3.2 Process flow/Representation

Following are activity diagrams of "**Face Fit**".

3.2.1 Sign In and Sign Up Activity for Customer
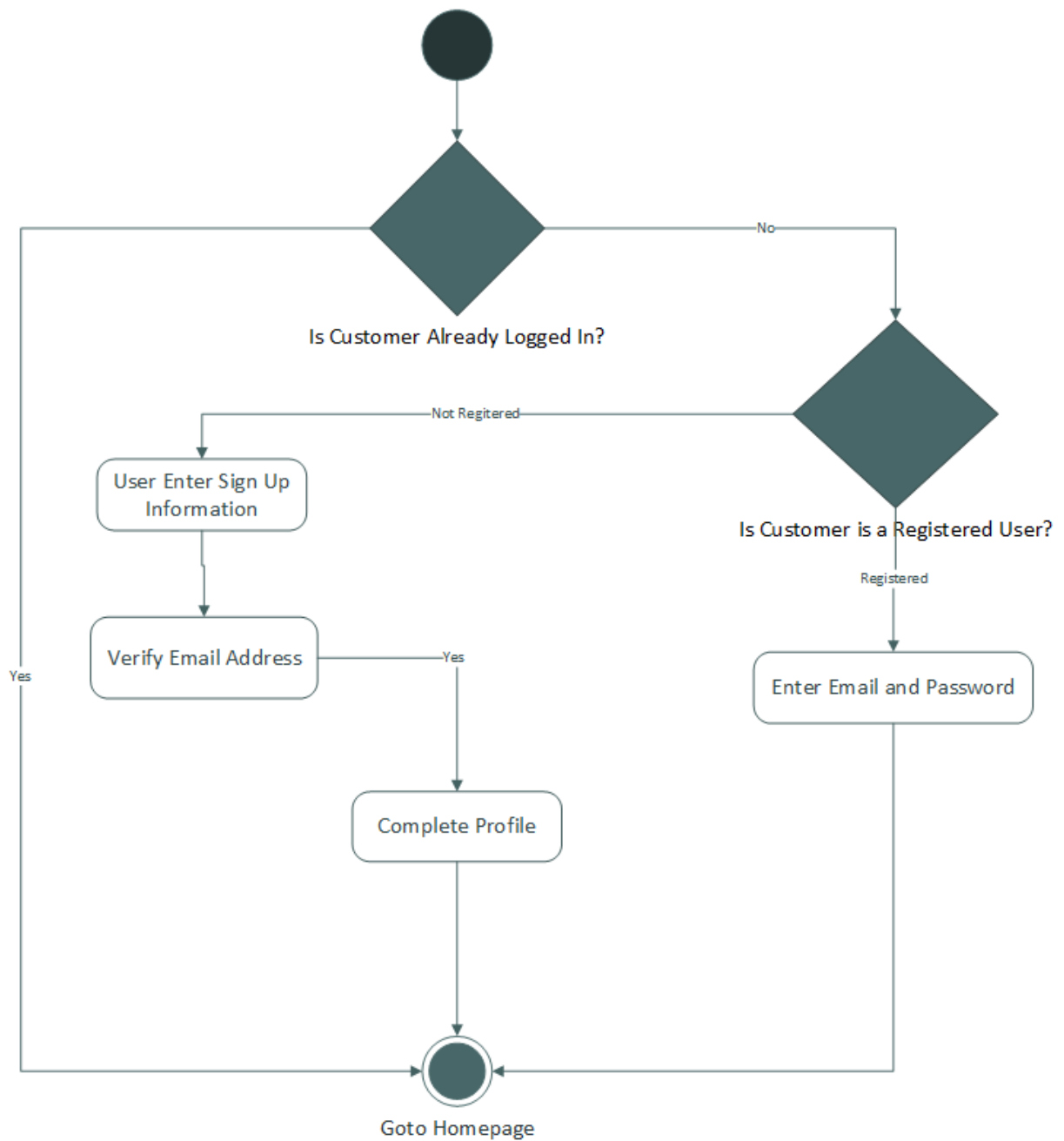
]

Figure 2-Sign In and Sign Up Activity for Customer
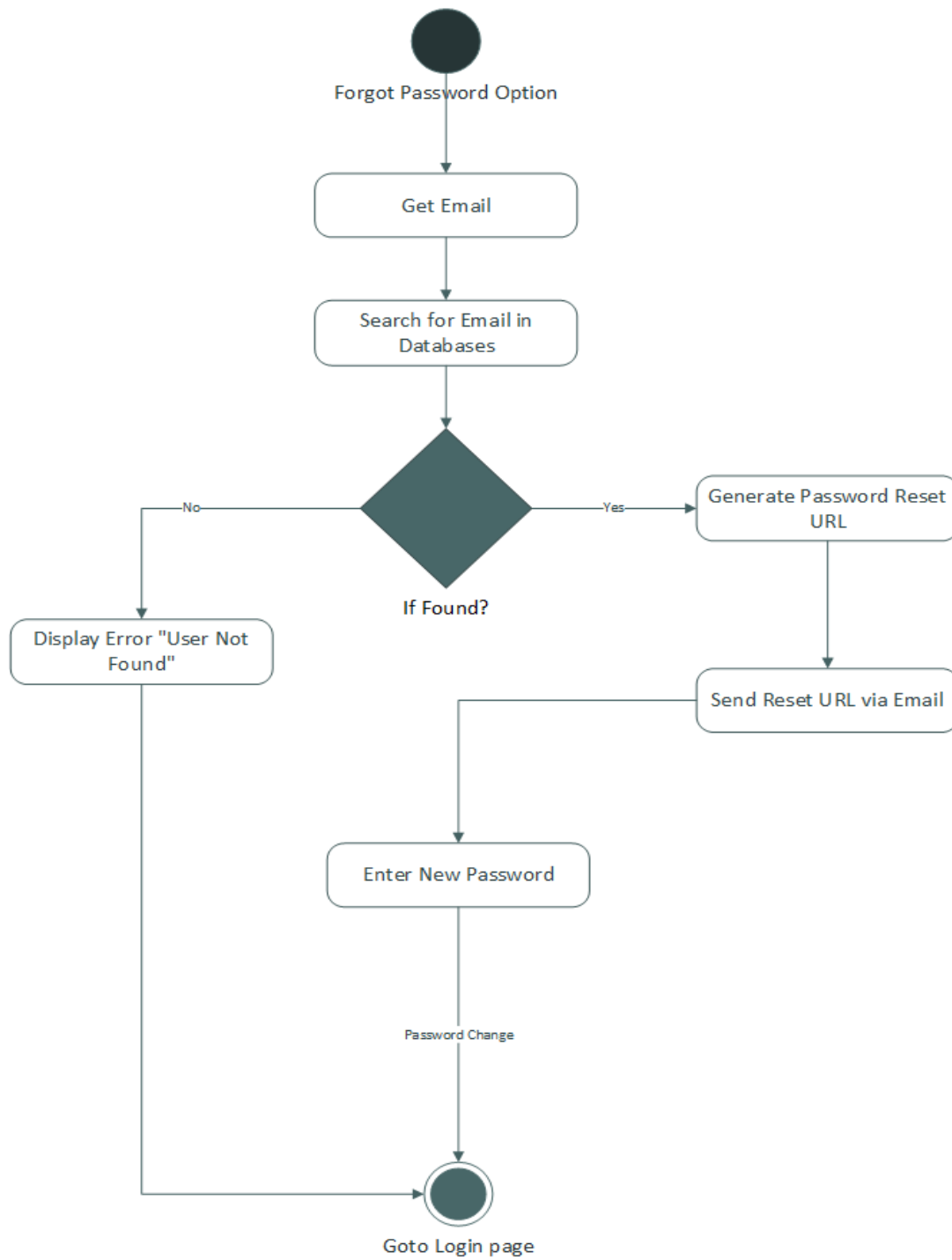
3.2.2 Forgot Password for Customer

