Detailed Design Report

IYTE CLUBSY

Bora Gültekin - 230201070, Yiğit Demirkapu - 220201005

Contents

1 Introduction 4
1.1 Problem Definition . . . 4
1.2 Purpose . . . 4
1.3 Scope . . . 4
1.4 Definitions and Abbreviations . . . 4
1.5 References . . . 5
1.6 Overview . . . 5

2 System Overview 5

3 Design Considerations 6
3.1 Constraints, Assumptions and Dependencies . . . 6
3.1.1 Assumptions and Dependencies . . . 6
3.1.2 Constraints . . . 7
3.2 Design Goals and Guidelines . . . 7
3.2.1 Usability . . . 7
3.2.2 Portability . . . 7
3.2.3 Extensibility . . . 7
3.2.4 Reliability . . . 7

4 Data 8
4.1 Design and Descriptions . . . 8
4.1.1 Club . . . 8
4.1.2 User . . . 9
4.1.3 ClubAction . . . 10
4.2 Data Dictionary . . . 11
4.2.1 Club . . . 11
4.2.2 User . . . 11
4.2.3 ClubAction . . . 11

5 User Interface Design 12
5.1 Overview . . . 12
5.2 Screenshots . . . 12

6 Detailed Design 17
6.1 Login Activity . . . 17
6.1.1 Classification . . . 17
6.1.2 Definition and Responsibilities . . . 17
6.1.3 Constraints . . . 17
6.1.4 Interactions . . . 17
6.1.5 Processing . . . 17
6.1.6 Exports . . . 17
6.2 Register Activity . . . 18
6.2.1 Classification . . . 18
6.2.2 Definition and Responsibilities . . . 18
6.2.3 Constraints . . . 18
6.2.4 Interactions . . . 18
6.2.5 Processing . . . 19

6.2.6 Exports . . . 19
6.3 Reset Password Activity . . . 19
6.3.1 Classification . . . 19
6.3.2 Definition and Responsibilities . . . 19
6.3.3 Constraints . . . 19
6.3.4 Interactions . . . 19
6.3.5 Processing . . . 20
6.3.6 Exports . . . 20
6.4 Club List Activity . . . 20
6.4.1 Classification . . . 20
6.4.2 Definition and Responsibilities . . . 20
6.4.3 Interactions . . . 21
6.4.4 Processing . . . 21
6.4.5 Exports . . . 21
6.5 My Clubs Activity . . . 21
6.6 Profile Activity . . . 22
6.6.1 Classification . . . 22
6.6.2 Definition and Responsibilities . . . 22
6.6.3 Constraints . . . 22
6.6.4 Interactions . . . 22
6.6.5 Processing . . . 23
6.6.6 Exports . . . 23
6.7 Club Details Activity . . . 23
6.7.1 Classification . . . 23
6.7.2 Definition and Responsibilities . . . 23
6.7.3 Interactions . . . 23
6.7.4 Processing . . . 24
6.7.5 Exports . . . 24
6.8 Create Club Action Activity . . . 25
6.8.1 Classification . . . 25
6.8.2 Definition and Responsibilities . . . 25
6.8.3 Interactions . . . 25
6.8.4 Processing . . . 25
6.8.5 Exports . . . 25
6.9 Main Activity . . . 26
6.9.1 Classification . . . 26
6.9.2 Definition and Responsibilities . . . 26
6.9.3 Interactions . . . 26
6.9.4 Processing . . . 26
6.9.5 Exports . . . 27
6.10 Time Picker Fragment . . . 27
6.10.1 Classification . . . 27
6.10.2 Definition and Responsibilities . . . 27
6.10.3 Interactions . . . 27
6.10.4 Processing . . . 27
6.10.5 Exports . . . 28
6.11 Date Picker Fragment . . . 28
6.11.1 Classification . . . 28
6.11.2 Definition and Responsibilities . . . 28

6.11.3 Interactions . . . 28
6.11.4 Processing . . . 28
6.11.5 Exports . . . 29

7 Technologies and Tools **30**
7.1 Android Studio . . . 30
7.2 Genymotion . . . 30
7.3 Google Firebase . . . 30

8 Conclusion **30**

1 Introduction

This document provides a full explanation of IYTE Clubsy application which is the final design project of IYTE Computer Engineering Department Software Engineering course, including all design issues.

For the final revision of the project we have omitted some features and instead added new features that we decided they would better be suited for this kind of application and the minimalistic, simple design approach which we aim for.

1.1 Problem Definition

In our institution, social clubs currently rely on e-mail groups or mobile instant messaging applications to exchange information and to plan events. This poses a reliability issue, as these methods are not specifically designed for this purpose and are susceptible to miscommunication. Furthermore, using more than one communication medium may cause confusion. Our solution is to design a system aimed to fix these problems and improve upon existing systems by providing an easy to use and simple mobile application designed for this specific purpose which will allow users to interact with their social clubs in a simpler manner.

1.2 Purpose

Main purpose of this document is to define the design details of IYTE Clubsy software project. It gives overall information about the project such as system overview, design considerations, data design, system architecture, user interface design and detailed design.

1.3 Scope

IYTE Clubsy is aimed for students or employees of IYTE that are interested in social clubs and their events. People who are already members of such clubs are the main users. Application is based on Android Platform and therefore will be used on mobile devices.

The application will make the communication between members of the social clubs more reliable and much easier. Announcements will be posted by admin of the club under announcements tab in the main activity screen. Moreover, upcoming events can be created by the admin of the club and can be seen under the events tab in the same main activity screen.

