**

## **APPROVAL**

This Project titled “**Android Based Application for Event Management**”, submitted by Md. Kamrul Hasan, ID No: 151-15-5498, Syed Md Samsudduha, ID No: 151-15-4773 and Md Shahadat Hossain, ID No: 151-15-4774 to 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 10-12-2018

### **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**

---

**Narayan Ranjan Chakraborty**

**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

---

**Md. Tarek Habib**

**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

---

**Dr. Mohammad Shorif Uddin**

**Professor**

Department of Computer Science and Engineering  
Jahangirnagar University

**External Examiner**

## DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Moushumi Zaman Bonny, Senior 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:

---

**Moushumi Zaman Bonny**  
Senior Lecturer  
Department of CSE  
Daffodil International University

### Submitted by:

---

**Md Kamrul Hasan**  
ID: -151-15-5498  
Department of CSE  
Daffodil International University

---

**Syed Md Samsudduha**  
ID: -151-15-4773  
Department of CSE  
Daffodil International University

---

**Md Shahadat Hossain**  
ID: -151-15-4774  
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 fell grateful and wish our profound our indebtedness to **Moushumi Zaman Bonny**, **Senior Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of Android Application influenced us to carry out this project. Her 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, Professor and Head, Department of CSE, Daffodil International University 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 patience of our parents.

## **ABSTRACT**

Nowadays, Smart phone is a very essential for our every day life. It is not only communication tool but also it's used in various purpose like education, entertainment etc. with Android System Smart phone popular day by day. Android is open source thats why some of development tools free to use. So that it's easy to develop any kind of Android application.

In this project we have developed and android base mobile application for user who want to book their event in online without any hassle. Using this application user can customize their own event also can booked their individual Artist and can booked any types of event like as wedding, birthday, corporate, concert etc. There are some unique features in our application like career opportunity. By using this feature anyone can easily build their career like as photographer, singer, dancer etc. Anyone can easily contact with us by using our call us option. Anyone can checked their ordered history. Finally it can be said that this application is very helpful for those, who want to celebrate his event.

# TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
<b>CHAPTER</b>	
<b>CHAPTER 1: INTRODUCTION</b>	<b>01-02</b>
1.1 Introduction	01
1.2 Motivation	01
1.3 Objectives	02
1.4 Expected Outcome	02
1.5 Report Layout	02
<b>CHAPTER 2: BACKGROUND</b>	<b>03-04</b>
2.1 Introduction	03
2.2 Related Works	03
2.3 Comparative Studies	03
2.4 Scope of the Problem	04
2.5 Challenges	04
<b>CHAPTER 3: REQUIREMENT SPECIFICATION</b>	<b>05-08</b>
3.1 Business Process Modeling	05
3.2 Requirement Collection & Analysis	06
3.3 Use Case Modeling & Description	06
3.4 Logical Data Model	07
3.5 Design Requirements	08

<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>09-26</b>
4.1 Front-end Design	09
4.2 Back-end Design	20
4.3 Interaction Design & UX	25
4.4 Implementation Requirements	26
<b>CHAPTER 5: IMPLEMENTATION AND TESTING</b>	<b>27-32</b>
5.1 Implementation of Database	27
5.2 Implementation of Front-end Design	28
5.3 Implementation of Interactions	30
5.4 Testing Implementation	30
5.5 Test Results & Reports	31
<b>CHAPTER 6: CONCLUSION AND FUTURE SCOPE</b>	
6.1 Discussion & Conclusion	33
6.2 Scope for Future Developments	33
<b>REFERENCES</b>	<b>34-35</b>
<b>APPENDIX</b>	<b>36-40</b>

## List of figures

<b>Figure</b>	<b>Page</b>
Figure 3.1: A general block diagram	5
Figure 3.2: Admin use case	6
Figure 3.3: User use case diagram	7
Figure 3.4: Logical data model diagram	7
Figure 4.1: Login & Registration screen	9
Figure 4.2: Home & navigation screen	10
Figure 4.3: Order history & About screen	11
Figure 4.4: Why us & Wedding screen	12
Figure 4.5: Royal wedding & Stage booking system	13
Figure 4.6: Birthday & Corporate screen	14
Figure 4.7: Conference & product booking system	15
Figure 4.8: Concert system screen	16
Figure 4.9: Book artist system	17
Figure 4.10: Customize and career system	18
Figure 4.11: Map and booking confirmation	19
Figure 4.12: Firebase home page	20
Figure 4.13: Booking system database	21
Figure 4.14: Band list database	22
Figure 4.15: Fair type database	23
Figure 4.16: Location database	23
Figure 4.17: User information database	24
Figure 4.18: Authentication database	25



## List of tables

<b>Figure</b>	<b>Page</b>
Figure 5.5.1: Test case for login	31
Figure 5.5.2: Test case for check home page	32
Figure 5.5.3: Test case for check booking system	32

# CHAPTER 1

## Introduction

### 1.1 Introduction

Each time when we saw a program like 'Ittyadi' rewards or those pop concerts and wedding we would have speculated how people achieved to arranged like this brilliant shows. Every minor detail from our entering the venue to our going out is a result of proper development. That's event management for us.

“Android Based Event Management Application” is an android mobile application which is online booking based. The main goal us to use this application, to book any kinds of event from anywhere. Customer can easily check their order information in future.

### 1.2 Motivation

Event management system is used to control all the process related to event. In one event so many service providers work together and it is very tough to control these providers. It is also essential for event organizer that he has all the contacts details of these service providers so that he can contact them any time to plan an event at given time.

We analysis many event management applications but we only found web application where proper make up their event theme. But we can't found any android base application. For this reason, we motivated. Then we create our Event management application. we have industrialized this application on android based. This system will let people know about the cost of different events and their services with this application user book their required event like music concert, wedding event, birthday event etc. This system is more user friendly than existing systems. So, involved people can easily access the system.

### **1.3 Objectives**

The main objective of the project on event management system is to manage the details of event, booking, customer, employee. Customer can easily book their required event from category. It manages all the material about event, package, cost etc. It is an application to decrease the manual work for managing the event & save their respected time. The user gets all the properties at a single place.

### **1.4 Expected Outcome**

The outcome of an activity, process, or situation is the situation that exists at the end of it [1], [2]. Our project will change the thinking of manual work of the event management. We can expect from our project people are using this application, they can easily book there any types of event. It's not a just booking platform user can get job opportunity by using career feature.

### **1.5 Report Layout**

We have organized this report as follows: In the chapter 2, we discuss about background of our project. We also briefed about the related work comparative studies scope of problem and challenges. In the chapter 3, we declared the requirement specification of the project. In this section we defined the business process modelling, requirement collection and analysis, use case modelling and description logical data model, design requirement. In business process modelling we defined the data flow diagram, use-case. In the chapter 4, we specified the front-end and the back-end design. In the chapter 5, we implementation of the whole project and testing every section of the project. In the chapter 6, we discuss about the conclusion and the scope for future of the development project. At last, we declared all of the related reference to related our report information.

## **CHAPTER 2**

### **Background**

#### **2.1 Introduction**

In recent days, most of us spend lots of time on Internet. We are more comfortable in online shopping or online Activity. That's why we thought that to develop something new that can be helpful for people and it could make their event very easier. It could help easier decision for how event occur. When we start to design the platform, we always keep some fact in our knowledge that the design should be only for who went to occur the event. It supports all types of information in event and they can also book there event any type of harass [3]. It's very easier platform for occur event. They can easily book event, artist and also be develop the own career. That type of application is not present in time. That's why we inspired and we can develop this application.

#### **2.2 Related Works**

This is very difficult for one person to manage their own event activities. It could be said that there is no suitable android application. Recently users, use many websites for book their event but they could not satisfy with web interface. In the present time android smart phone user increasing day by day [4]. Most of the smart phone screen size between 5 to 7 inch that's why web page not properly displayed in this smart phone screen. Therefore, we believe that "Android Based Event Management" Application helps for people properly booking their event without any hassle.

#### **2.3 Comparative Studies**

The comparative method is often used in the early stages of the development of a branch of science. It can help the researcher to ascent from the initial level of exploratory case studies to a more advance level of general theoretical model [5]. Most of the people are arrange their event by using event management. "BD Showbiz" event management is one the popular event management site. But this

company is not arranging all types of event and also their feature also limited. Those features are not enough to fulfil user requirement.

Without “BD Showbiz” there is another platform for user, “watermelon” that only organized the wedding and birthday event [6]. They are also some limitation in their site.

So that we are interested to developed “Event Management” android based project. We think that our unique function can make user friendly and fulfil user requirement successfully.

## **2.4 Scope of the Problem**

Event management is for booking an event through android based applications. Its main programming language for front-end will be java and for back-end will be json. We have focused here to simplify and increase the efficiency of the booking system for users. Our target is to minimize manual data entry and ensure data accuracy also the security during data process. We have kept options for customers to view the event with the essential report. Users will also be able to view a visual confirmation that the booked was place successfully.

## **2.5 Challenges**

There is a common thing to perform an activity is challenges. There is no work without challenge similarly, in our project face some challenges. First of all, we trying to create our event management application in all stage of people. Because of some particular area among rural areas people are very conservative. They are arranging their event any kind of help without event management. That’s why they fail to arrange their event properly and lose their money. So, user should be informed the consequence of our Event Management as well as tell them the benefits of our application.

## CHAPTER 3

### Requirement Specification

#### 3.1 Business Process Modeling

Business process modeling is mapping out regular business processes and finding ways to improve them. It is a part of the practice of business process management. Process modeling software gives an analytical representation of 'as-is' processes in an organization and contrasts it with 'to-be' processes for making them more efficient [7], [8].

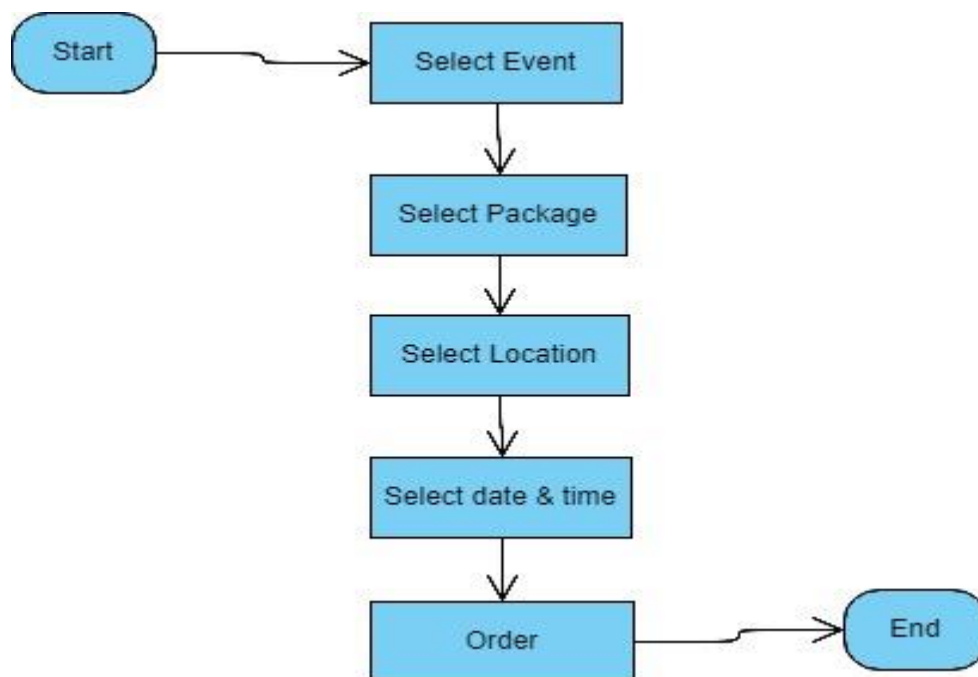


Figure 3.1: A general block diagram

### 3.2 Requirement Collection and Analysis

Software requirement is a functional or non-functional need to be implemented in the system. Functional means providing particular service to the user.

For example, in context to banking application the functional requirement will be when customer selects "View Balance" they must be able to look at their latest account balance. Software requirement can also be a non-functional, it can be a performance requirement. For example, a non-functional requirement is where every page of the system should be visible to the users within 5 seconds [9].