Figure 3-Forgot Password for Customer

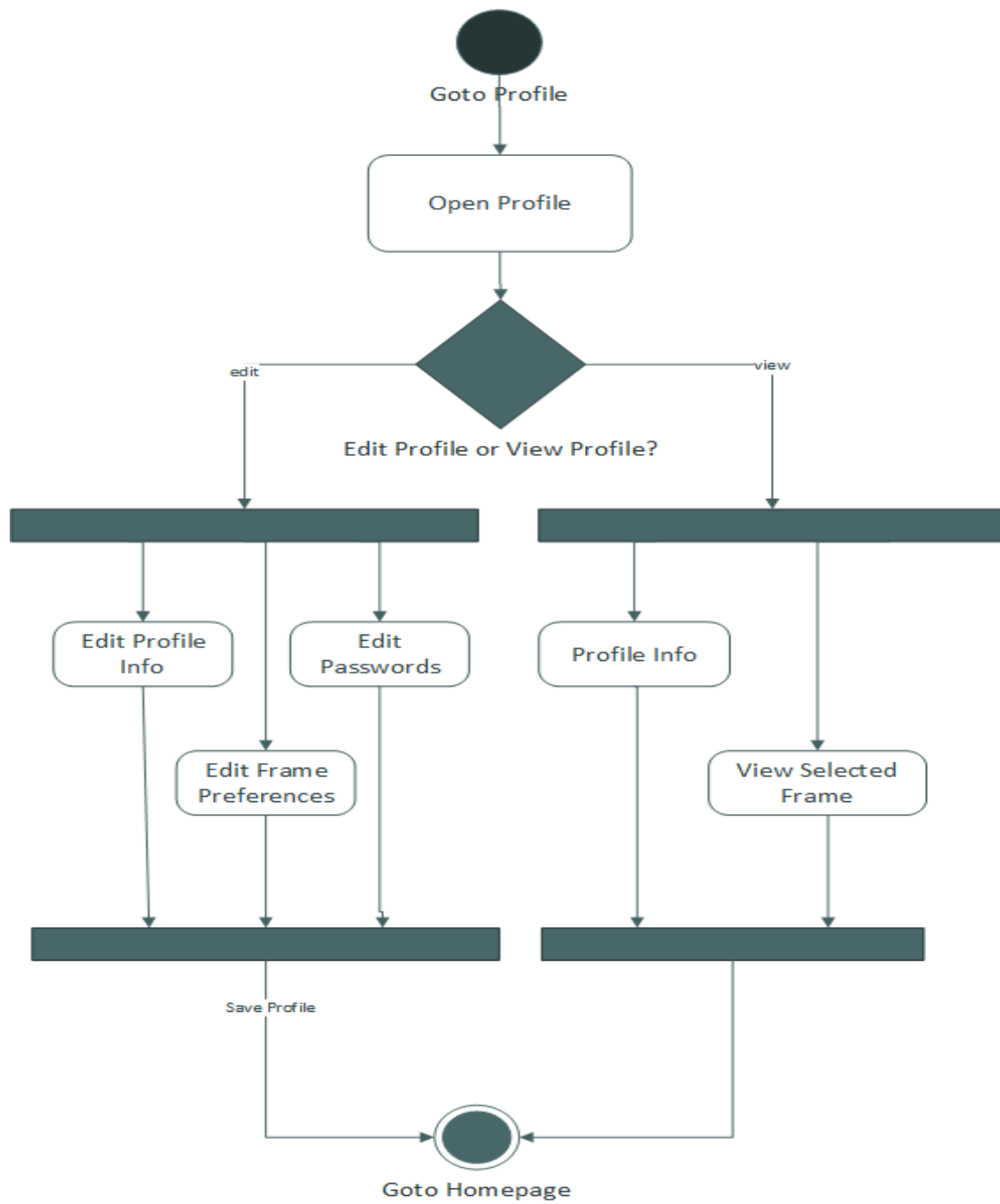### 3.2.3 View and Edit Profile for Customer