1.4 Definitions and Abbreviations

IYTE: Izmir Yüksek Teknoloji Enstitüsü(Izmir Institute of Technology)
SRS: Software Requirements Specification
API: Application Programming Interface
SDK: Software Development Kit
IDE: Integrated Development Environment
IM: Instant Messaging

1.5 References

1. Sommerville, Ian. *Software Engineering (10th Edition).* Addison-Wesley, 2004.
2. Savage, John and Barclay, Kenneth. *Object-Oriented Design with UML and Java.* Elsevier, 2003.
3. IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

1.6 Overview

This document contains detailed descriptions about the IYTE Clubsy project.

In ”Introduction”, problem, purpose and scope of our project is given. We stated the reason behind the usefulness and necessity of our product, and what makes our product charming. We have also defined some important terms written this report.

In ”System Overview”, the overall design and working mechanism of the project is broadly given.

”Design Considerations” details the design constraints, assumptions, dependencies, goal and guidelines.

”Data” section states the data objects, architectural relations between them and how they are kept in the database. In addition, descriptions of each and every attribute is given in detail. To make things clearer, snapshots of the database is also given.

”User Interface Design” section mainly explains the implementation of the GUI and gives example to each GUI layout related to their respective activities.

”Detailed Design” section explains the components(activities) of the application in detail one by one.

In ”Technologies and Tools”, helpful off-the-shelf products that are used to create and maintain the project are explained.

”Conclusion”

2 System Overview

Our application is intended to work on Android Platform based devices. Authentication, Real-Time Database and Cloud Function are provided by Google Firebase Platform. Firebase API allows us to integrate these systems and allow our application to communicate with the database and other systems.

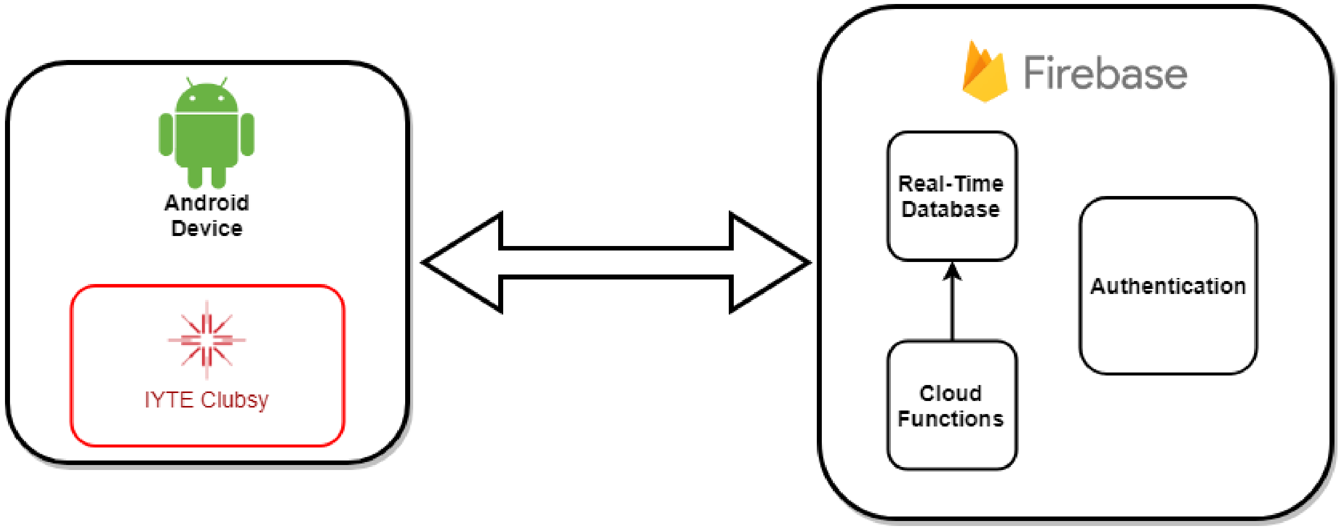


Figure 1: System Overview

Authentication let us keep our users credentials securely and safely on Google's Firebase Servers. Users privacy are maintained by Google, such that even developers can not access users passwords.

Real-Time Database keeps the entire data in JSON format and allows real-time queries to be executed. It also allows us to bind Listeners to data change in the database.

Lastly, Cloud Functions are used to create triggers over the database. A seperate runtime environment that constantly listens to database changes and fires the appropriate function for execution. Used for reliable notification delivery of events and announcements in our application.

3 Design Considerations

In this section, assumptions, dependencies and constraints in the design of this system is explained. Design goals and guidelines are also clarified.

3.1 Constraints, Assumptions and Dependencies

3.1.1 Assumptions and Dependencies

Our application is developed solely for Android devices. Therefore users are assumed to have an Android device.

In terms of software, application requires constant Internet-connection to work. Aside from that, Profile/ID, Notification and High Accuracy Location permissions are needed to be granted for the application in order to work as intended. Google Maps integration requires the said software to be present in the device, along with Google Play services.

3.1.2 Constraints

At the start of the project, team members had zero experience or knowledge about Android programming. In addition they were inexperienced about the Gooogle's Firebase platform, Real-Time Database with JSON Tree structures and TypeScript/Node.js environment for Cloud Functions. All necessary technologies are researched and tested in order to provide a user friendly application.

3.2 Design Goals and Guidelines

We have followed Agile Software Development methodology in our project. The reason for that was we simply didn't have any experience or knowledge about the platform itself. Therefore we have started from the basics and improved upon it part by part as we learned along the way.

Observer Design Pattern was used to have an interactive user interface and to retrieve data in real-time from the database.

We also have used Adapter Design Pattern since it was mandatory to build GUI in Android applications. (e.g. Scroll-able Lists)

3.2.1 Usability

Application was designed with simplicity in mind. Only the most required features were added and implemented. Any feature that did not provide core necessity was omitted. This resulted in a user friendly interface and easy to use, reliable application.