So, basically software requirement is two types, functional and non-functional.

### 3.3 Uses-Case modeling & Description

A use case diagram in the Unified Modeling Language (UML) is a type of interactive diagram defined by and created from a Use-case analysis. Its resolution is to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependences between those use cases [10], [11]. The main purpose of a use case diagram is to show what system functions are performed for which actor.

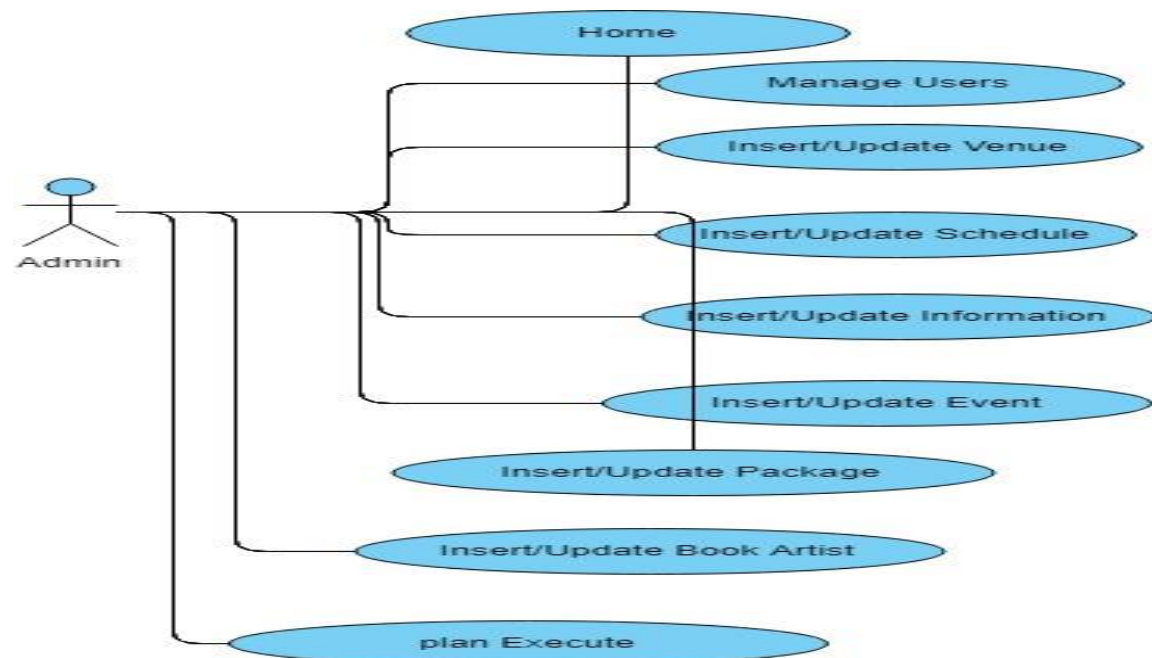


Figure 3.2: Admin use case

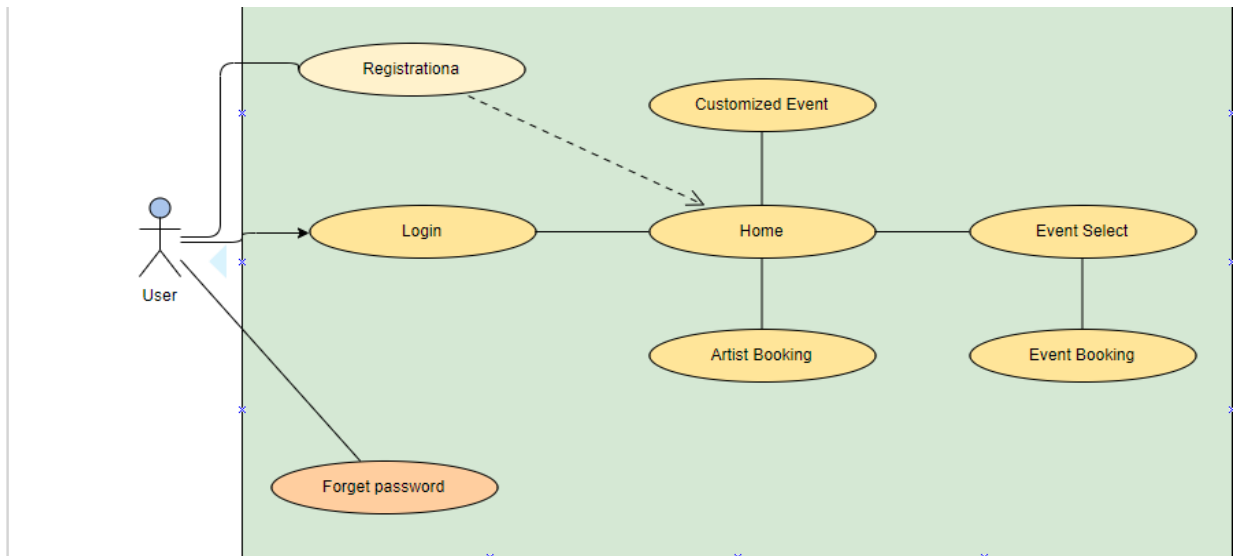


Figure 3.3: User Use Case Diagram

### 3.4 Logical Data Model

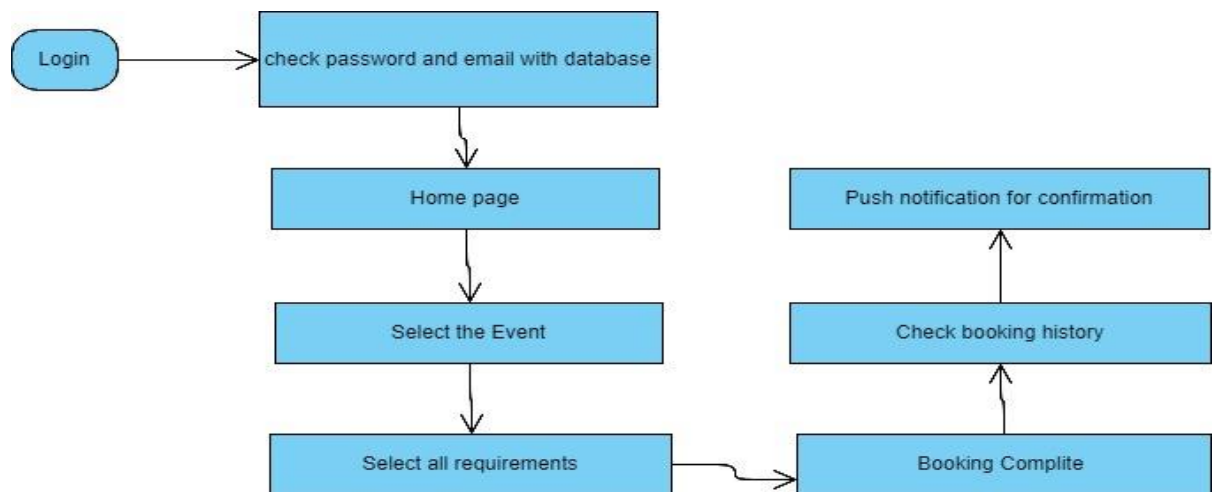


Figure 3.4: Logical data model diagram



### 3.5 Design Requirements

In this system we work in android platform.in this system user can be easily booked their desire event. We know that android is an open-source mobile operating system that is based on Linux kernel [12]. It is designed mainly for touchscreen mobile devices which include smart phones and tablet computers. Apart from its use in mobile devices, the technology has also been implemented in digital cameras, smart televisions and other electronics appliances. We used xml layout for design purpose. In Android, an XML-based layout is a file that describes the different widgets to be used in the UI and the relations between those widgets and their containers. Android treats the layout files as resources [13]. XML-based layouts are very helpful if you know the UI components at the time of compiling

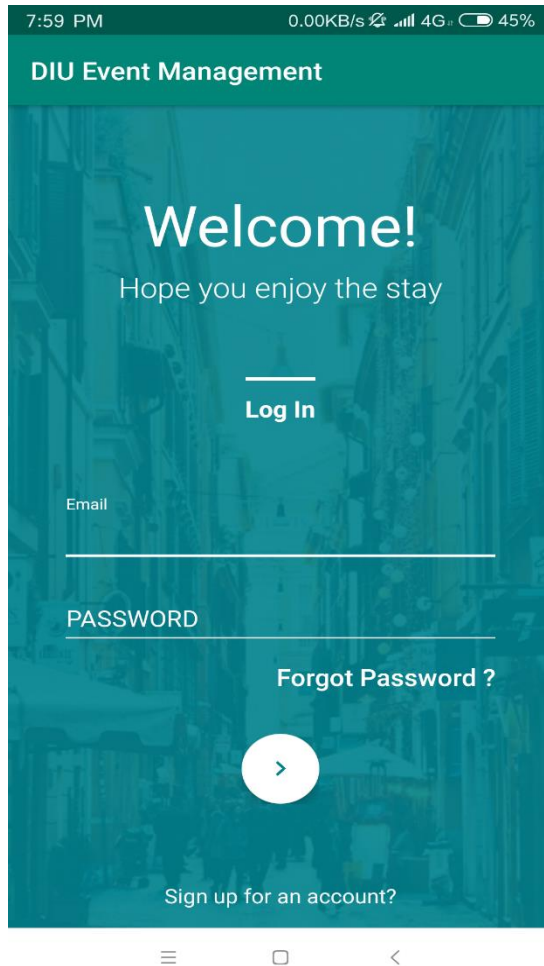
#### 1 XML-based layout has the following advantages

- XML is a very popular and commonly-used format. Later, a lot of developers are quite relaxed with it.
- It helps to provide parting of the UI from the code logic. This provided flexibility to change one without very much affecting the other.
- Producing XML output is cooler than writing direct code, making it easier to have drag-and-drop UI tools to produce interfaces for android apps [14].

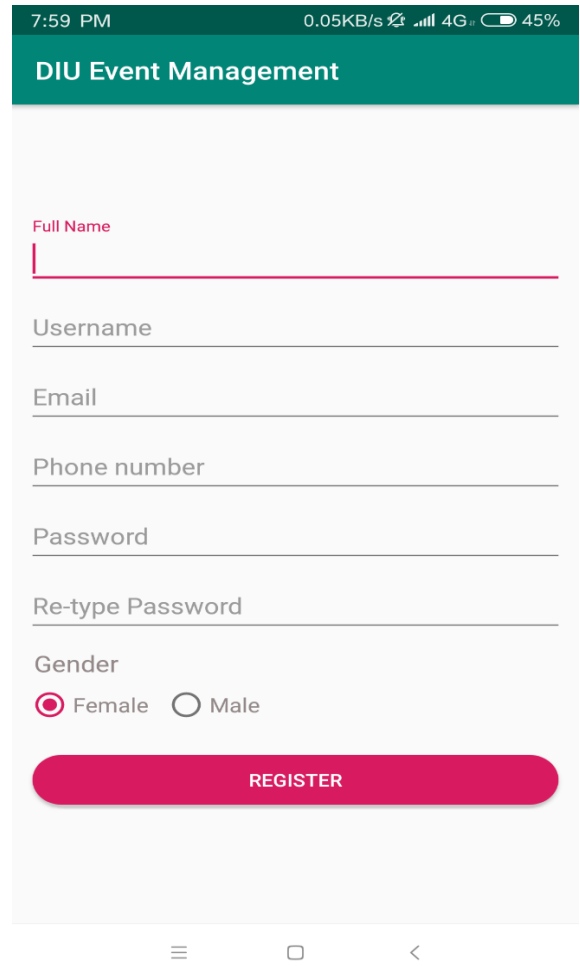
# CHAPTER 4

## Design Specification

### 4.1 Front-End Design

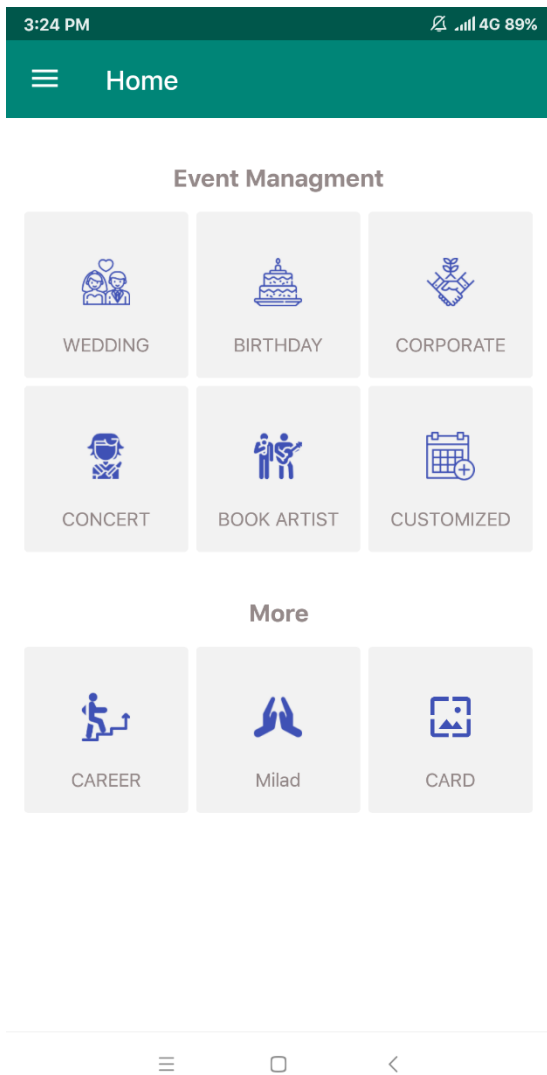


(a)