Figure 4- View and Edit Profile for Customer

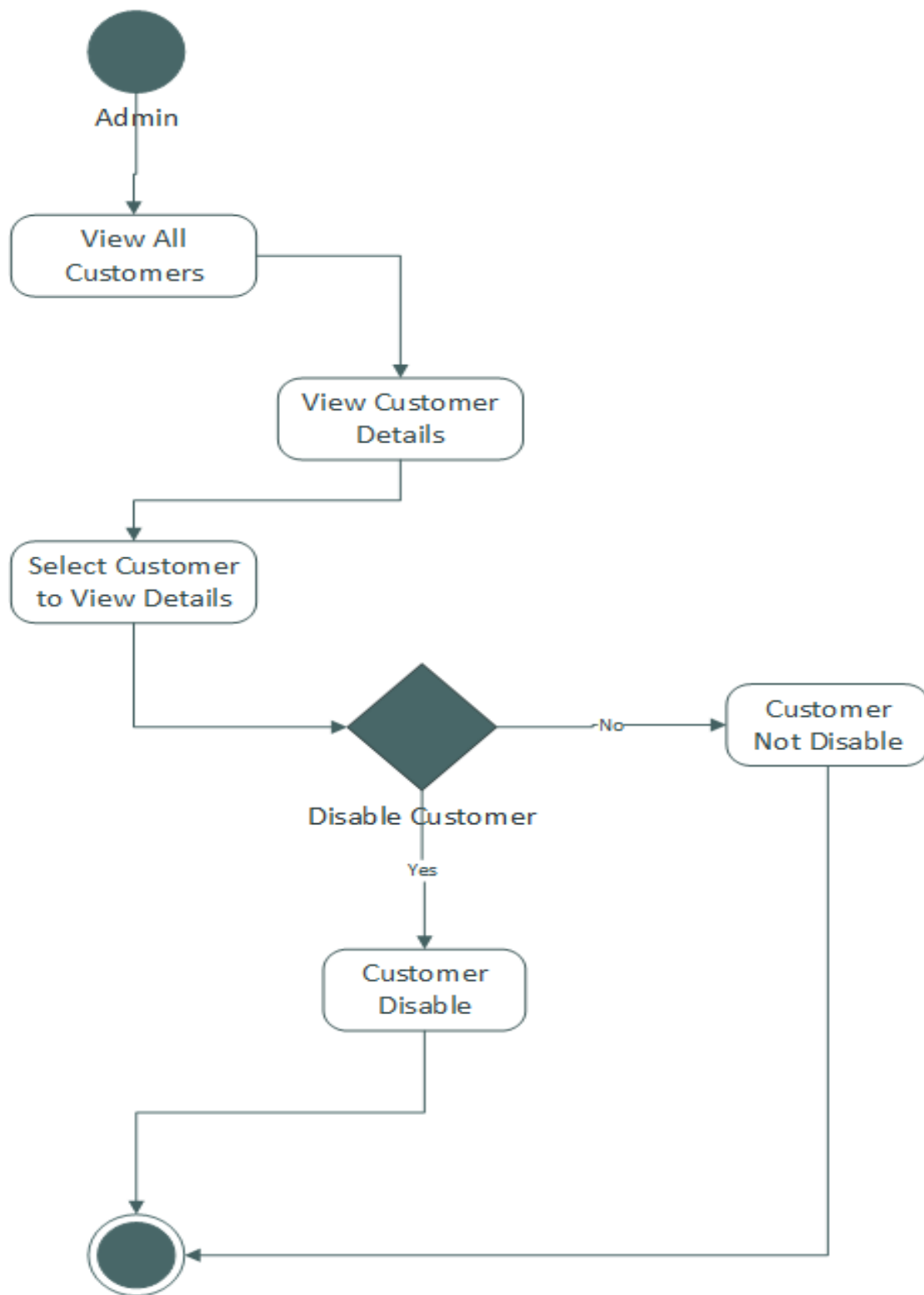### 3.2.4 Manage Customers by Admin

Figure 5- Manage Customers by Admin

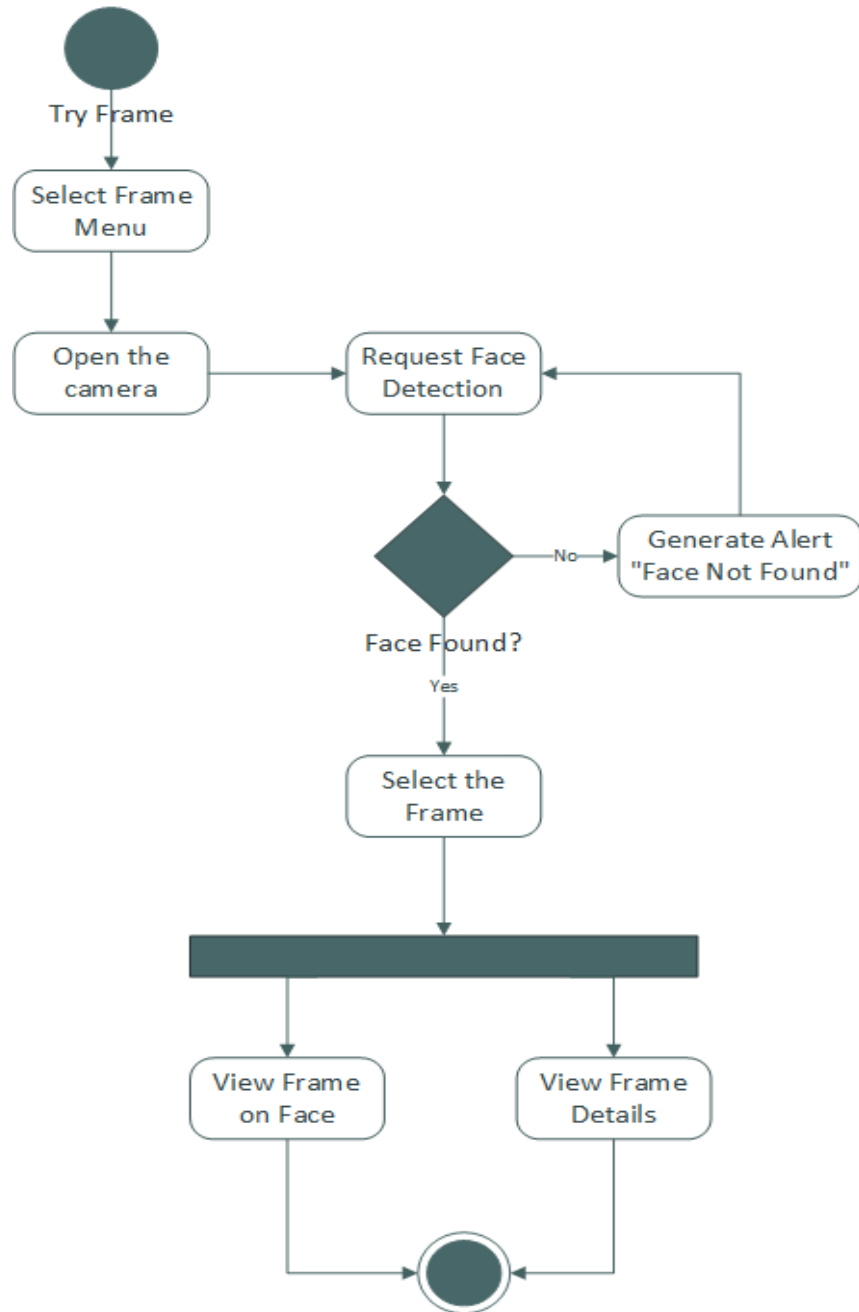### 3.2.5 Frame Experience by Customer

Figure 6- Frame Experience by Customer
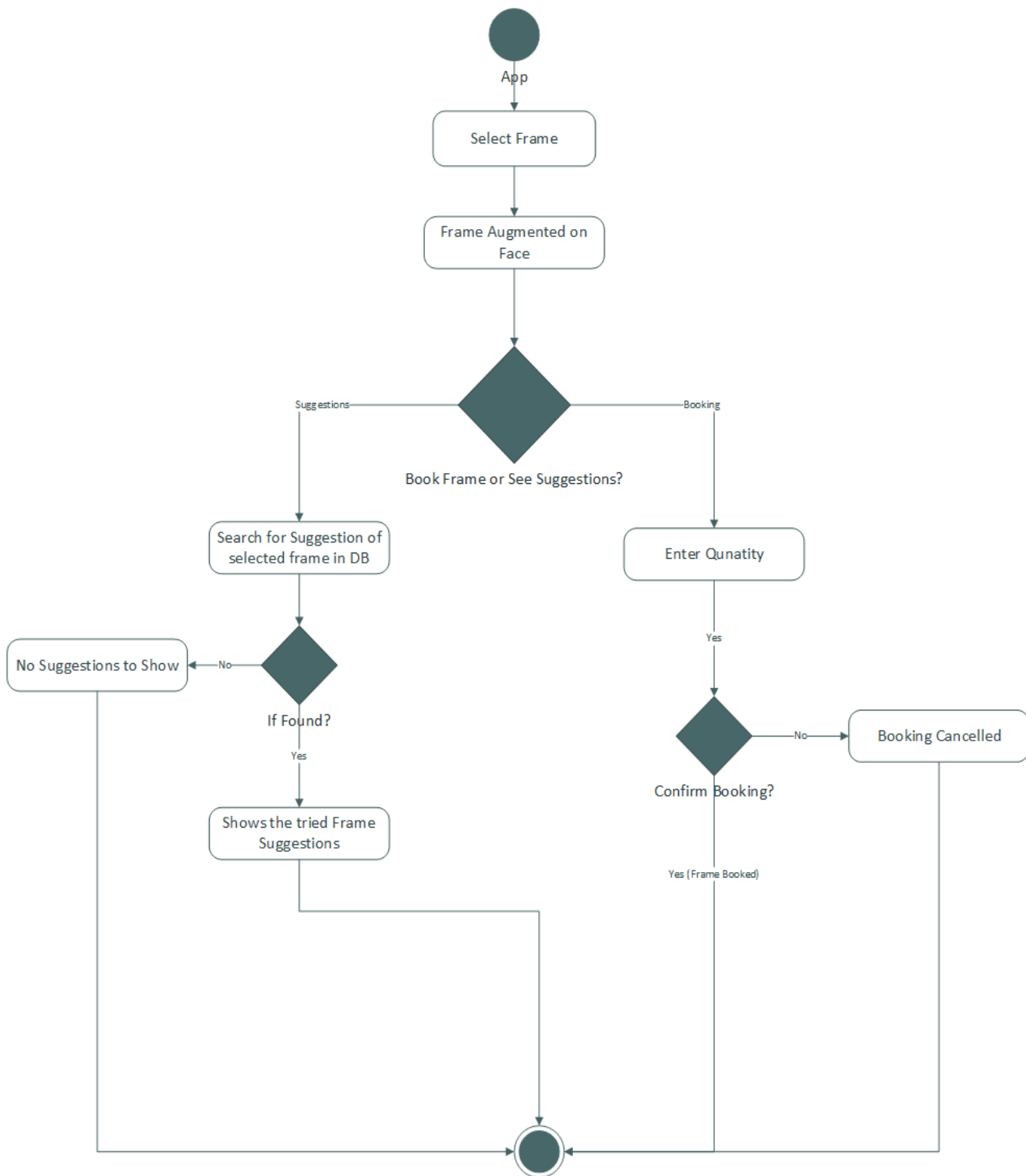
3.2.6 Frame Booking and Suggestions

Figure 7- Share Experience Frame Booking and Suggestions
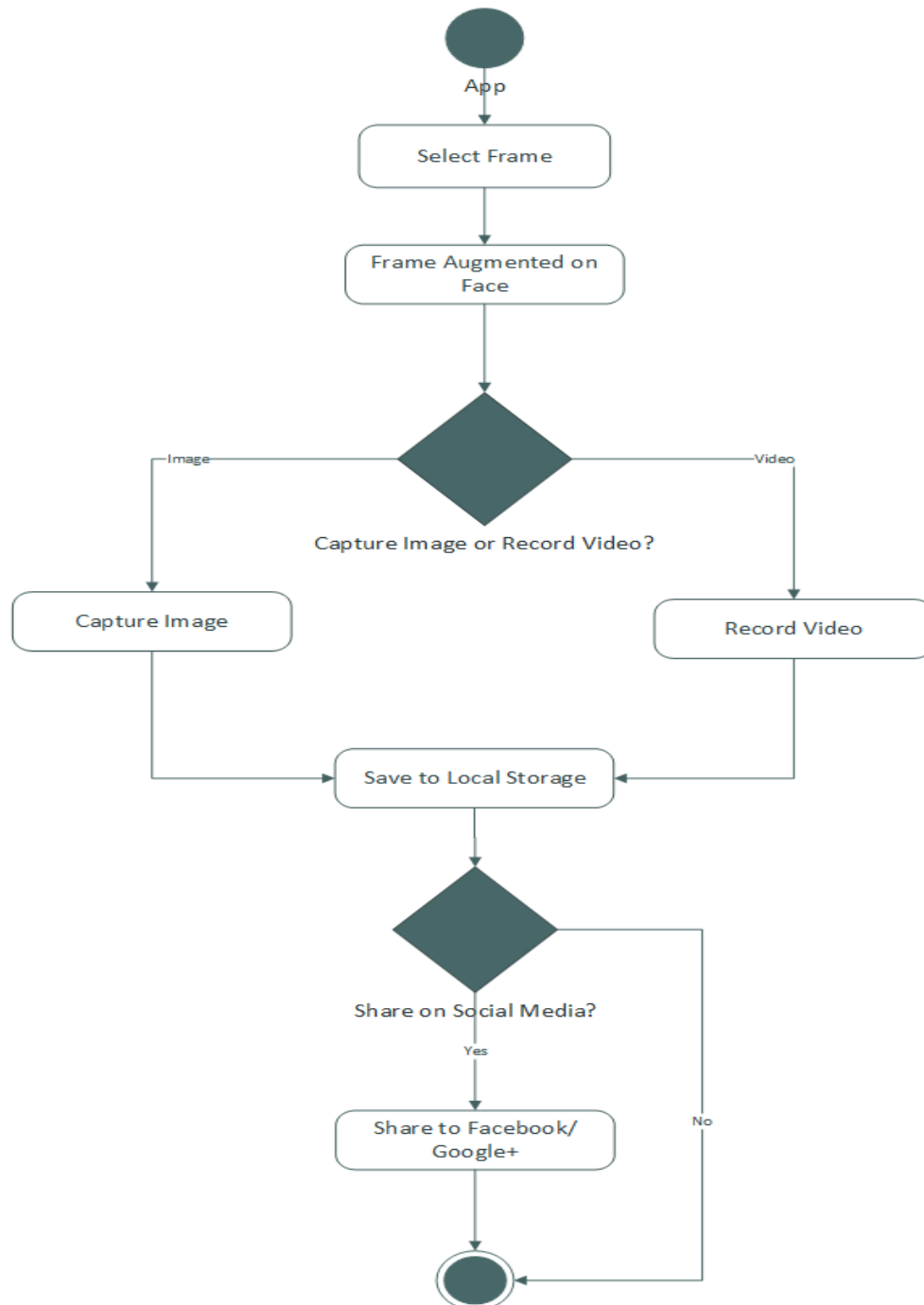
3.2.7 Share Experience on Social Media