3.2.2 Portability

We have designed our project to work on only Android devices. However, all features and the application itself can be ported to work on iOS devices without any constraints as well.

3.2.3 Extensibility

IYTE Clubsy is bare-bones application. It implements only the most important core features that are needed. Therefore, application can be extended depending on the needs of the users. Among the ideas that first come to mind are in-club chat/IM and User Profile Pictures. All these can be added easily upon request, using currently implemented libraries, services and APIs.

3.2.4 Reliability

Notifications are one of the most important feature for our application, since we aimed to develop a reliable application. Users should be notified of events and announcements without any mistake or delay. Therefore we have used a robust component called Cloud Functions. This component works by itself, without any dependency from outside. It is essentially a server that listens to database changes 24/7 and sends the appropriate notifications to correct devices. Even if the said device were off at that time, when users starts up the device, they still receive past notifications.

4 Data

4.1 Design and Descriptions

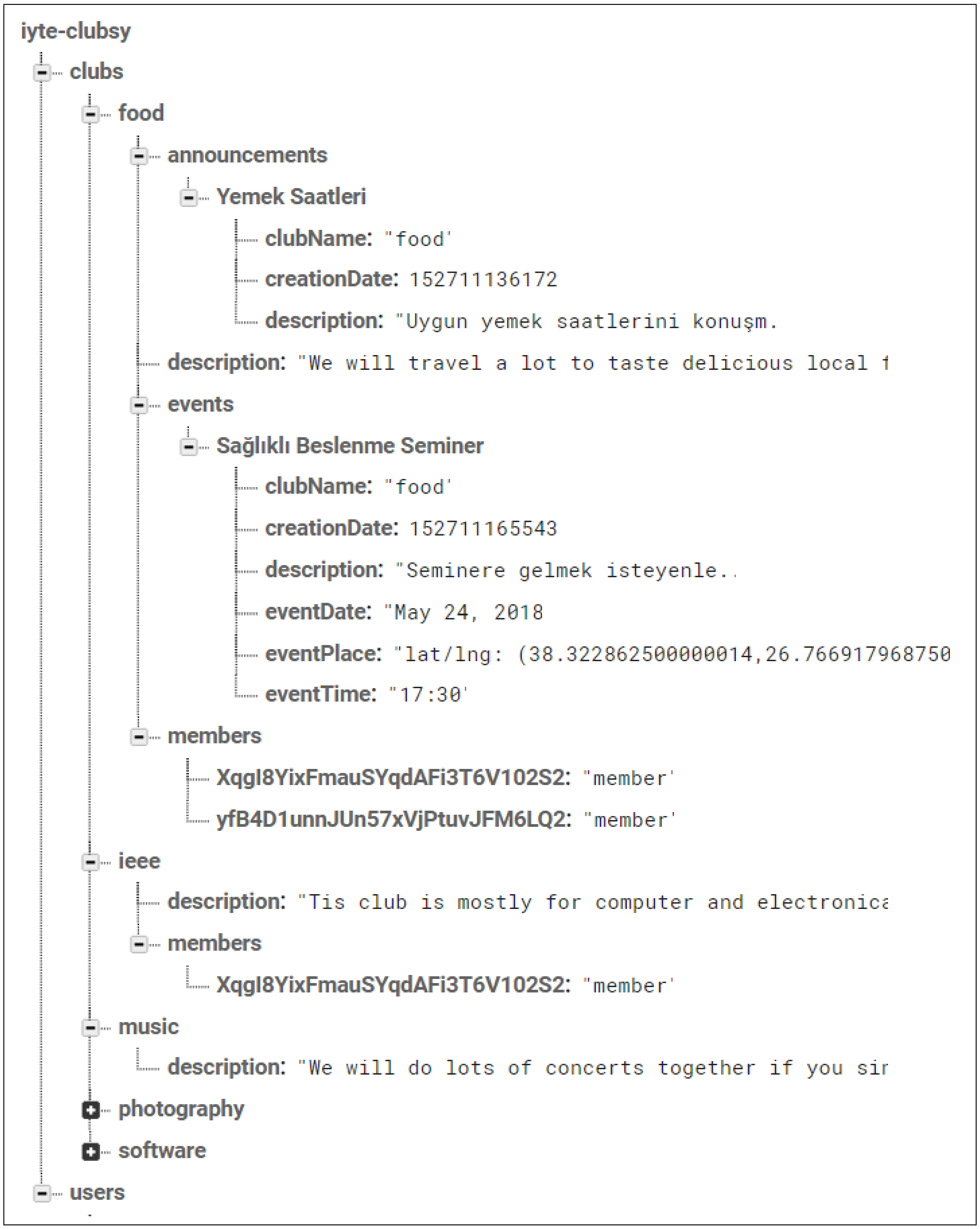
In this project, there are predefined objects such as **User**, **Club** and **ClubAction**. However, due to design constraints of Android Platform and the project itself, these objects are kept inside the database and are not created inside the program, except for **ClubAction**. They are queried and used whenever needed through the use of Real-Time Database.

Data objects and their information are kept in the database of Firebase, named Real Time Database. As stated in the SRS, data is kept in JSON tree format, which keeps everything as Strings. One advantage of using JSON Tree is that it eliminates unwanted redundancy.

Real-Time Database also allows us to use Listener design pattern, and query information whenever there is change in the database, automatically.

4.1.1 Club

Clubs can have **members**, **description**, **announcements** and **events**. Initially there is only description. An example tree can be seen below. Announcements and Events are separate data objects and as such, it will be explained under the ClubAction.