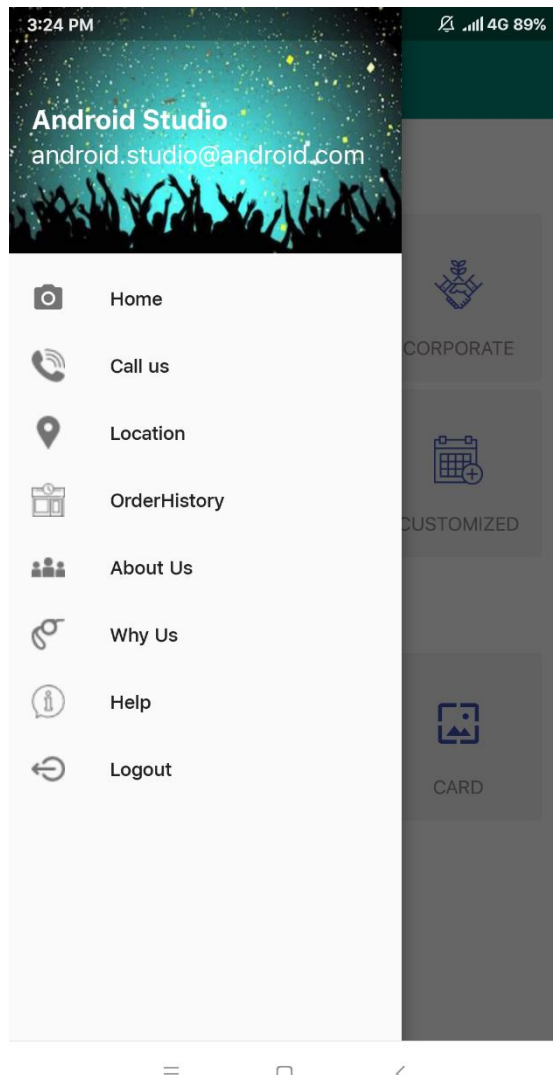


(b)

Figure 4.1: (a) Login and (b) Registration Screen

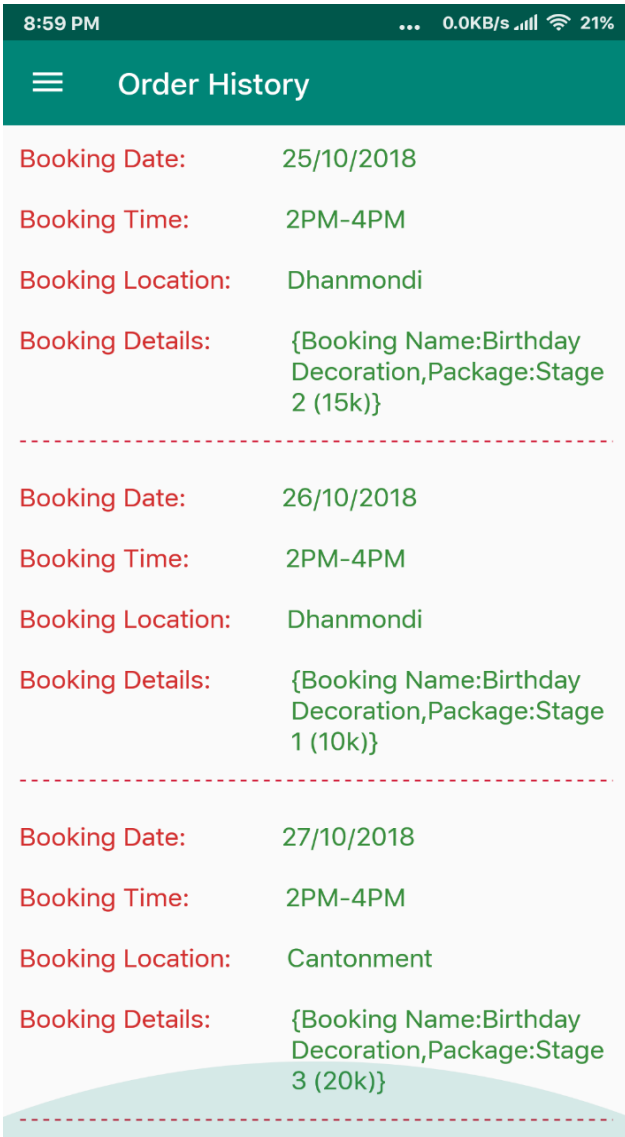


(a)

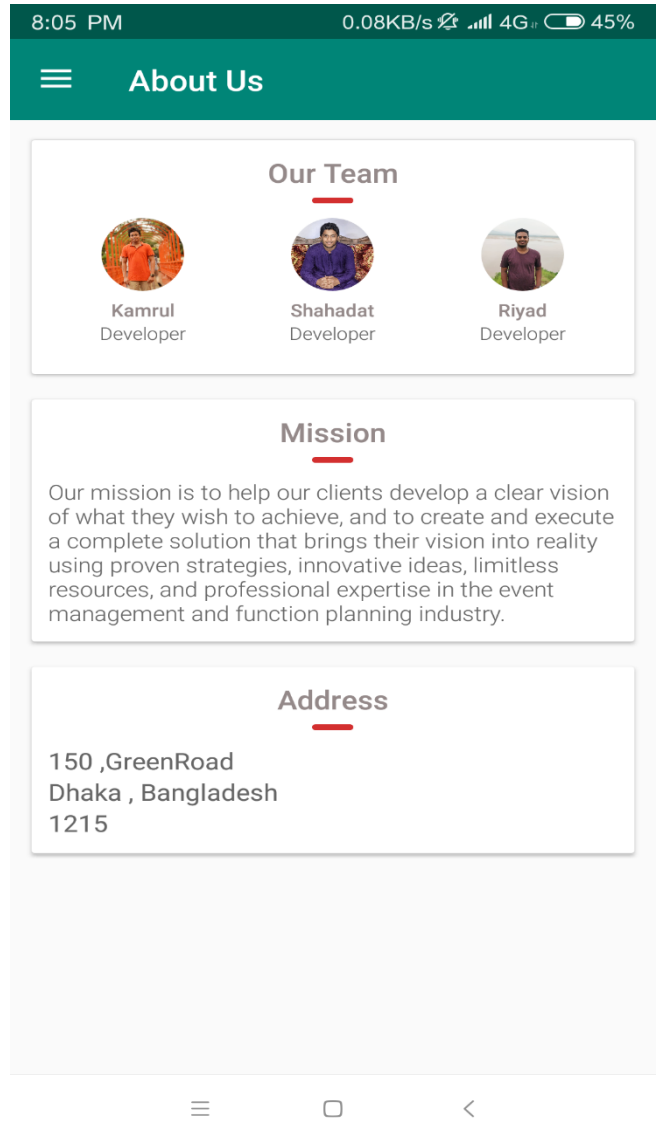


(b)

Figure 4.2: (a) Home and (b) Navigation Screen

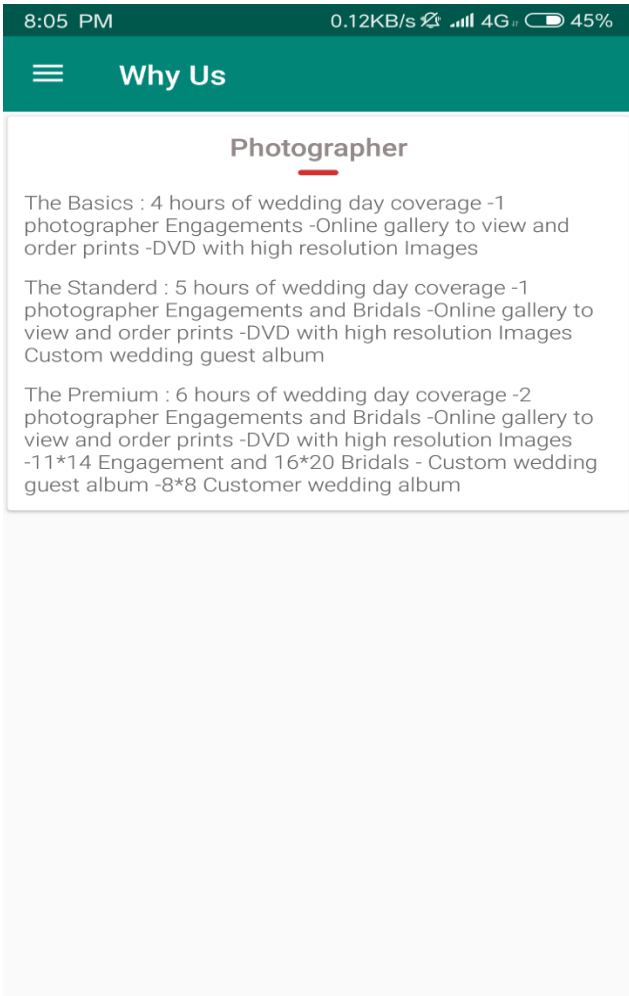


(a)

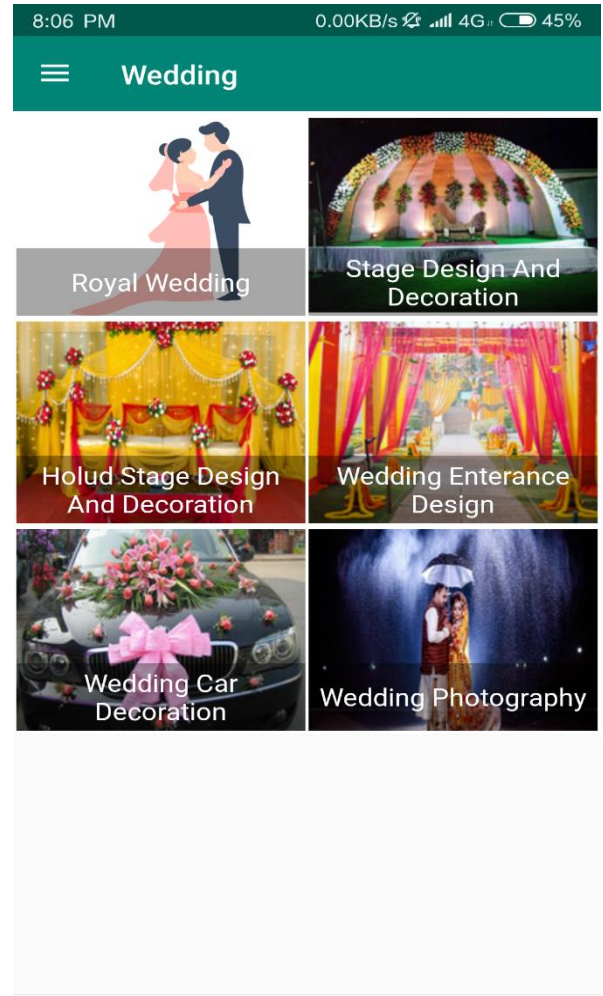


(b)