Figure 8- Share Experience on Social Media

4. Design models

The design model is an object model describing the realization of use cases, and serves as an abstraction of the implementation model and its source code. The design model is used as essential input to activities in implementation and test.

4.1 Class Diagram

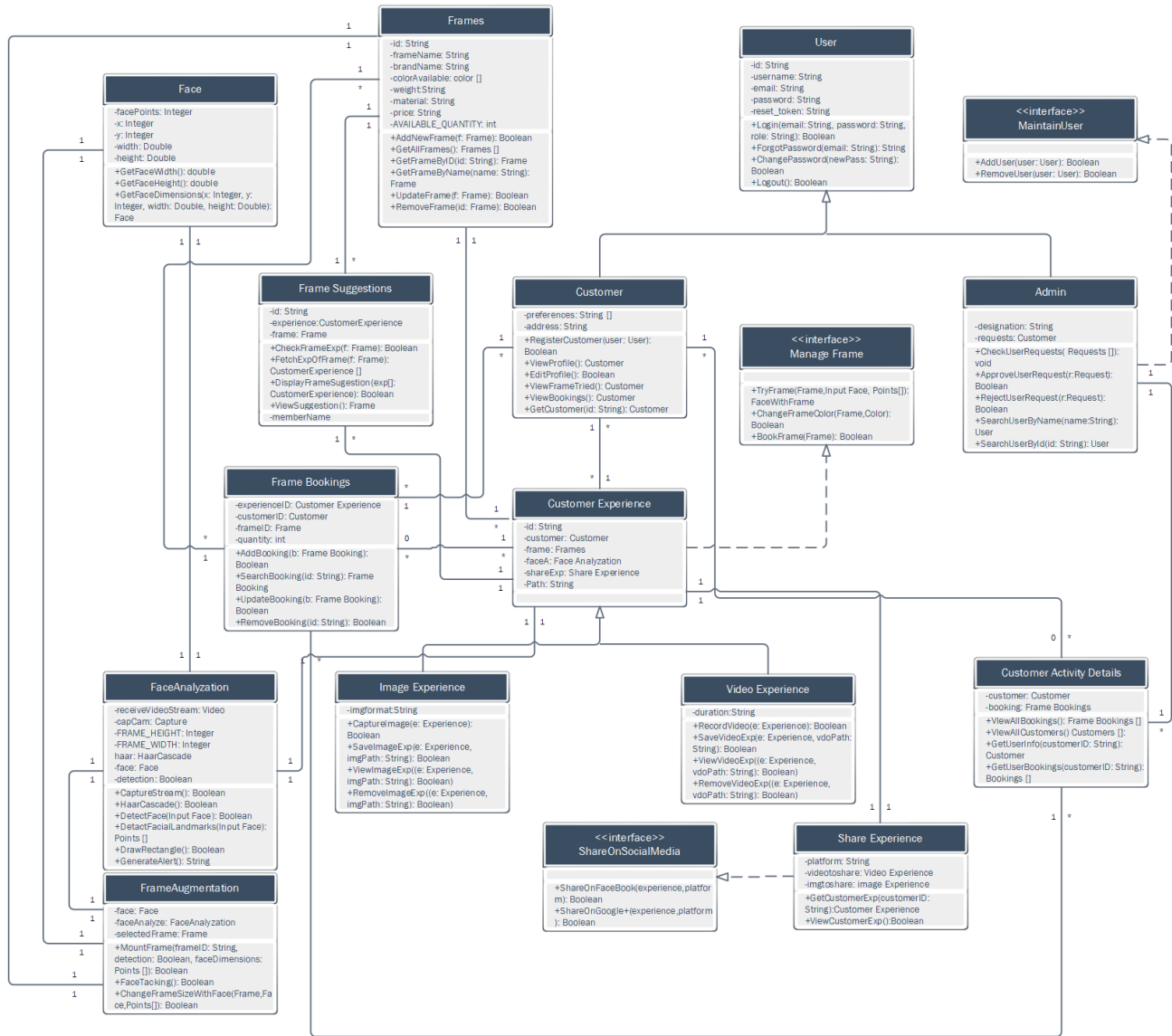Following is the Class Diagram for Face Fit:

Figure 9-Class Diagram

4.2 Sequence Diagrams
Following are the Sequence Diagrams for Face Fit:

4.2.1 Sequence Diagram for Face Detection

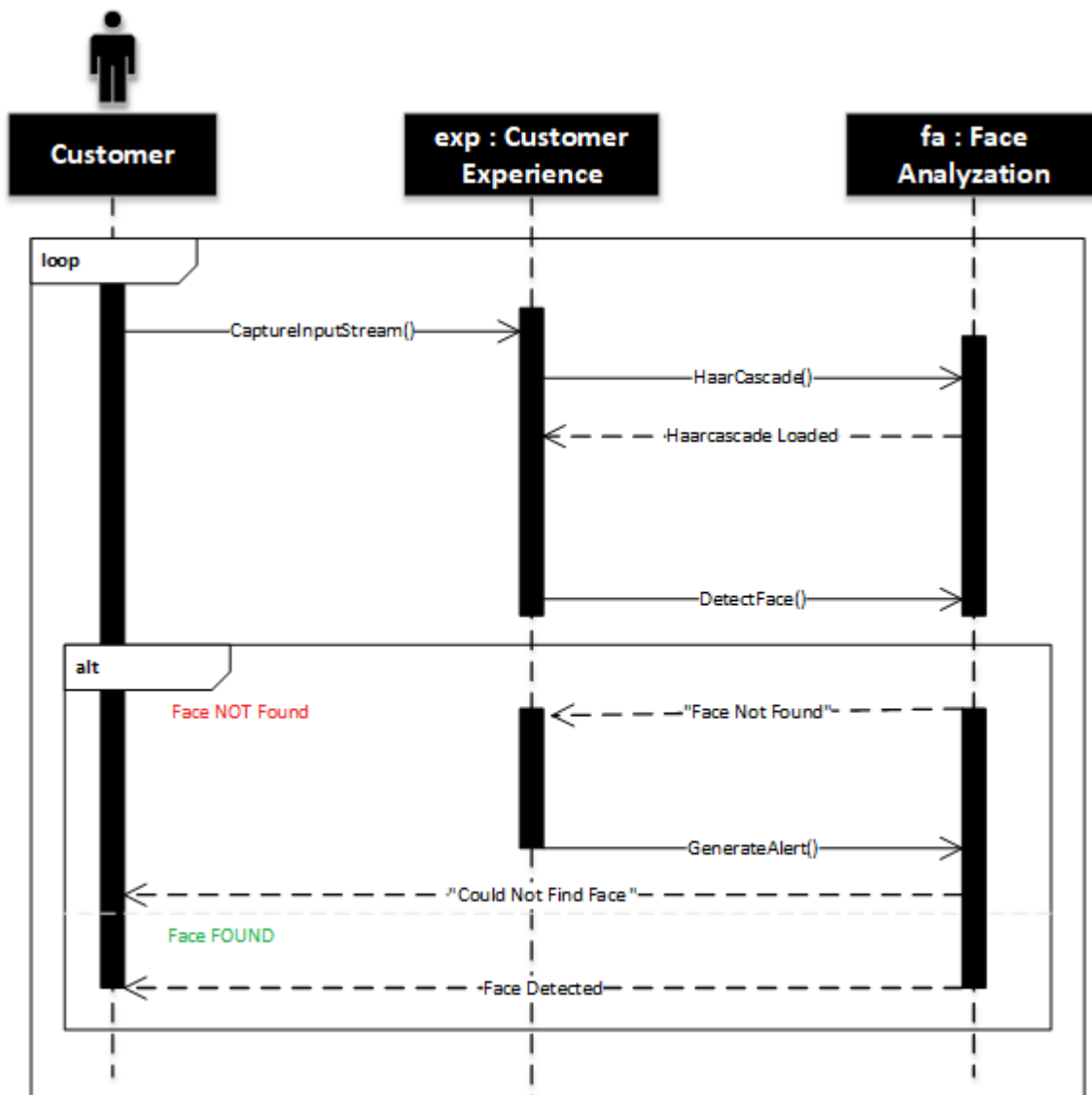**Figure 10-Sequence Diagram for Face Detection**