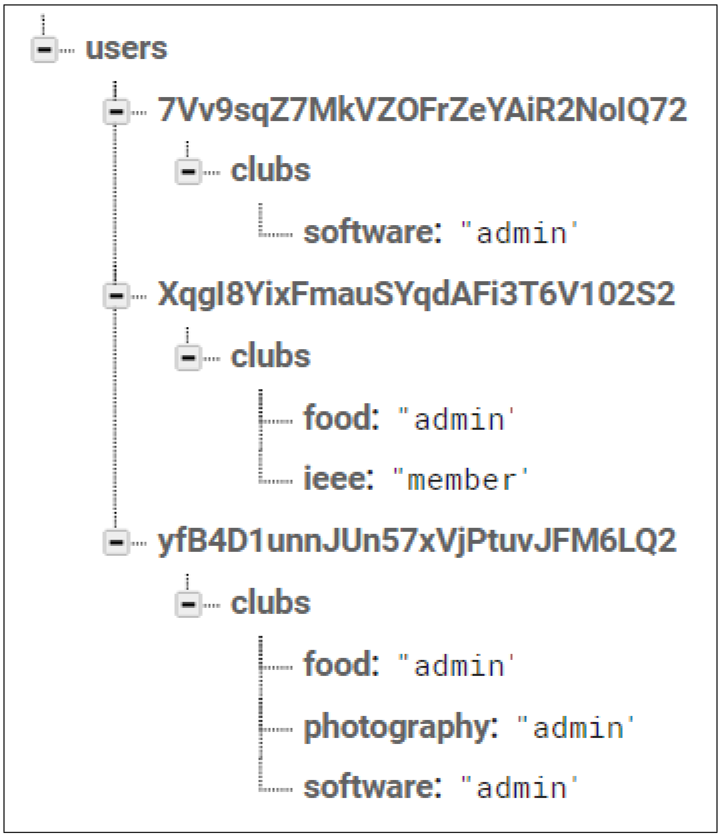


```
iyte-clubsy
  clubs
    food
      announcements
        Yemek Saatleri
          clubName: "food"
          creationDate: 152711136172
          description: "Uygun yemek saatlerini konuşm.
      description: "We will travel a lot to taste delicious local f
      events
        Sağlıklı Beslenme Seminer
          clubName: "food"
          creationDate: 152711165543
          description: "Seminere gelmek isteyenle..
          eventDate: "May 24, 2018
          eventPlace: "lat/lng: (38.322862500000014,26.766917968750
          eventTime: "17:30"
      members
        Xqgl8YixFmauSYqdAFi3T6V102S2: "member"
        yfB4D1unnJUn57xVjPtuvJFM6LQ2: "member"
    ieee
      description: "Tis club is mostly for computer and electronica
      members
        Xqgl8YixFmauSYqdAFi3T6V102S2: "member"
    music
      description: "We will do lots of concerts together if you sir
    photography
    software
  users
```

Figure 2: Clubs JSON Tree

4.1.2 User

Users tree keep the information of users. They are identified by their Firebase Authentication UID values. Every user also has a list of **Clubs** which they have subscribed to. An example tree can be seen below.



```
users
    7Vv9sqZ7MkVZOFrZeYAiR2NoIQ72
        clubs
            software: "admin'
    XqgI8YixFmauSYqdAFi3T6V102S2
        clubs
            food: "admin'
            ieee: "member'
    yfB4D1unnJUn57xVjPtuvJFM6LQ2
        clubs
            food: "admin'
            photography: "admin'
            software: "admin'
```

Figure 3: Users JSON Tree

4.1.3 ClubAction

ClubAction objects refer to Announcements or Events sub-trees inside the **Clubs**.

Announcements can be identified with their Subject and they also have **clubName**, **creationDate** and **description**. **creationDate** is in UNIX Epoch format.

Events have additional fields, namely **eventDate**, **eventPlace**, **eventTime**. **eventPlace** keeps the latitute and logitude so that we can open Google Maps on the desired location. See Figure 2 for examples.

This object also has it's own class inside the program, where it additionaly holds another attribute, called **eventType**, which states this objects type, whether it is Event or Announcement.

4.2 Data Dictionary

4.2.1 Club

Root of the tree is both the ID and the name of the club.
announcements: Announcement type of **clubAction** objects are kept here.
events: Event type of **clubAction** objects are kept here.
description: Describes the club, what is it's focus and area of interest.
members: List of UID's of the subscribed members and admins.

4.2.2 User

Root of the tree is the Firebase Authentication UID of the user.
clubs: List of club names which the user has subscribed to. Their values determines whether the user is admin or regular member.

4.2.3 ClubAction

ClubAction refers to 2 similar types of objects, Announcement and Event. Root of the tree is both the ID and the subject.

Announcements have;
clubName: States the name of the Club which the action is associated with. This is purely for the ease of programming.
description: Description of the action. Main body of the event or announcement.
creationDate: The creation time of the database entry, in UNIX epoch time.

Events additionally have;
eventDate: Start date of the event.
eventTime: Time of occurance of the event in 24h format.
eventPlace: Latitude and Longitude of event location. Used to show the location in Google Maps application.

Also, both objects have an attribute inside the class;
eventType: States the type of the ClubAction in Strings. Appropriate values are "Announcement" and "Event".

5 User Interface Design

5.1 Overview

In Android applications, GUI is defined as layout files which are in XML format. Every activity states and uses one of the layout files. Whenever an activity is brought up, it's layout file is accessed and shown on the screen. Buttons, click-able objects and editable objects are tied to the related activity class through the use of Listener objects. Every layout file can be seen in the next section.