Figure 4.3: (a) Order History and (b) About Screen

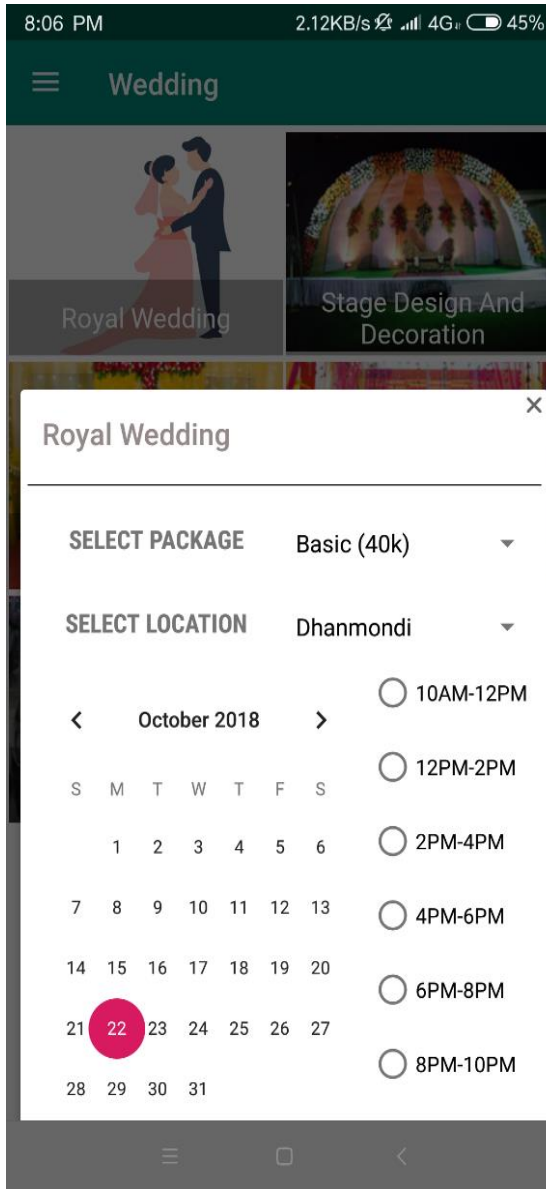


(a)

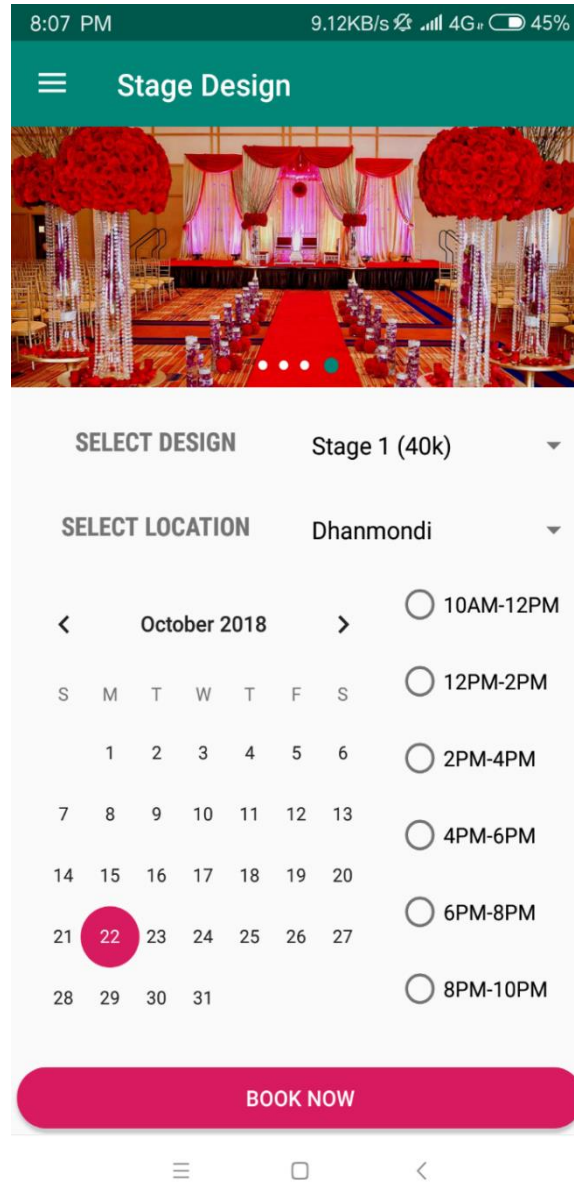


(b)

Figure 4.4: (a) Why Us Screen and (b) Wedding Screen

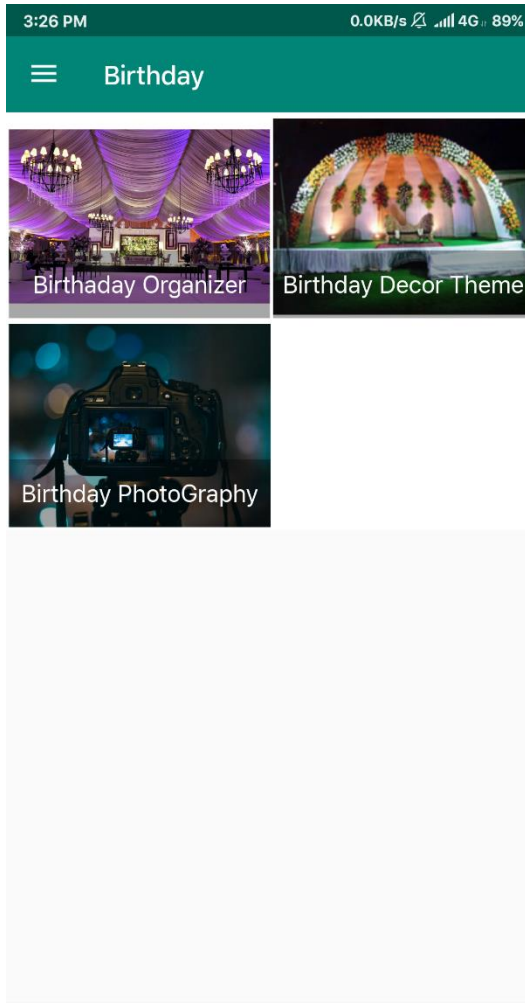


(a)

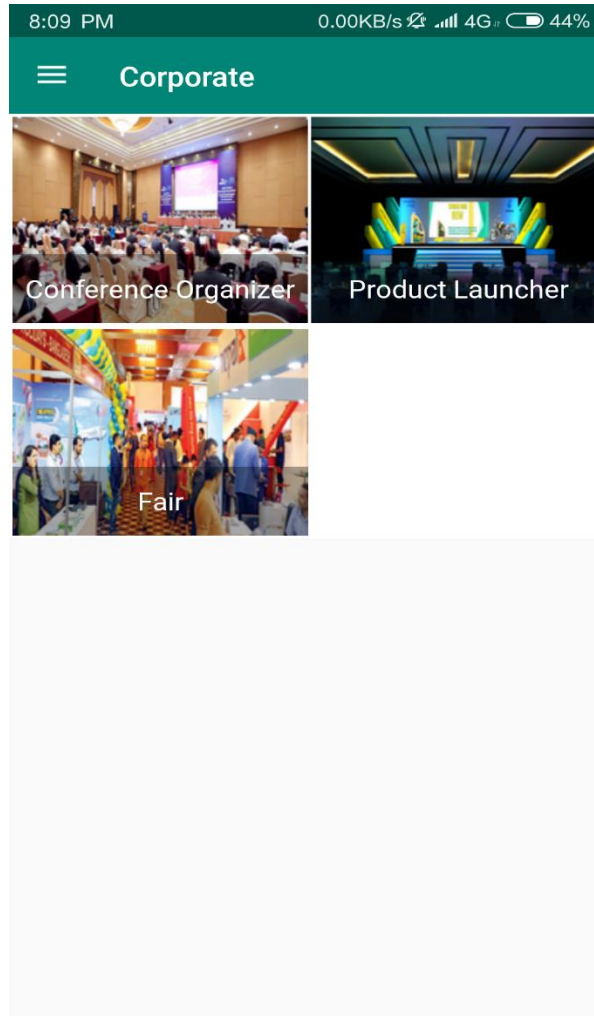


(b)

Figure 4.5: (a) Royal Wedding and (b) Stage Booking System

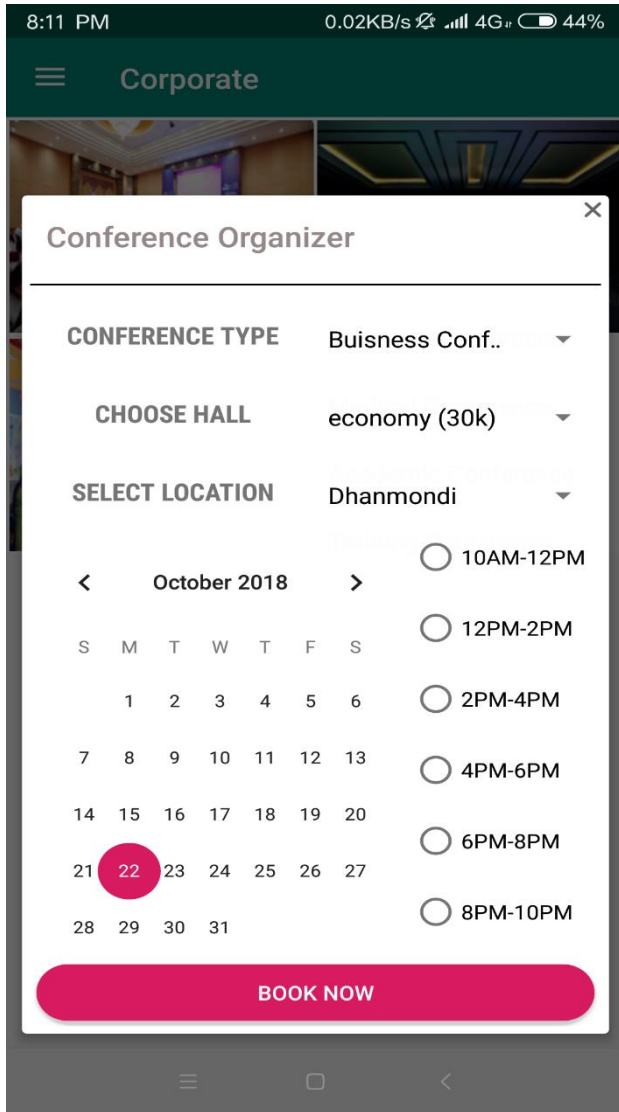


(a)

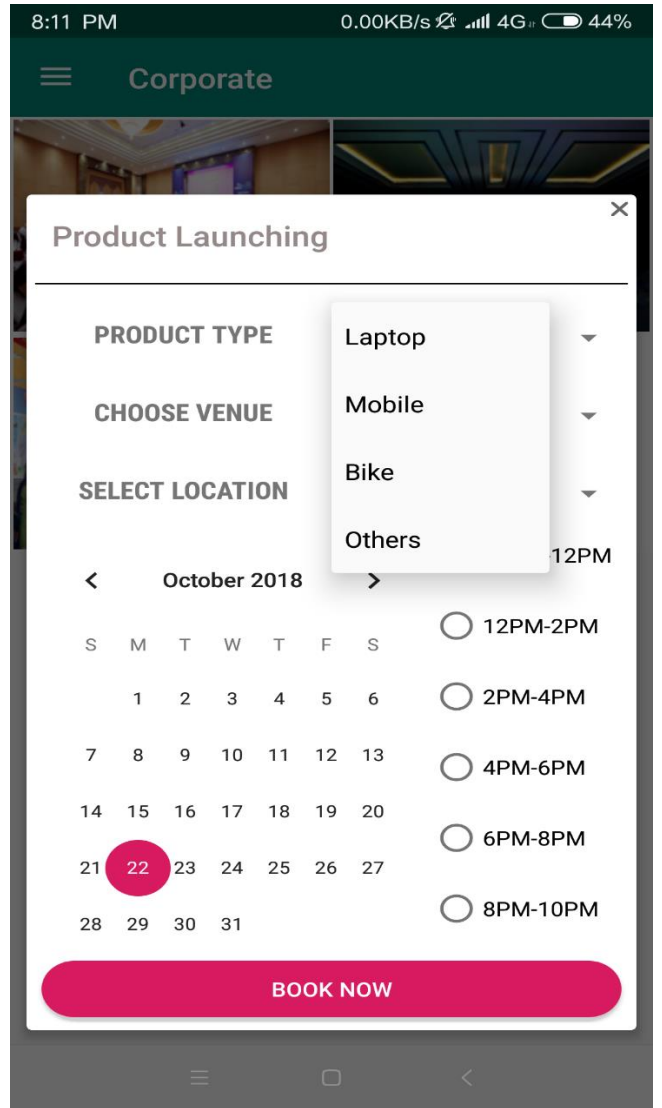


(b)