4.2.2 Sequence Diagram for Facial Landmark

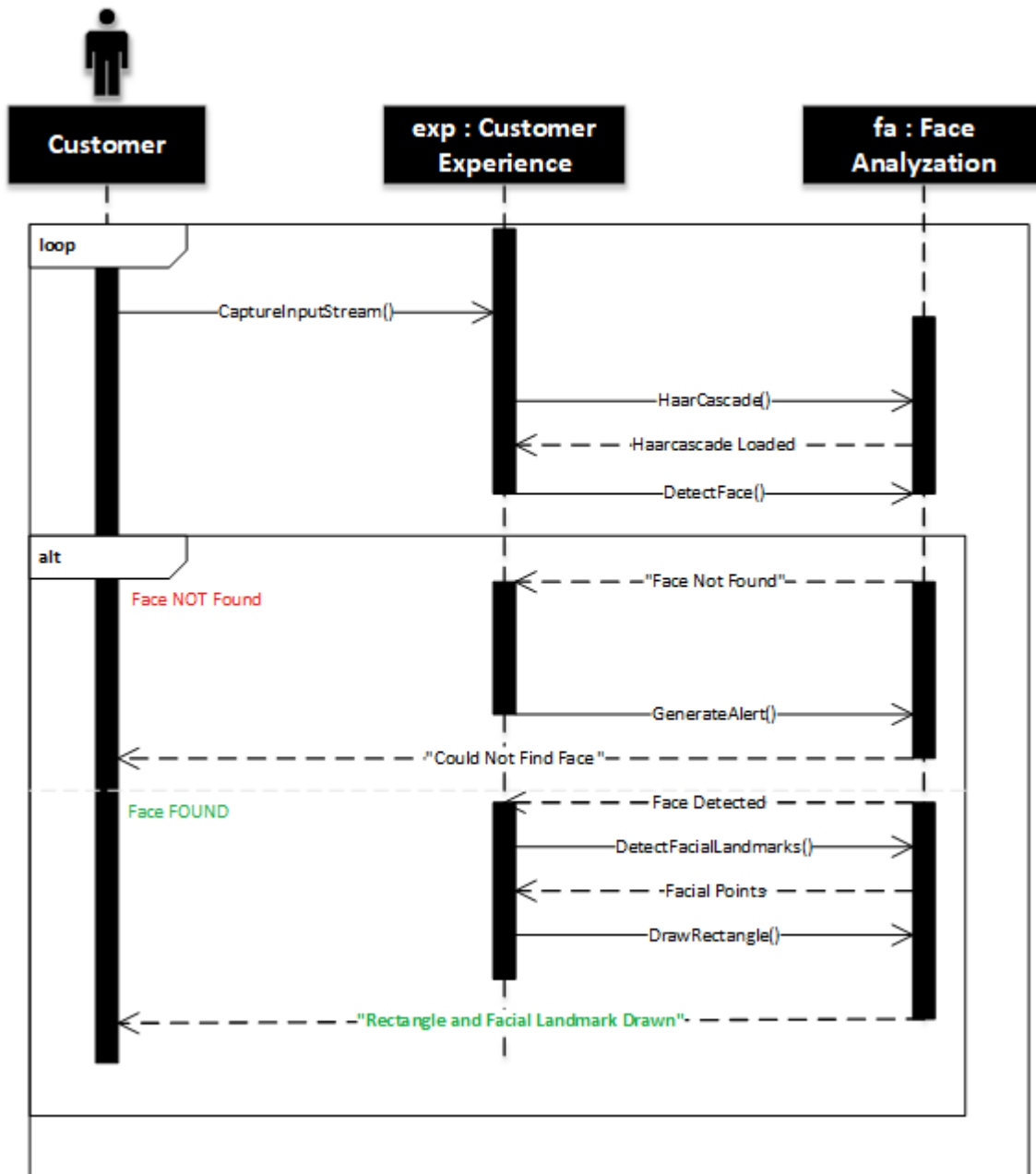**Figure 11-Sequence Diagram for Facial Landmark**

4.2.3 Sequence Diagram for Frame Augmentation on Face
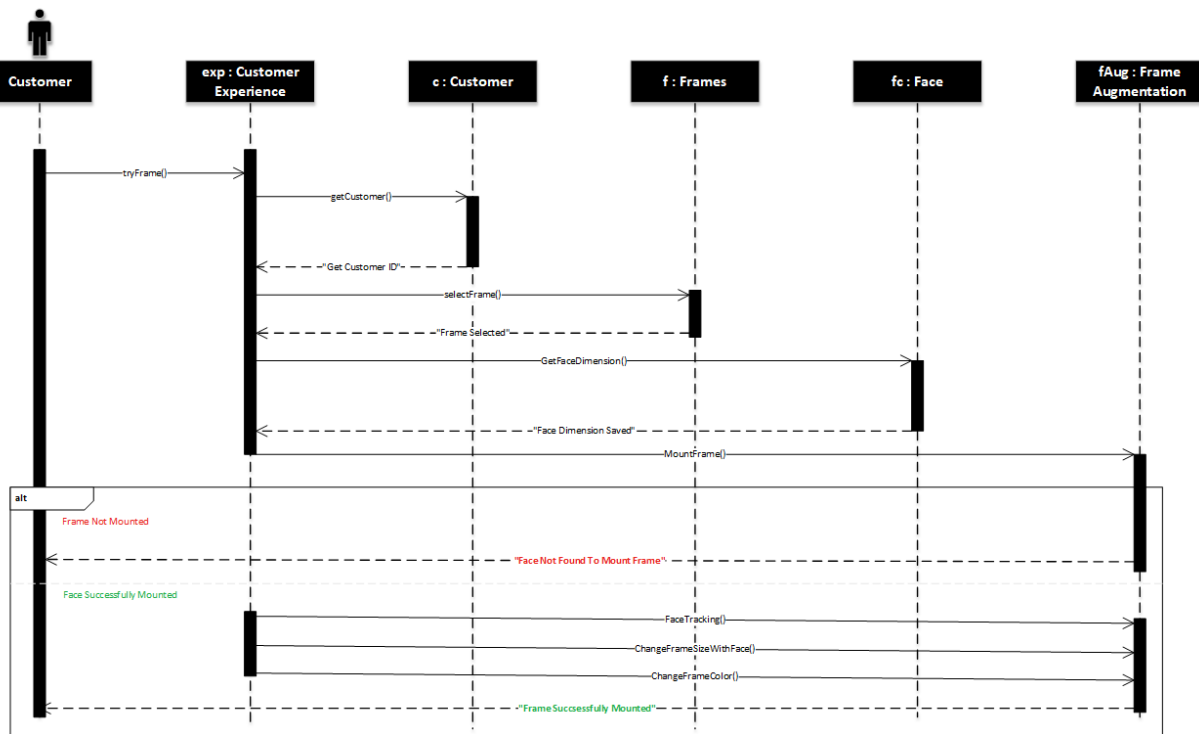
Figure 12-Sequence Diagram for Frame Augmentation on Face