5.2 Screenshots

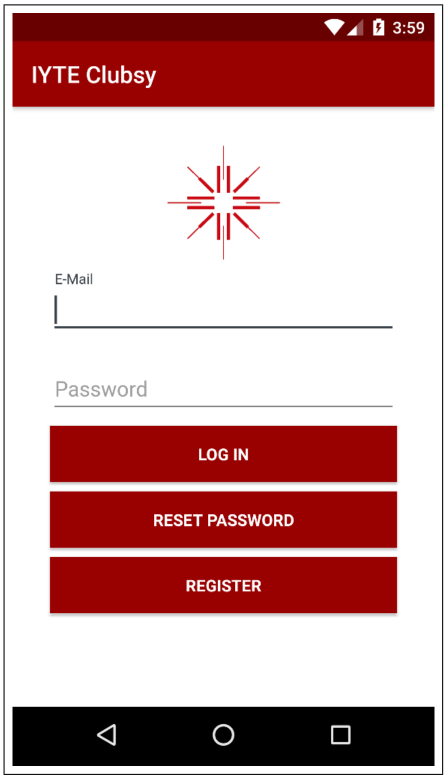


Figure 4: Login Activity

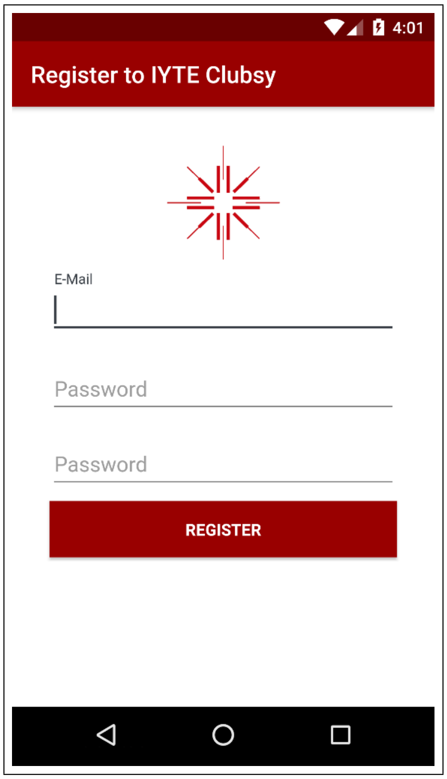


Figure 5: Register Activity

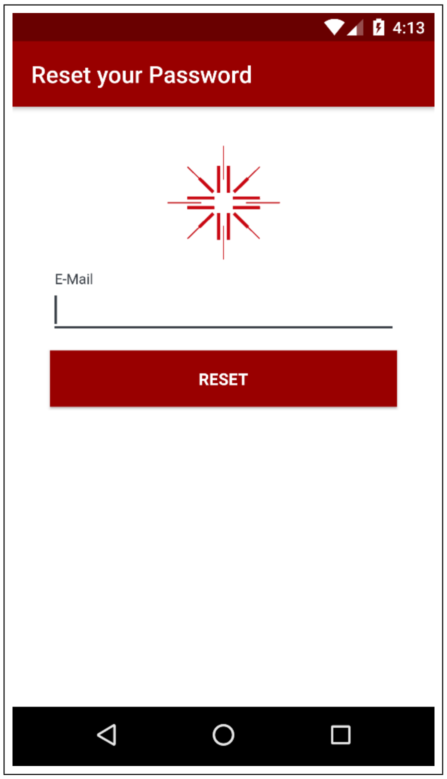
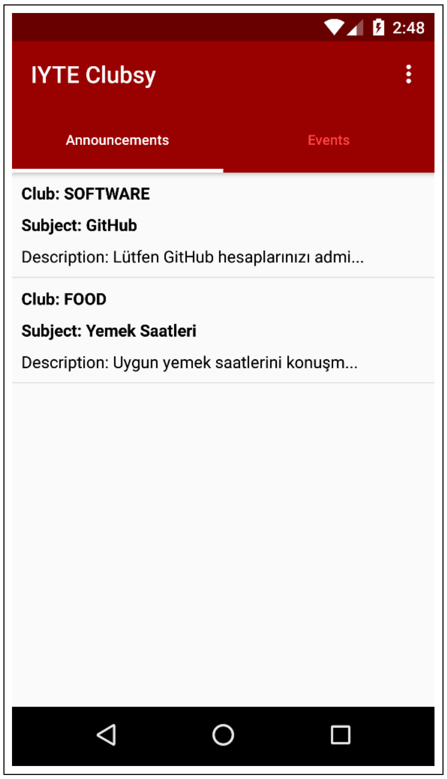


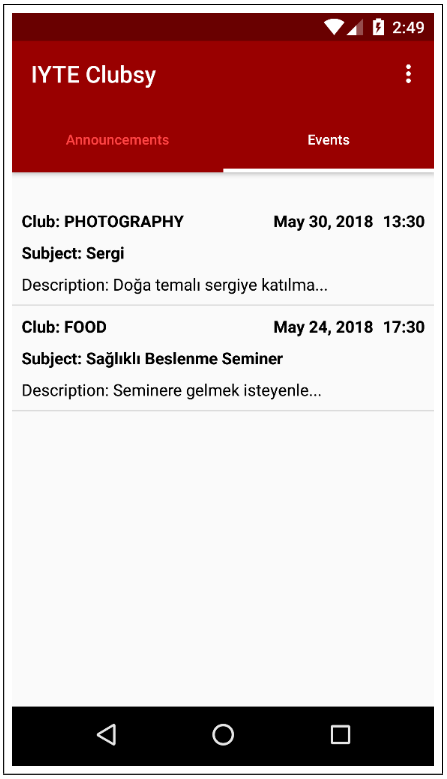
Figure 6: Reset Password Activity



Club: SOFTWARE
Subject: GitHub
Description: Lütfen GitHub hesaplarınızı admi...