Figure 4.6: (a) Birthday and (b) Corporate Screen



(a)



(b)

Figure 4.7: (a) Conference and (b) Product Booking System



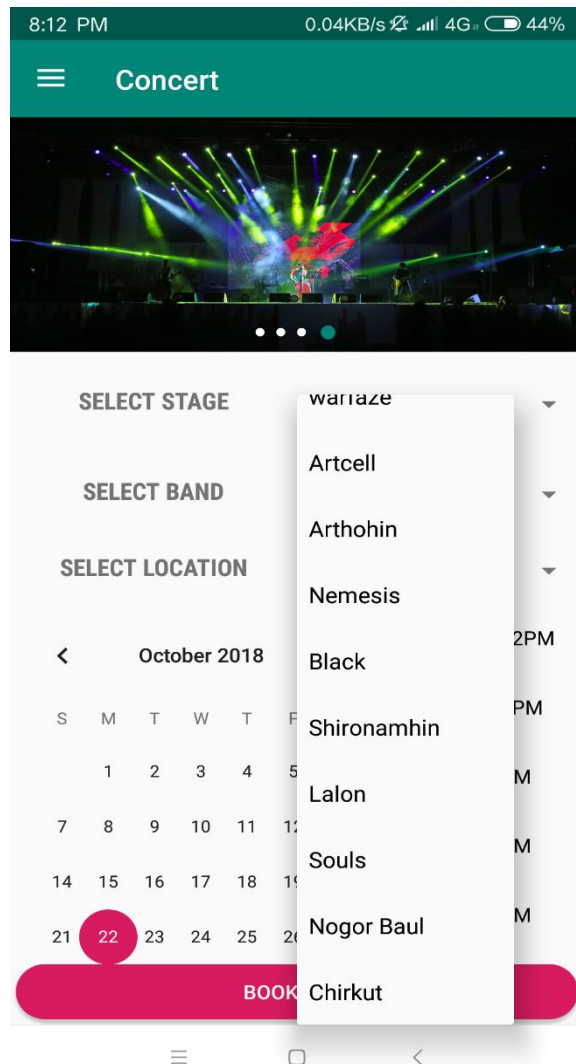
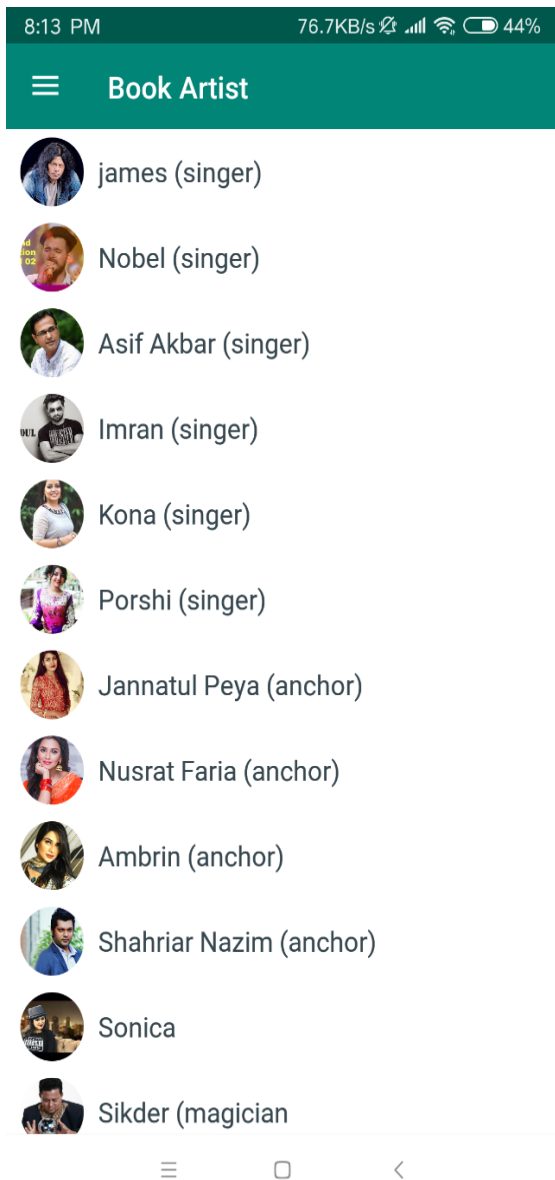
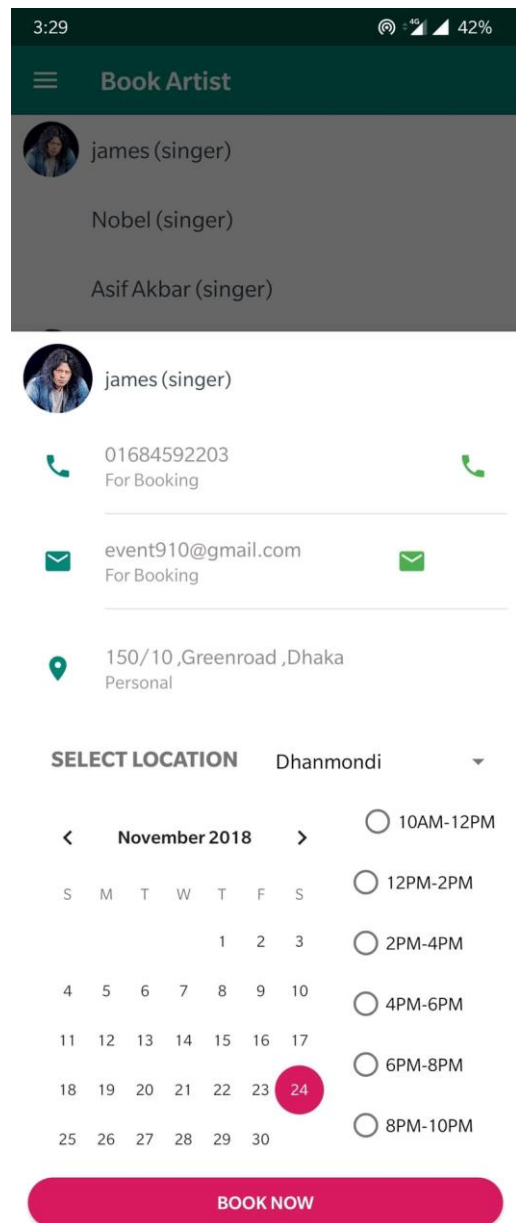


Figure 4.8: Concert System Screen

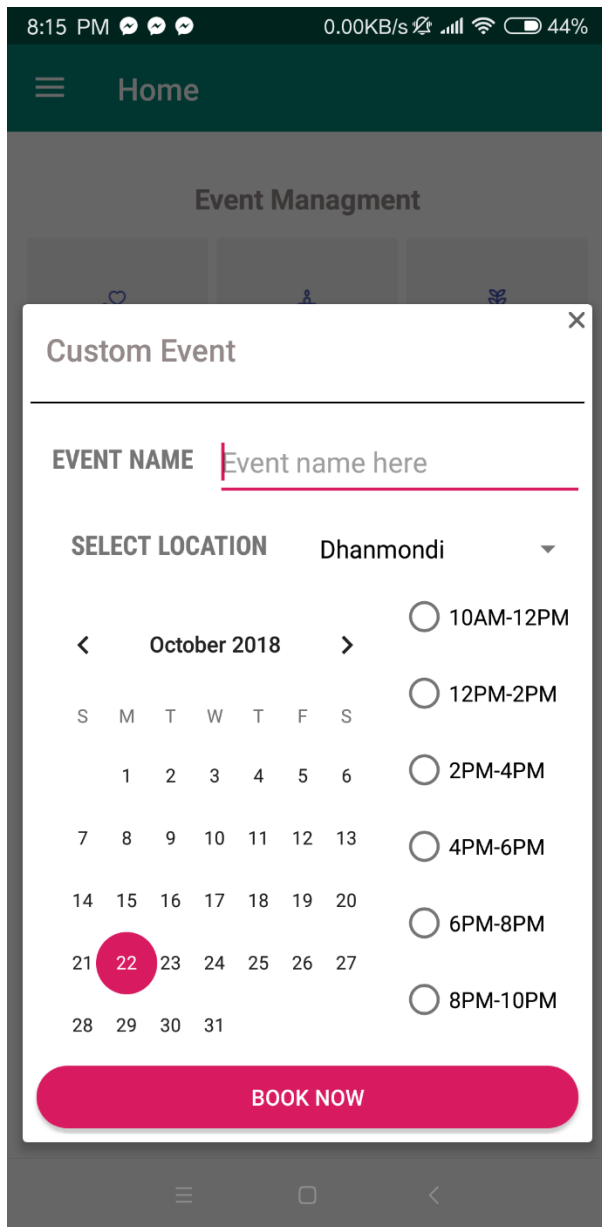


(a)

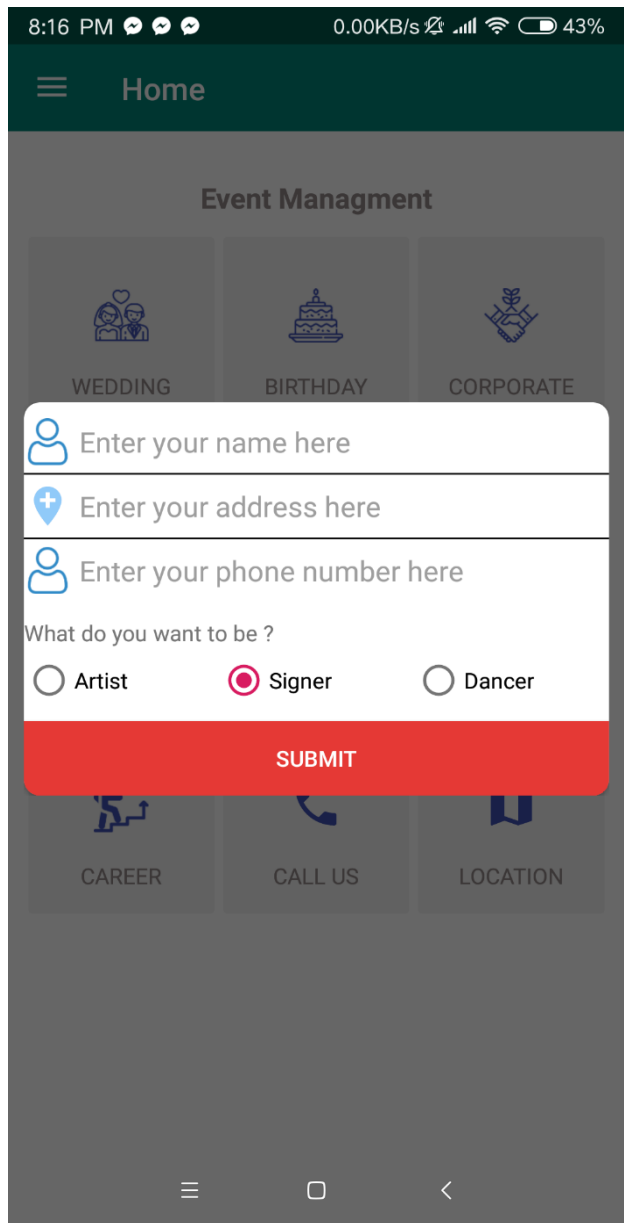


(b)