4.2.4 Sequence Diagram for Frames Suggestions of Selected Frame
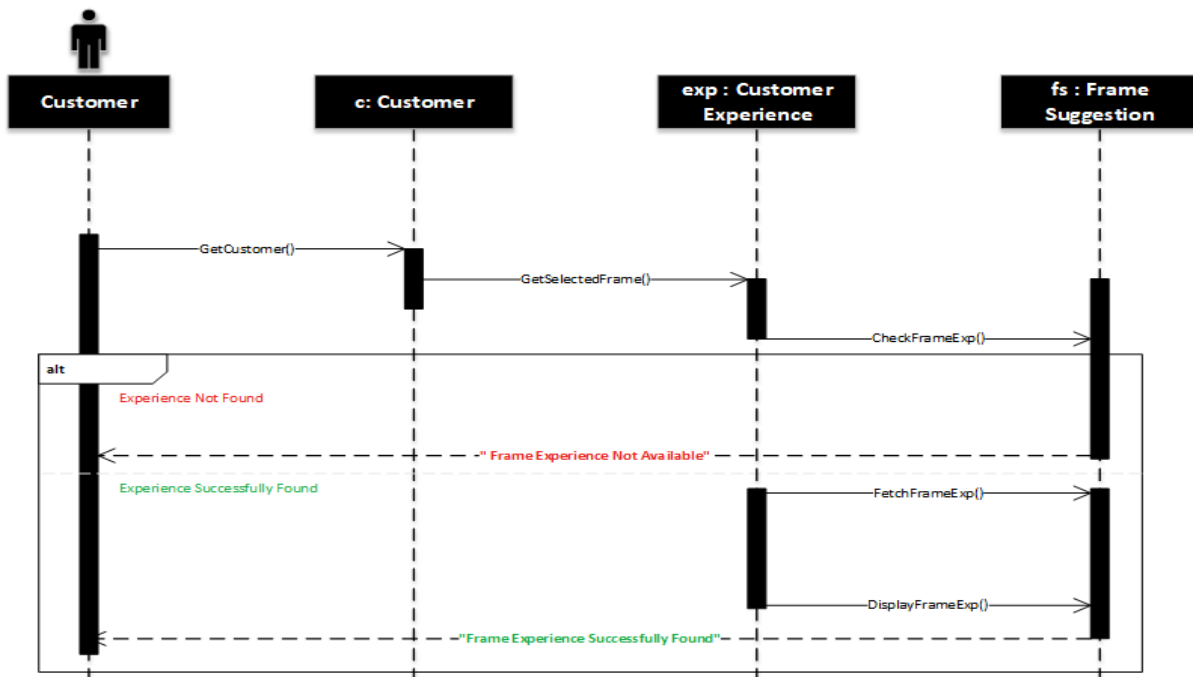
Figure 13-Sequence Diagram for Frames Suggestions of Selected Frame

4.2.5 Sequence Diagram for Frame Bookings

Figure 14-Sequence Diagram for Frame Bookings

4.2.6 Sequence Diagram for Sharing Frame Experience

Figure 15-Sequence Diagram for Sharing Frame Experience

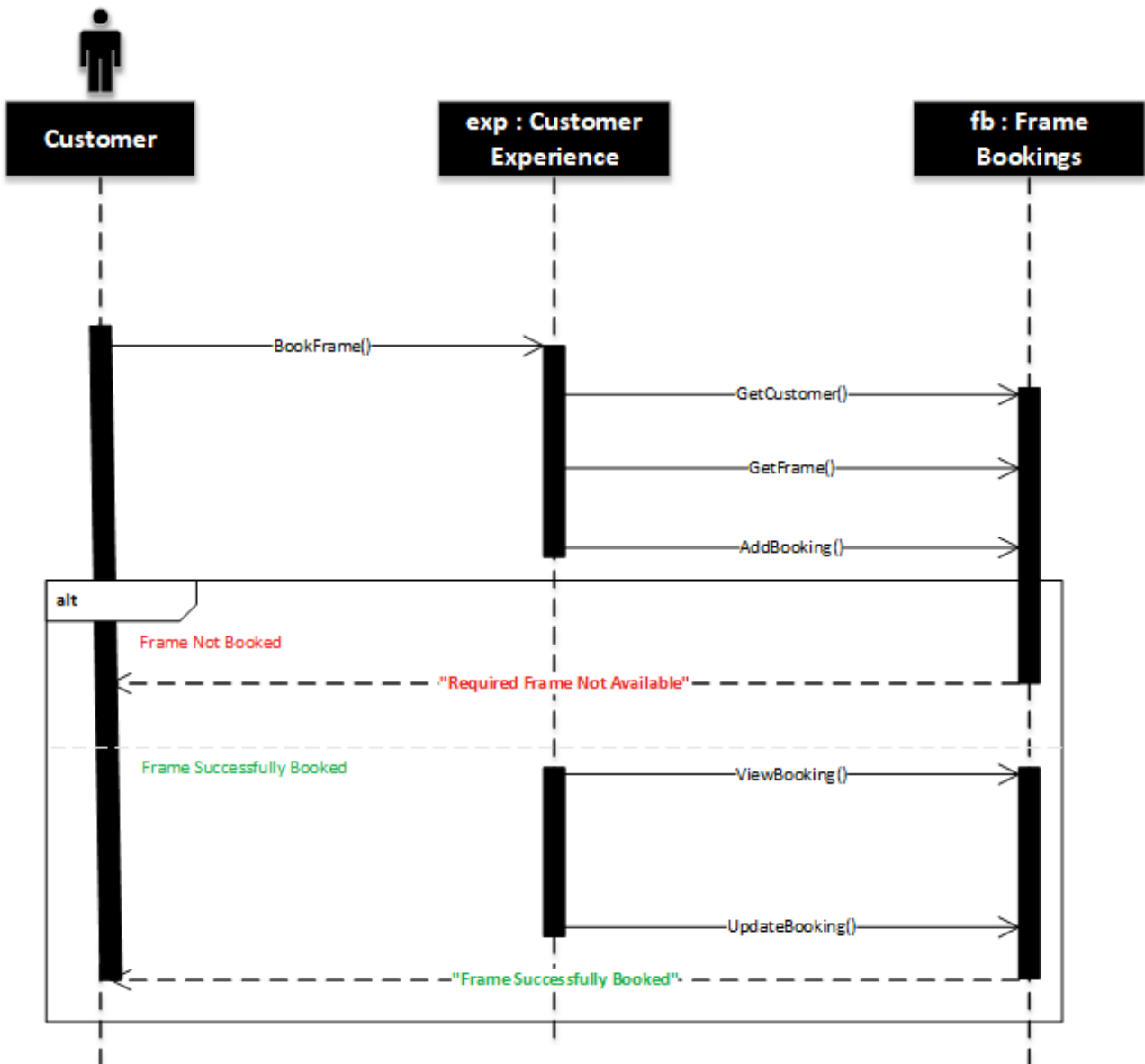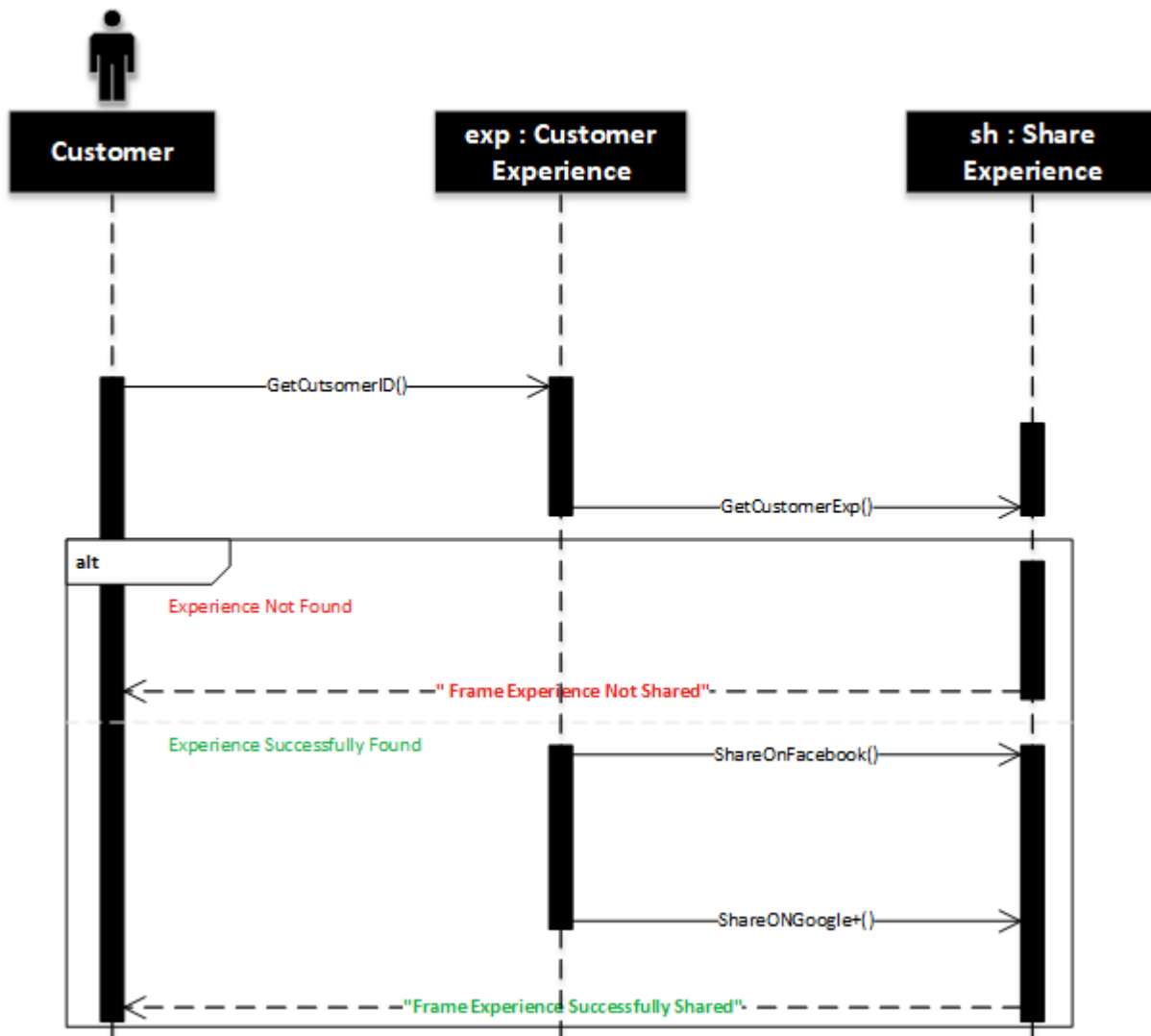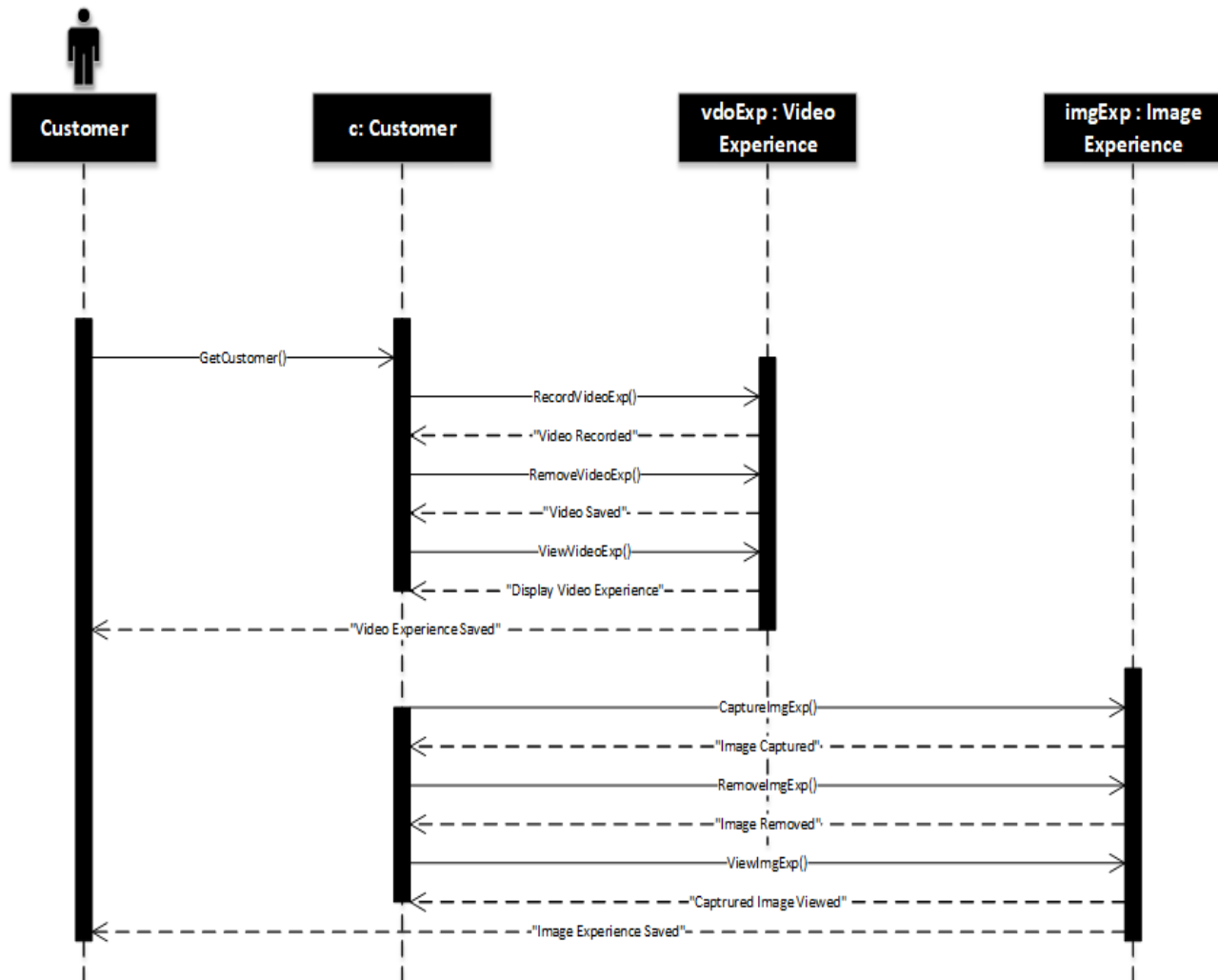### 4.2.7 Sequence Diagram for Recoding/Capture Experience