Club: FOOD
Subject: Yemek Saatleri
Description: Uygun yemek saatlerini konuşm...

Figure 7: Main Activity, Announcements Tab



Club: PHOTOGRAPHY **May 30, 2018 13:30**
Subject: Sergi
Description: Doğa temalı sergiye katılma...

Club: FOOD **May 24, 2018 17:30**
Subject: Sağlıklı Beslenme Seminer
Description: Seminere gelmek isteyenle...

Figure 8: Main Activity, Events Tab

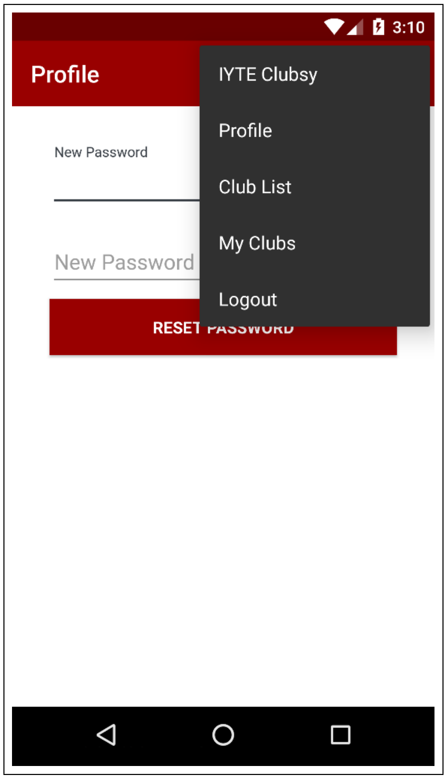
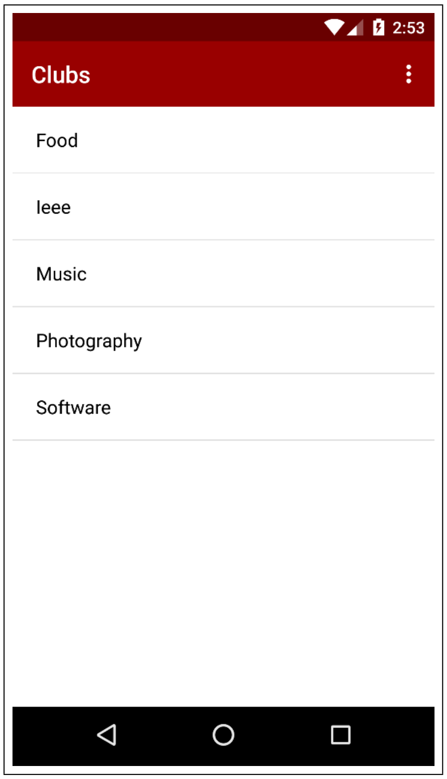


Figure 9: ToolBar



Food

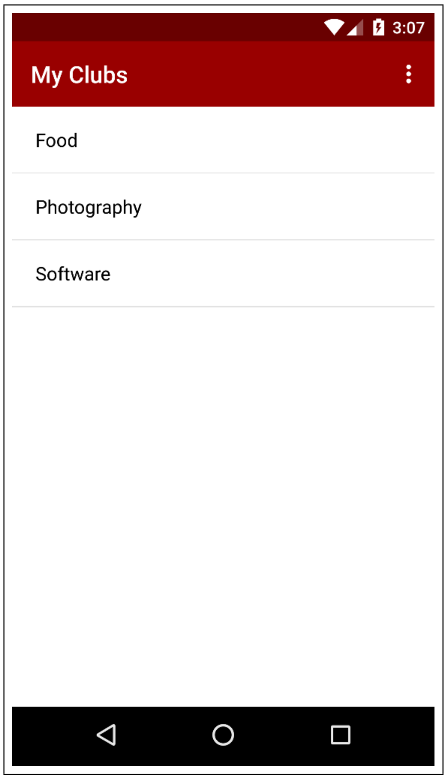
Ieee

Music

Photography

Software

Figure 10: Club List Activity

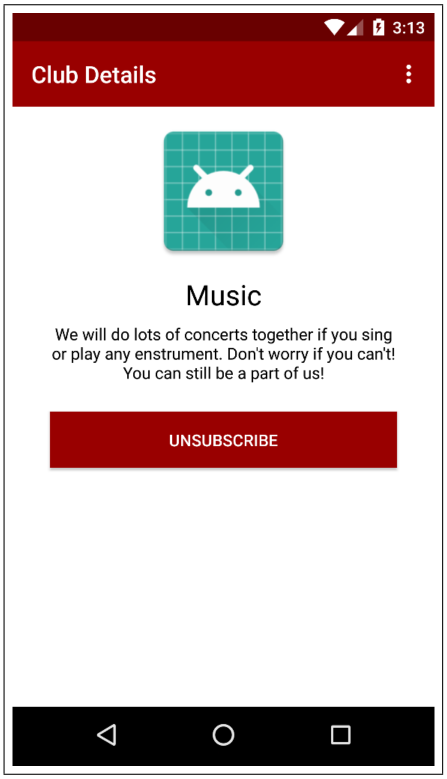


Food

Photography

Software

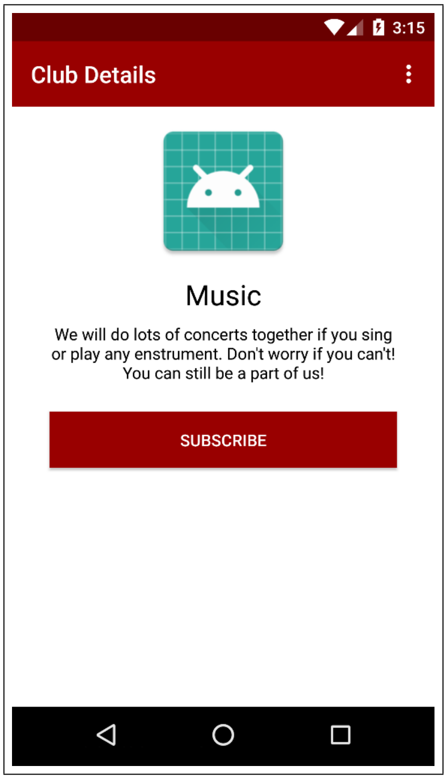
Figure 11: My Clubs Activity



Music

We will do lots of concerts together if you sing or play any enstrument. Don't worry if you can't! You can still be a part of us!