Figure 4.9: (a) and (b) Book Artist System



(a)

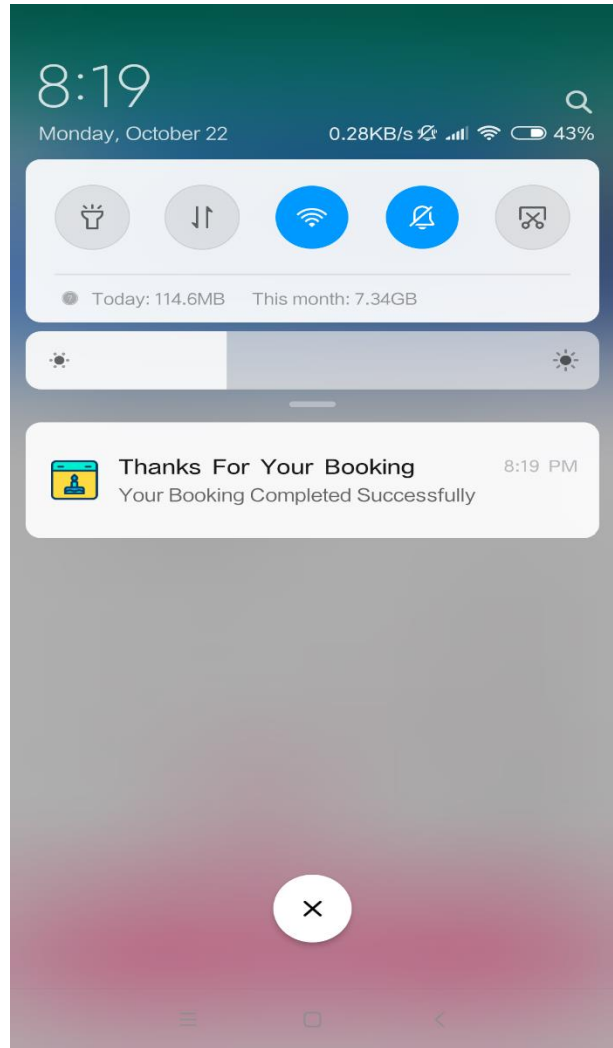


(b)

Figure 4.10: (a) Customize and (b) Career System



(a)



(b)

Figure 4.11: (a) Map and (b) Booking Confirmation Notification

## 4.2 Back-End Design

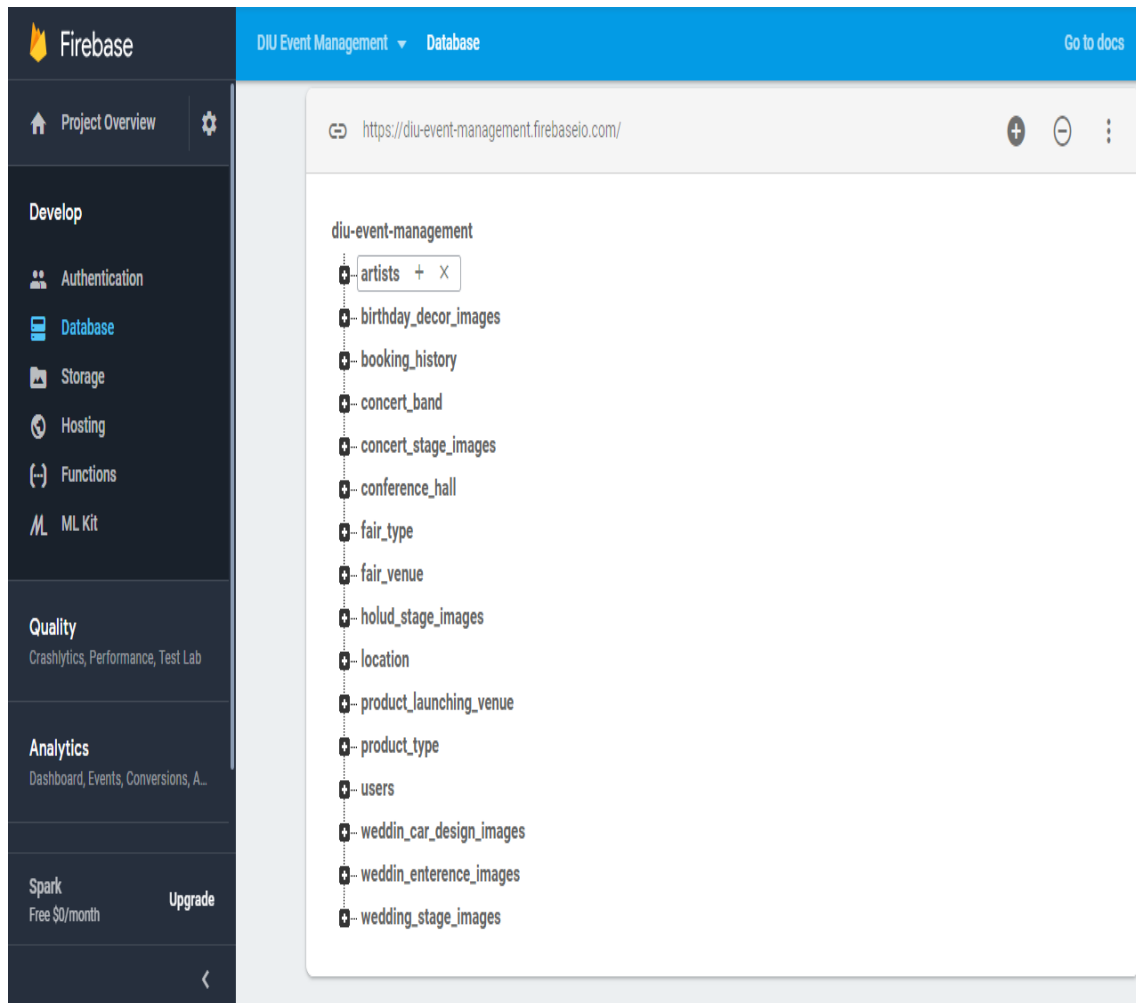


Figure 4.12: Firebase home page

```
-LPR8DyM8D8h4vK4YQ_a
  booking_date: "26/10/2018"
  booking_details: "{Booking Name:Birthday Decoration,Package:Stage"
  booking_location: "Dhanmondi"
  booking_time: "2PM-4PM"
  user_email: "kamrul@gmail.com"
  user_id: "pwUcFnddiAXqhKcecMI4Rky9m4"
  user_phone_number: "01684592203"

-LPR8Fwt7THQNJZOi11Q
  booking_date: "27/10/2018"
  booking_details: "{Booking Name:Artist Booking,Artist Name:Nusrat"
  booking_location: "Panthapath"
  booking_time: "6PM-8PM"
  user_email: "smriyad7733@gmail.co"
  user_id: "SFMEVGRm7AV0W0wAiQcuxvLKrRt"
  user_phone_number: "01618262626"
```