Figure 16-Sequence Diagram for Recoding/Capture Experience

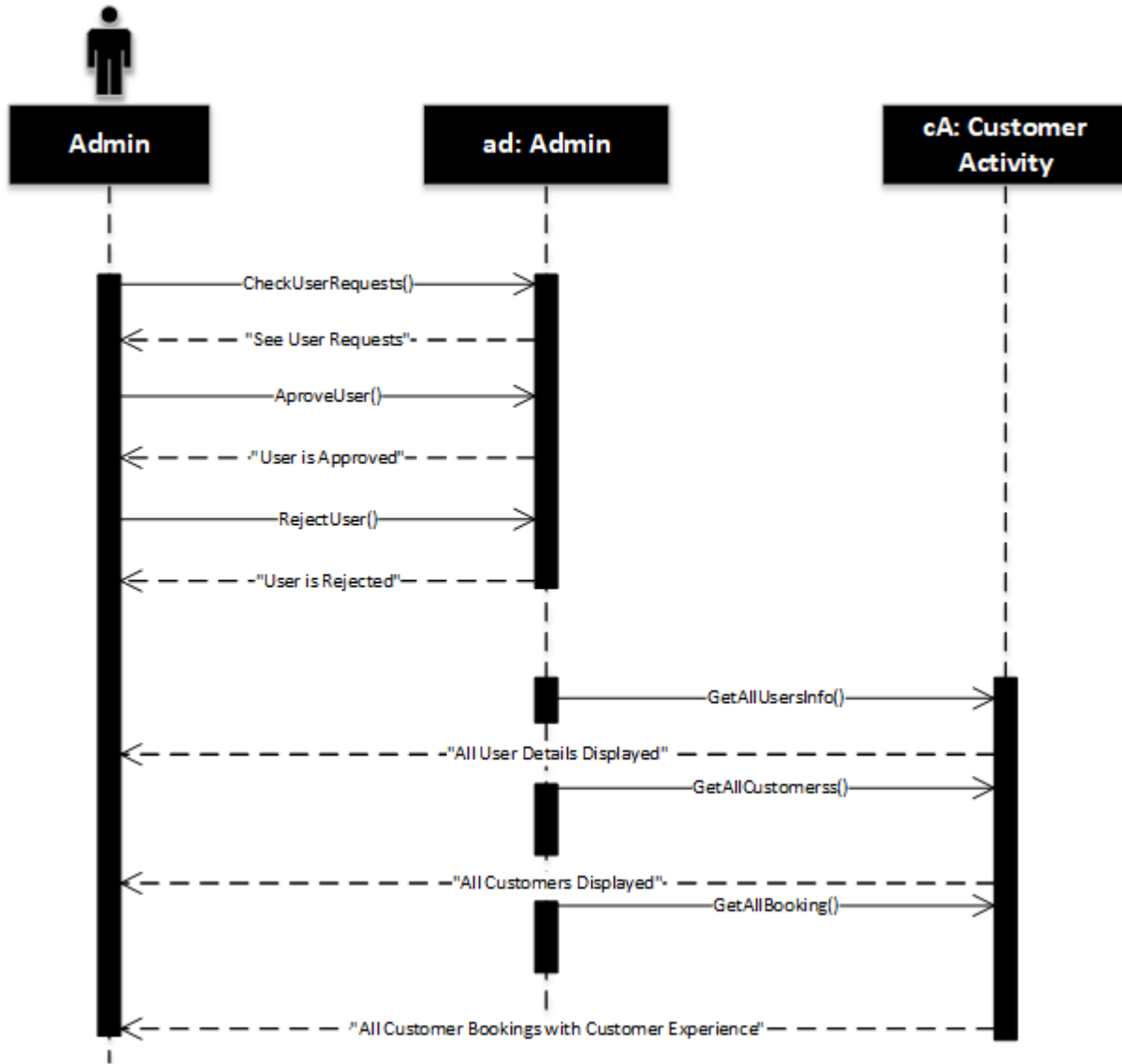## 4.2.8 Sequence Diagram for Admin to Manage Customer

Figure 17-Sequence Diagram for Admin to Manage Customer

5. Data design

The persistent data for "**Face-Fit**" is stored in a NoSQL database called Firebase. The Firebase Real-time Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. Amazon S3 service is used which stores the frame models and is called via API in our applications' backend.