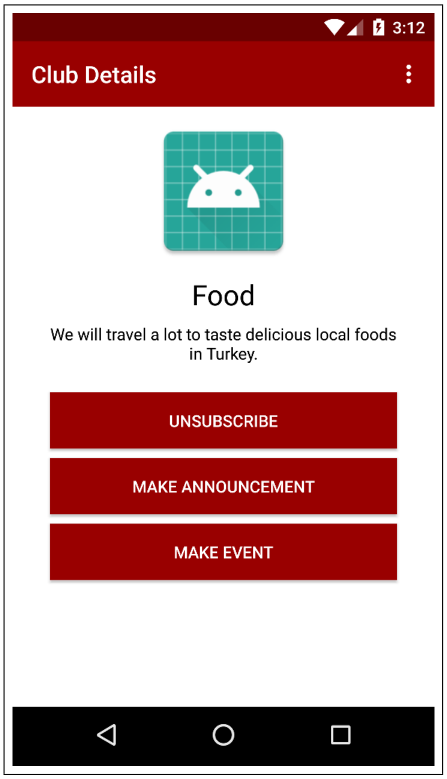
Figure 12: Club Details Activity(Subbed)



Music

We will do lots of concerts together if you sing or play any enstrument. Don't worry if you can't! You can still be a part of us!

Figure 13: Club Details Activity(Unsubbed)



We will travel a lot to taste delicious local foods in Turkey.

Figure 14: Club Details Activity(Admin)