Figure 4.13: Booking System Database

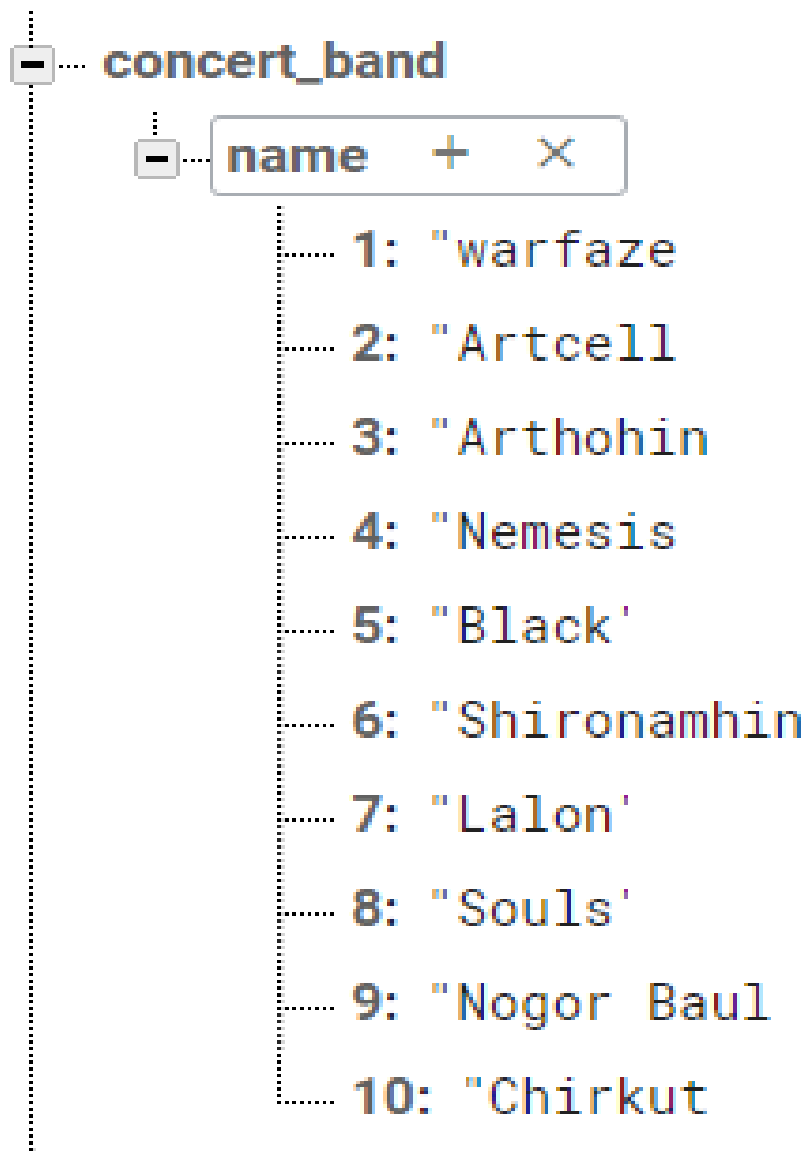


Figure 4.14: Band List Database

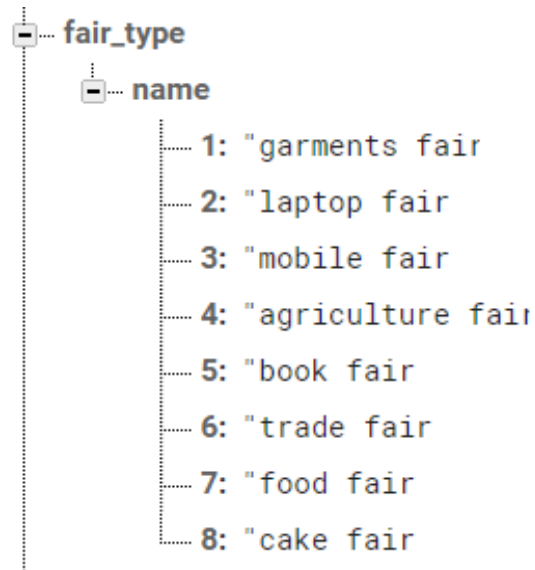


Figure 4.15: Fair Type Database

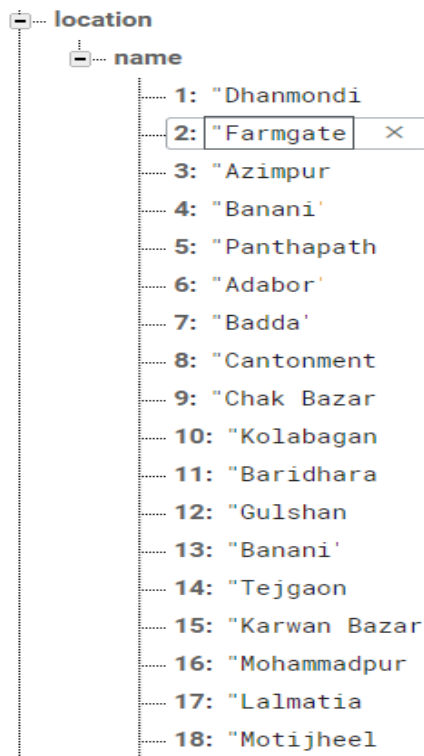


Figure 4.16: Location Database



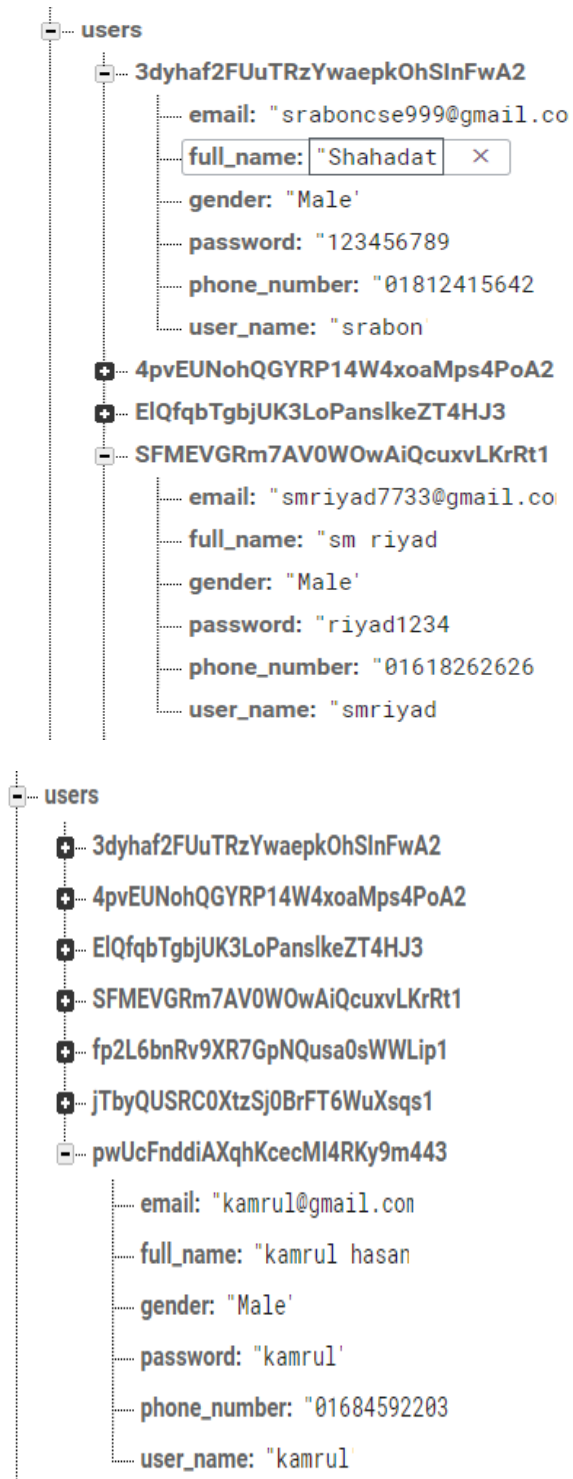


Figure 4.17: User Information Database

<input type="text" value="Search by email address, phone number, or user UID"/> <span style="float: right;"> <a href="#">Add user</a> <span style="margin-left: 10px;">↻</span> <span style="margin-left: 10px;">⋮</span> </span>				
Identifier	Providers	Created	Signed In	User UID ↑
sraboncse999@gmail.com	✉	Oct 21, 2018	Oct 21, 2018	3dyhaf2FUuTRzYwaepkOhSlnFwA2
sraboncse699@gmail.com	✉	Oct 22, 2018	Oct 22, 2018	4pvEUNohQGYP14W4xoaMps4P...
srabon00@gmail.com	✉	Oct 22, 2018	Oct 22, 2018	ElQfqbtgbiUK3LoPanslikeZT4HJ3
smriyad7733@gmail.com	✉	Oct 21, 2018	Oct 22, 2018	SFMEVGRm7AV0WOWAiQcuxvLKr...
cuteanis12@gmail.com	✉	Oct 21, 2018	Oct 21, 2018	fp2L6bnRv9XR7GpNQusa0sWWLi...
srabon@gmail.com	✉	Oct 21, 2018	Oct 21, 2018	jTbyQUSRC0XtzSj0BrFT6WuXsqs1
alamjacky334@gmail.com	✉	Oct 21, 2018	Oct 21, 2018	n5507Kv6uNhY9i62GAW7sxNqcT...
kamrul@gmail.com	✉	Oct 21, 2018	Oct 22, 2018	pwUcFnddiAXqhKcecMI4Rky9m443

Rows per page: 50 ▾ 1-8 of 8 < >

Figure 4.18: Authentication Database

### 4.3 Interaction Design and UX

Interaction design is specifically a discipline, which interaction between system and user via interface. Actually, an interaction design which can describe the user problem domain, process the problem and find the result, do the action and solve the problem [15].

On the other case, user experience focuses on the overall experience between user and an application. In our project there are some feature that can interact the user, one is career. User can develop their career using this application [16].

For UX we keep our system vary simple and user friendly and better performance to work.

## **4.4 Implementation Requirements**

Implementation is the carry out, execution and practice of a plan. It also works with a method, design and idea standard policy for doing something. Implementation is the action that must follow the preliminary thinking in order to actually occur [17].

There are few technology and tools used in the development of this project. These are: Android, java, json and Android mobile phone

### **4.4.1 Android**

Android is a mobile operating system developed by google and it's used by smartphones. Android operating system is based on the Linux kernel. Android is open source that's why different Android-based phones often in different type of graphical user interface.

Android phones typically come with several built-in applications. It's also supports third-party programs. Developers can easily create programs for Android using free Android software developer kit (SDK) [18], [19].

Android programs are written in Java and run through a Java virtual machine. And also run through a smartphone.

## CHAPTER 5

### Implementation and testing

#### 5.1 Implementation of Database

For implementation of database we used firebase from google. Firebase Realtime database is a cloud hosted database that supports multiple platforms Android, iOS and Web. All the data is stored in JSON setup and any changes in data, reflects directly by execution a sync across all the platforms & devices [20]. This allows us to build more elastic real-time apps easily with nominal exertion.

Firebase real time database is a diagram less database in which the data is stored in JSON format. Mostly, the entire database is a big JSON tree with many nodes. So when you plan your database, you need to prepare the json construction in way that the data is available in calmer way by avoiding nesting of child nodes

Firebase provides great support when comes to offline data. It automatically stores the data offline when there is no internet connection [21]. When the device connects to internet, all the data will be pushed to real-time database. However qualifying disk determination stores the data offline even though app resumes. Disk tenacity can be enabled by calling below one line code. Here is complete guide about firebase offline abilities.

To insert data, you can use `setValue()` method on to database situation path. This will create or update the value on path provided. The realtime database accepts multiple data types `String`, `Long`, `Double`, `Boolean`, `Map<String, Object>`, `List<Object>` to store the data [22]. You can also use custom java objects to store the data which is very helpful when storing model class directly in database.

To read the data, you need to attach the `ValueEventListener()` to the database reference. This event will be activated whenever there is a change in data in real time. In `onDataChange()` you can perform the desired actions onto new data.

To update data, you can use the same `setValue()` method by passing new value. You can also use `updateChildren()` by passing the path to update data without troubling other child nodes data [23].

To delete data, you can simply call `removeValue()` method on to catalogue reference. You can also pass null to `setValue()` method which do the same delete operation.

## 5.2 Implementation of Front-end Design

An Android app is made up of two parts: the front end and the back end. The front end is the visual part of the app that the user interacts with, and the back end, which contains all the code that drives the app.

### Front-end:

1. It's all about what a user able to see.
2. It's may be buttons, images, layout, input field etc.
3. To design these components in android you need XML and Android programming. Therefore, these programming are called front-end programming [24].

### Example For Login Button, Registration Button and Progress bar:

```
<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal">

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/loginBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:clickable="true"
        android:textStyle="bold"
        android:tint="@color/cyan_800"
        app:backgroundTint="@android:color/white"
        app:fabSize="normal"
        app:rippleColor="@color/cyan_800"
        app:srcCompat="@drawable/ic_arrow_right"/>

    <ProgressBar
        android:id="@+id/progress_bar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:theme="@style/ProgressBar.White"
        android:visibility="gone" />

</RelativeLayout>
```

```

<View
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_weight="1" />

<LinearLayout
    android:id="@+id/sign_up_for_account"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/spacing_medium"
    android:gravity="center"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign up for an account?"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Subhead"
        android:textColor="@color/grey_50" />

</LinearLayout>

```

### Home Button:

```

android:id="@+id/wedding"
style="@style/RippleStyleBlack"
android:layout_width="match_parent"
android:layout_height="wrap_content">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:clickable="true"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="15dp">

    <ImageView
        android:layout_width="35dp"
        android:layout_height="35dp"
        android:layout_marginTop="18dp"
        android:layout_marginBottom="18dp"
        android:tint="@color/indigo_500"
        app:srcCompat="@drawable/wedding"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="WEDDING"
        android:textAlignment="center"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"
        android:textColor="@color/grey_90" />

</LinearLayout>

```

### **5.3 Implementation of Interactions**

Process of manipulating the factors that avoid unified change control so that only accepted changes are realized

- Process to review, analyze, and approve change requests - timely answer minimizes the opportunity of harmful effect on time, cost, or the possibility of a change
- Involves managing the changes that are agreed
- Involves preserving integrity of the presentation extent baselines
- Involves supporting the entire impact fashioned by the change request
- Process to review, approve/reject suggested preventive, or corrective actions and Involves organizing changes across the complete project [25]

### **5.4 Testing Implementation**

The purpose of testing is to determine faults. Testing is the procedure of trying to discover every possible fault or weakness in a work product. It provides a way to check the functionalities of mechanisms, sub-assemblies, and/or a finished product it is the procedure of exercise software with the intent of safeguarding that the software system meets its supplies and user prospects and does not fail in an improper manner [26]. There are various types of tests. Each test type addresses an exact testing requirement.

#### **5.4.1 Types of Testing**

##### **Unit Testing**

Unit testing focuses confirmation effort on the smallest unit of software design, the module. The unit testing, we have is white box concerned with and some modules the steps are conducted in parallel [27].

### **Integration Testing:**

Testing is done for each module. After testing all the components, the modules are united and testing of the final system is done with the test data, specially intended to show that the system will function successfully in all its aspects situations [28]. Thus, the system testing is a approval that all is correct and an occasion to show the user that the system works. The purpose of addition testing is to verify functional, performance and reliability requirements placed on major design items.

### **System Testing:**

System testing ensures that the entire united software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system addition test [29]. System testing is based on process metaphors and flows, stressing pre-driven process links and incorporation points. We have tested unit testing. Finally, we have come a successfully conclusion.

## **5.5 Test Results and Reports**

### **Unit Testing Reports:**

**Unit testing:** unit testing is a level of software testing where individual units/ components of a software are tested [30].

#### **5.5.1 Test Case 1 (Check Login page)**

**Table 5.1: Test Case for login**

Sl No. of test case:	1
Name of test:	login test
Sample Input :	Click on login that runs LoginActivity.java
Expected output :	login GUI should display
Actual output :	login GUI displayed as expected
Remarks:	Test Successful



### 5.5.2 Test Case 2 (Check Home page)

**Table 5.2: Test Case for Check Home page**

SI No. of test case:	2
Name of test :	Check Home page test case
Sample Input :	Clicking the Login Button that runs MainActivity.java
Expected output :	Home page details are viewed.
Actual output :	Home page details are viewed.
Remarks:	Test case successful

### 5.5.3 Test Case 3 (Check Booking System)

**Table 5.3: Test Case for Check Booking System**

SI No. of test case:	3
Name of test :	Check Booking System test case
Sample Input :	Clicking the Booking Button that runs confirmation toast
Expected output :	Confirmation toast and notification are showed.
Actual output :	Confirmation toast and notification are showed.
Remarks:	Test case successful

## **CHAPTER 6**

### **Conclusion and future scope**

#### **6.1 Discussion and Conclusion**

Event management is a nice-looking and inspiring profession which weights a lot of hard work and dynamism. As the name advises, it means abstracting, planning, establishing and finally performing etc. This business is just new in Bangladesh, but holds a lot of promise for extension. It offers huge scope for ambitious young people. Event is the planning and application of events, large and small that meet the promotion goals of a group. Event management is an area that is growing rapidly, and is likely to have a better growth rate in the next decade. Typical events ready by professional event managers include product launches, parties, sponsored events, sporting events, competitions, concerts and festivals, fundraisers, trade shows, corporate open days, seminars and tours. Event management sector have greater potentiality and scope to develop and can pay for service as well as economical, business and cultural growth of Bangladesh.