Following will show the collection of the system along with the attribute types.

JSON Schema:

```
{
  "Admin" : {
   "designation" : "String",
   "requests" : "Customer"
  },
  "Customer" : {
   "address" : "String",
   "preferences" : "String"
  },
  "Customer Activity Details" : {
   "cusId" : {
    "booking" : "Frame Bookings",
    "cusId" : "Customer"
   }
  },
  "Customer Experience" : {
   "expid" : {
    "customer" : "Customer",
    "expid" : "String",
    "faceA" : "Face Analyzation",
    "frame" : "Frames",
    "savePath" : "String",
    "shareExp" : "ShareExperience"
   }
  },
  "Frame bookings" : {
   "id" : {
    "cusID" : "Customer",
    "expId" : "Customer Experience",
    "frameID" : "Frames",
    "id" : "String",
    "quantity" : "String"
   }
  },
  "Frames" : {
   "id" : {
    "Quantity" : "String",
    "brandName" : "String",
    "colorAvailable" : "String",
    "frameName" : "String",
    "id" : "String",
    "material" : "String",
```

```
   "price" : "String",
   "weight" : "String"
  }
 },
 "Image Experience" : {
  "type" : "String"
 },
 "Share Experience" : {
  "expid" : {
   "expid" : "Customer Experience",
   "imgExp" : "Image Experience",
   "platform" : "String",
   "vdoExp" : "Video Experience"
  }
 },
 "User" : {
  "id" : {
   "email" : "String",
   "id" : "String",
   "password" : "String",
   "reset_token" : "String",
   "username" : "String"
  }
 },
 "Video Experience" : {
  "duration" : "String"
  }
 }
```

**Table 1-Firebase Data Design**

5.1 Data dictionary

Following is the data dictionary of system along with the attributes interactions and use in the functions of the Class.

| Class	Object	Attributes	Functions/Methods
User	user	id, username, password, reset_token	Login() ForgotPassword() ChangePassword() Logout()
Customer	customer	id, username, password, preferences, address	RegisterCustomer() ViewProfile() EditProfile()
Admin	admin	designation, requests(Customer)	CheckUserRequests() ApproveUser() RemoveUser()

|			
Frames	frames	id, frameName, brandName, colorAvailable, weight, material, price, quantity	AddNewFrame() ViewFrame() UpdateFrame() RemoveFrame()
Face	face	facePoints, x, y, width, height	GetFaceDimensions() GetFaceHeight() GetFaceWidth()
Face Analyzation	faceAn	revieveVideoStream, haarcascade, face(Face), detection(Bool)	CaptureStream() Haarcascade() DetectFace() DetectFaceLandmark() DrawRectangle() GenerateAlert()
Frame Augmentation	fmAug	face(Face), faceAn(Face Analyzation), selectFrame(Frame)	MountFrame() FaceTracking() ChangeFrameSizeWithFace()
Customer Experience	cusExp	id, customer(Customer), frame(Frames), faceAn(Face Analyzation), path	TryFrame() BookFrame() ChnageFrameColor()
Image/Video Experience	cusExp	id, customer(Customer), frame(Frames), faceAn(Face Analyzation), path	CaptureImage() SaveImage() ViewImage() RecordVideo() SaveVideo() ViewVideo()
Share Experience	imgExp, vdoExp	platform, cusExp(Customer Experience)	GetCustomerExp() ShareOnFacebook() ShareOnGoogle+()
Frame Bookings	fmBook	expID(Customer Experience), cusID(Customer), frameID(Frames),	AddBooking() SearchBooking() UpdateBooking()

8.1 Screen images

The following images show the login and Sign up interface of the Customers mobile application.
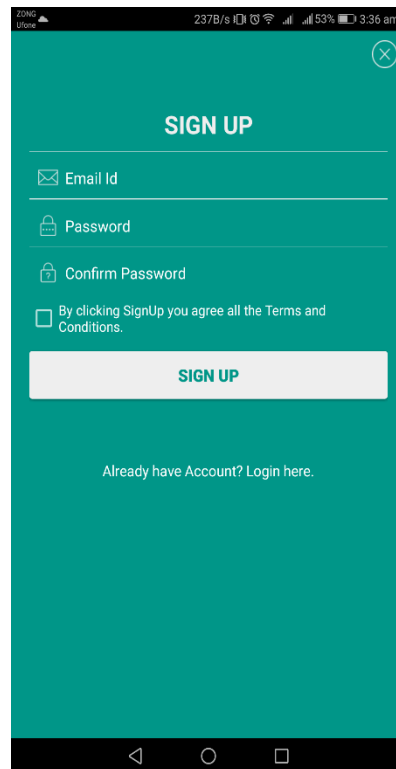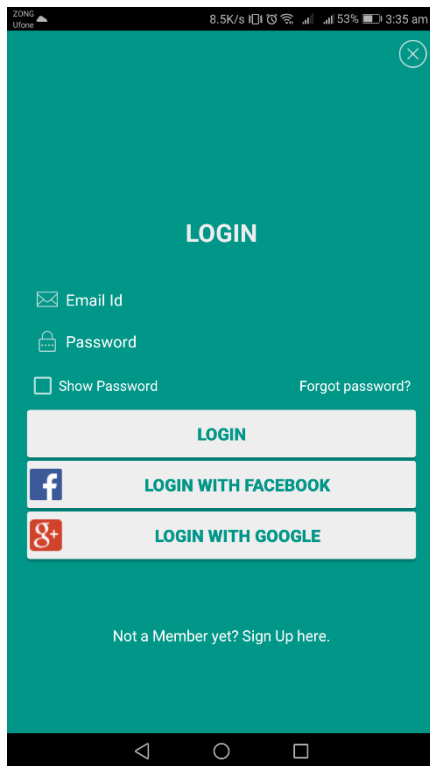
The following images show the forgot password interface of the Customers mobile application
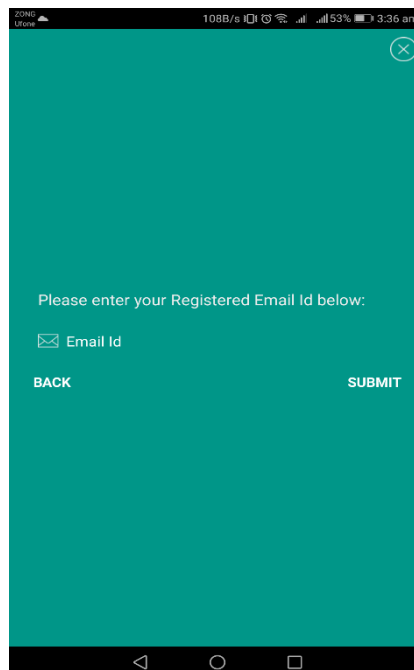
Following is the dashboard of the Application.
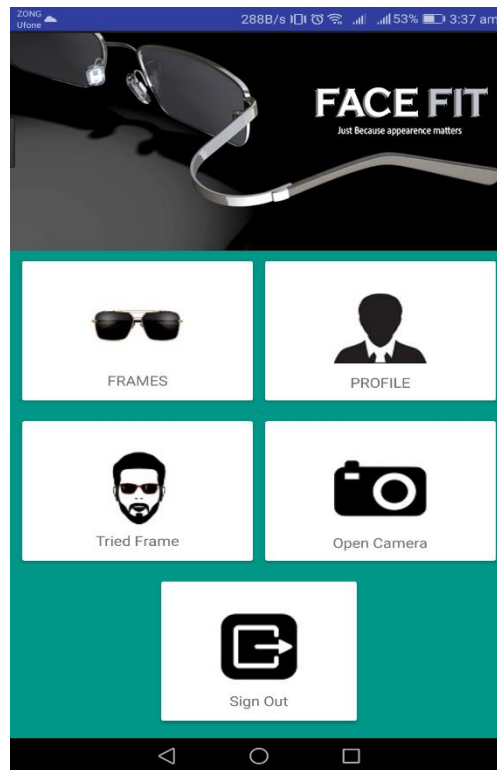
This is Facial Landmark Detection Results.