#### **6.2 Scope for Further Developments**

In future we added invoice system that user gets total cost of their booked event. Also, we want to add online payment gateway system so that user can easily pay their required payment via master card, visa card, bkash, rocket or any other online payment system. In future we add live chat feature in our application that user can easily contact with admit and moderator for any kind of information.

## REFERENCES

- [1] “Understanding outcomes”, available at << <https://www.collinsdictionary.com/amp/english/outcome> >> last accessed on 08-02-2018 at 11.51 am.
- [2] “Definition of comparative study”, available at << <http://www2.uiah.fi/projekti/metodi/172.htm> >> last accessed on 08-02-2018 at 12.15 pm.
- [3] “Business Process Modeling”, available at << <https://kissflow.com/bpm/business-process-modeling/> >> last accessed on 08-02-2018 at 12.40 pm.
- [4] “Software Requirements Analysis”, available at << <https://www.guru99.com/learn-software-requirements-analysis-with-case-study.html> >> last accessed on 08-02-2018 at 1.15 pm.
- [5] “Use Case Model”, available at << <http://www.gatherspace.com/static/use-case-model.html> >> last accessed on 08-04-2018 at 4.21 pm.
- [6] “Front-end-development”, available at << <https://www.coursereport.com/blog/front-end-development-vs-back-end-development-where-to-start> >> last accessed on 08-06-2018 at 4.45 pm
- [7] “Interaction & UX design”, available at << <https://uxplanet.org/interaction-design-and-the-five-pillars-ad319bd82f1> >> last accessed on 08-06-2018 at 9.45 pm
- [8] “Android Definition”, available at << <https://techterms.com/definition/android> >> last accessed on 08-09-2018 at 11.45 am
- [9] “Basic about android”, available at < <https://developer.android.com/> > last accessed on 08-09-2018 at 3.25 pm
- [10] “For android related information videos”, available at << <https://www.youtube.com/androiddevelopers> >> last accessed on 08-15-2018 at 10.15 am.
- [11] “Android Solution Forum”, available at << <https://stackoverflow.com/questions/tagged/android> >> last accessed on 08-17-2018 at 7.35 pm.
- [12] “Basic about Firebase”, available at << <https://angularfirebase.com/lessons/the-ultimate-beginners-guide-to-firebase/> >> last accessed on 08-18-2018 at 9.35 pm.
- [13] “Fragment”, available at << <https://developer.android.com/guide/components/fragments> >> last accessed on 08-18-2018 at 10.35 pm.
- [14] “Android SDK”, available at << [https://www.webopedia.com/TERM/A/Android\\_SDK.html](https://www.webopedia.com/TERM/A/Android_SDK.html) >> last accessed on 08-20-2018 at 10.35 am.
- [15] “Software Testing”, available at << [https://www.tutorialspoint.com/software\\_testing/](https://www.tutorialspoint.com/software_testing/) >> last accessed on 08-20-2018 at 12.35 pm.
- [16] “Software Testing”, available at << <http://softwaretestingfundamentals.com/system-testing/> >> last accessed on 08-25-2018 at 8.25 pm.
- [17] “JSON Basic”, available at << <https://www.tutorialspoint.com/json> >> last accessed on 09-03-2018 at 9.33 am.
- [18] Herbert Schildt, “Java: A Beginner's Guide, Sixth Edition 6th Edition” 201-237, April 2001

- [19] Barry Burd “Android Application Development All-in-One For Dummies 2nd Edition” 107-147, March 2003
- [20] Alan Dennis (Author), Barbara Haley Wixom (Author), Roberta M. Roth (Author) “Systems Analysis and Design, 6th Edition” 67-117, June 2007
- [21] “Event Management Package”, available at << <http://www.bdweddingplanners.com/>>> last accessed on 09-11-2018 at 8.35 pm.
- [22] “Logical Data Model”, available at << <http://www.bdweddingplanners.com/>>> last accessed on 09-14-2018 at 2.35 pm.
- [23] “Photo Resizer”, available at << <https://www.photoresizer.com/>>> last accessed on 09-17-2018 at 10.25 am.
- [24] “Icon Design”, available at << <https://www.flaticon.com/>>> last accessed on 09-25-2018 at 11.45 am.
- [25] “App Icon Generator”, available at << <https://appicon.co/>>> last accessed on 09-29-2018 at 9.14 pm.
- [26] “Google Map API”, available at << <https://cloud.google.com/maps-platform/>>> last accessed on 10-05-2018 at 9.55 am.
- [27] “Startup Animation”, available at << <https://www.lottiefiles.com/>>> last accessed on 10-9-2018 at 2.55 pm.
- [28] “Event package details”, available at << <http://www.bdweddingplanners.com/>>> last accessed on 10-12-2018 at 10.55 am.
- [29] “Run apps on the Android Emulator”, available at <<<https://developer.android.com/studio/run/emulator>>> last accessed on 10-14-2018 at 6.55 pm.
- [30] “Unit testing ”, available at << <http://softwaretestingfundamentals.com/unit-testing/>>> last accessed on 10-15-2018 at 9.15 pm.

## APPENDICES

### Appendix A:

Here some back-end code for login page:

```
public class LoginActivity extends AppCompatActivity {

    LinearLayout signUpText;

    private EditText inputEmail, inputPassword;
    private FirebaseAuth auth;
    private ProgressBar progressBar;
    private FloatingActionButton btnLogin;
    private Button btnResetPassword;

    String inputEmailText,inputPasswordText;

    public final Pattern EMAIL_ADDRESS_PATTERN = Pattern.compile(
        "[a-zA-Z0-9+._%~+]{1,256}" +
        "@" +
        "[a-zA-Z0-9][a-zA-Z0-9-]{0,64}" +
        "(" +
        "." +
        "[a-zA-Z0-9][a-zA-Z0-9-]{0,25}" +
        ")+"
    );

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        auth = FirebaseAuth.getInstance();

        if (auth.getCurrentUser() != null) {
            startActivity(new Intent(LoginActivity.this, MainActivity.class));
            finish();
        }

        signUpText = findViewById(R.id.sign_up_for_account);
        inputEmail = findViewById(R.id.userEmail);
        inputPassword = findViewById(R.id.userPassword);
        btnLogin = findViewById(R.id.loginBtn);
        btnResetPassword = findViewById(R.id.btn_reset_password);
        progressBar = findViewById(R.id.progress_bar);

        btnResetPassword.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this, ResetPasswordActivity.class));
            }
        });
    }
}
```

```

signUpText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(LoginActivity.this,SignUpActivity.class));
    }
});

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        inputEmailText = inputEmail.getText().toString();
        inputPasswordText = inputPassword.getText().toString();

        if (TextUtils.isEmpty(inputEmailText))
        {
            inputEmail.setError("Required");
            return;
        }
        if (TextUtils.isEmpty(inputPasswordText))
        {
            inputPassword.setError("Required");
            return;
        }

        if (inputPasswordText.length()<6)
        {
            inputPassword.setError("Password must be 6 character or more");
            return;
        }

        if(!EMAIL_ADDRESS_PATTERN.matcher(inputEmailText).matches()){
            inputEmail.setError("Please use a valid email address");
            return;
        }

        btnLogin.setVisibility(View.GONE);
        progressBar.setVisibility(View.VISIBLE);

        auth.signInWithEmailAndPassword(inputEmailText,inputPasswordText)
            .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {

                    if (task.isSuccessful())
                    {
                        progressBar.setVisibility(View.GONE);
                        startActivity(new Intent(LoginActivity.this,MainActivity.class));
                        finish();
                    } else
                    {
                        btnLogin.setVisibility(View.VISIBLE);
                        progressBar.setVisibility(View.GONE);
                        Toast.makeText(LoginActivity.this, "Something went wrong!!",
Toast.LENGTH_SHORT).show();
                    }

                }
            });

```

```

    }
  });
}

```

Back-end code for Registration from:

```

public class SignUpActivity extends AppCompatActivity {

    FirebaseDatabase firebaseDatabase;
    DatabaseReference user;

    private EditText fullName,userName,email,phoneNumber,password,confirmPassword;
    private Button btnSignUp;
    private ProgressBar progressBar;
    private FirebaseAuth auth;
    private RadioGroup gender;
    private RadioButton radioButton;
    private String
fullNameTxt,userNameTxt,emailTxt,phoneTxt,passwordTxt,confirmPasswordTxt,genderTxt;
    private String userID;

    public final Pattern EMAIL_ADDRESS_PATTERN = Pattern.compile(
        "[a-zA-Z0-9+._%{-+]{1,256}" +
        "@" +
        "[a-zA-Z0-9][a-zA-Z0-9-]{0,64}" +
        "(" +
        "." +
        "[a-zA-Z0-9][a-zA-Z0-9-]{0,25}" +
        ")+"
    );

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up);

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        //Initialize firebase database
        firebaseDatabase = FirebaseDatabase.getInstance();
        user = firebaseDatabase.getReference();

        fullName = findViewById(R.id.fullName);
        userName = findViewById(R.id.userName);
        email = findViewById(R.id.email);
        phoneNumber = findViewById(R.id.phoneNumber);
        password = findViewById(R.id.password);
        confirmPassword = findViewById(R.id.confirmPasword);
        btnSignUp = findViewById(R.id.registerBtn);
        progressBar = findViewById(R.id.sign_up_progress);
        gender = findViewById(R.id.gender);
    }
}

```

```

btnSignUp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        fullNameTxt = fullName.getText().toString();
        userNameTxt = userName.getText().toString();
        emailTxt = email.getText().toString();
        phoneTxt = phoneNumber.getText().toString();
        passwordTxt = password.getText().toString();
        confirmPasswordTxt = confirmPassword.getText().toString();

        if (TextUtils.isEmpty(fullNameTxt))
        {
            fullName.setError("Required");
            return;
        }
        if (TextUtils.isEmpty(userNameTxt))
        {
            userName.setError("Required");
            return;
        }
        if (TextUtils.isEmpty(emailTxt))
        {
            email.setError("Required");
            return;
        }
        if (TextUtils.isEmpty(phoneTxt))
        {
            phoneNumber.setError("Required");
            return;
        }

        if (TextUtils.isEmpty(passwordTxt))
        {
            password.setError("Required");
            return;
        }
        if (TextUtils.isEmpty(confirmPasswordTxt))
        {
            confirmPassword.setError("Required");
            return;
        }
        if (phoneTxt.length() < 11 && phoneTxt.length() > 11)
        {
            phoneNumber.setError("Phone number not valid");
            return;
        }

        if (passwordTxt.length() < 6)
        {
            password.setError("Password must be 6 character or more");
            return;
        }

        if (!EMAIL_ADDRESS_PATTERN.matcher(emailTxt).matches()){
            email.setError("Please use a valid email address");
        }
    }
}

```



```

        return;
    }
    else if(!passwordTxt.equals(confirmPasswordTxt)){
        confirmPassword.setError("Password not match..");
        return;
    }

    int selectedId = gender.getCheckedRadioButtonId();
    //find the radiobutton by returned id
    radioButton = findViewById(selectedId);
    genderTxt = radioButton.getText().toString();

    progressBar.setVisibility(View.VISIBLE);

    auth.createUserWithEmailAndPassword(emailTxt,passwordTxt)
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                if (task.isSuccessful())
                {
                    userID = auth.getCurrentUser().getUid();
                    User users = new User(fullNameTxt,
                        userNameTxt,
                        emailTxt,
                        phoneTxt,
                        passwordTxt,
                        genderTxt);

                    user.child("users").child(userID).setValue(users, new
DatabaseReference.CompletionListener() {
                        @Override
                        public void onComplete(@Nullable DatabaseError databaseError, @NonNull
DatabaseReference databaseReference) {
                            progressBar.setVisibility(View.GONE);
                            startActivity(new Intent(SignUpActivity.this,MainActivity.class));
                        }
                    });

                }else
                {
                    progressBar.setVisibility(View.GONE);
                    Toast.makeText(SignUpActivity.this, "Something went wrong!!",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}

```