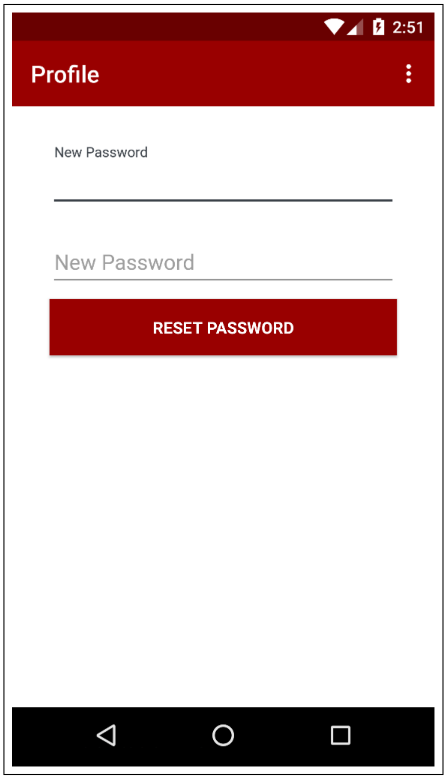


Figure 15: Profile Activity

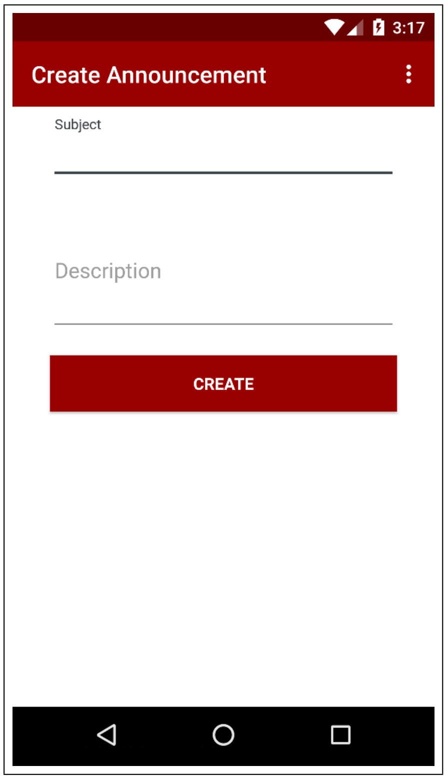


Figure 16: Create Announcement Activity

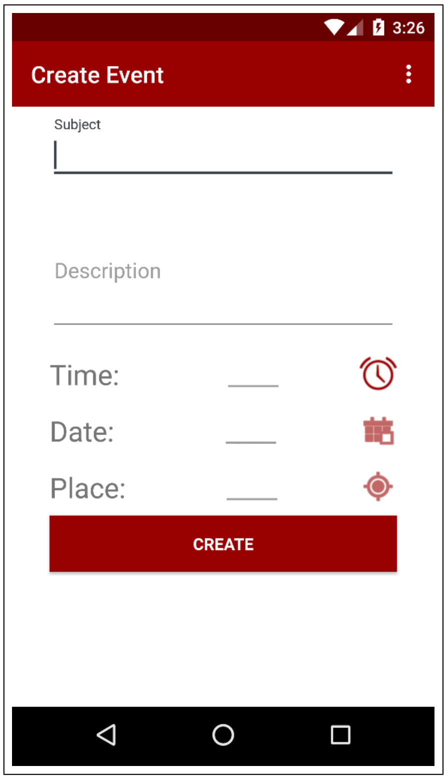


Figure 17: Create Event Activity

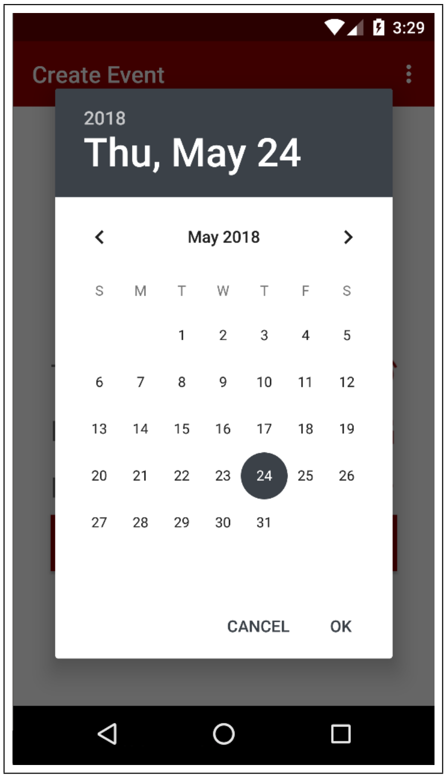


Figure 18: Date Picker Dialog

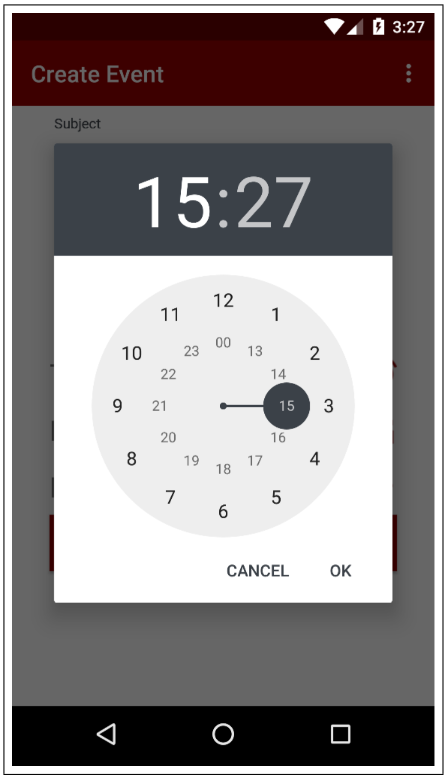


Figure 19: Time Picker Dialog

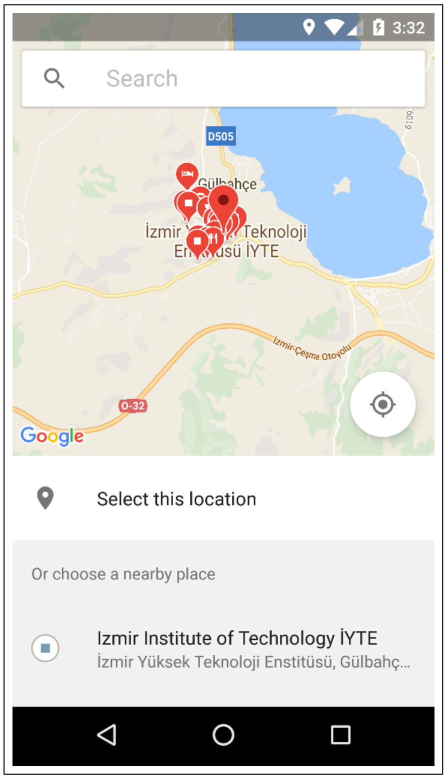


Figure 20: Place Picker Dialog

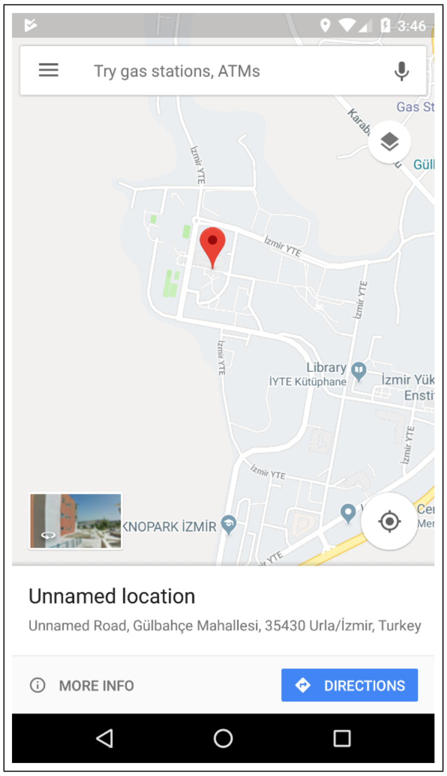


Figure 21: Google Maps

6 Detailed Design

6.1 Login Activity

6.1.1 Classification

Login Activity is a class that extends Android Activity class, with attributes and methods.

6.1.2 Definition and Responsibilities

Provides basic login feature to users, allowing them to access the application with their registered IYTE mails and passwords. Users can access other activities such as Register and Reset Password through this activity as well.

6.1.3 Constraints

Login needs already registered mail and correct password to let users access into the system.

6.1.4 Interactions

User is able to interact with this component through the GUI that is provided by the layout file associated with this activity. An example can be seen in the Screenshots section, Figure 4.

6.1.5 Processing

When Log In button is clicked, inputs are first checked. If there are empty inputs, users is warned. When there aren't, validity is checked. If the user entered valid credentials, activity jumps to Main Activity.

Upon clicking Register or Reset Password buttons, user is sent to respective activities.

6.1.6 Exports

Provide login to registered users. Users end up in Main Activity upon successful login.
Allow access to Register and Reset Password Activities.

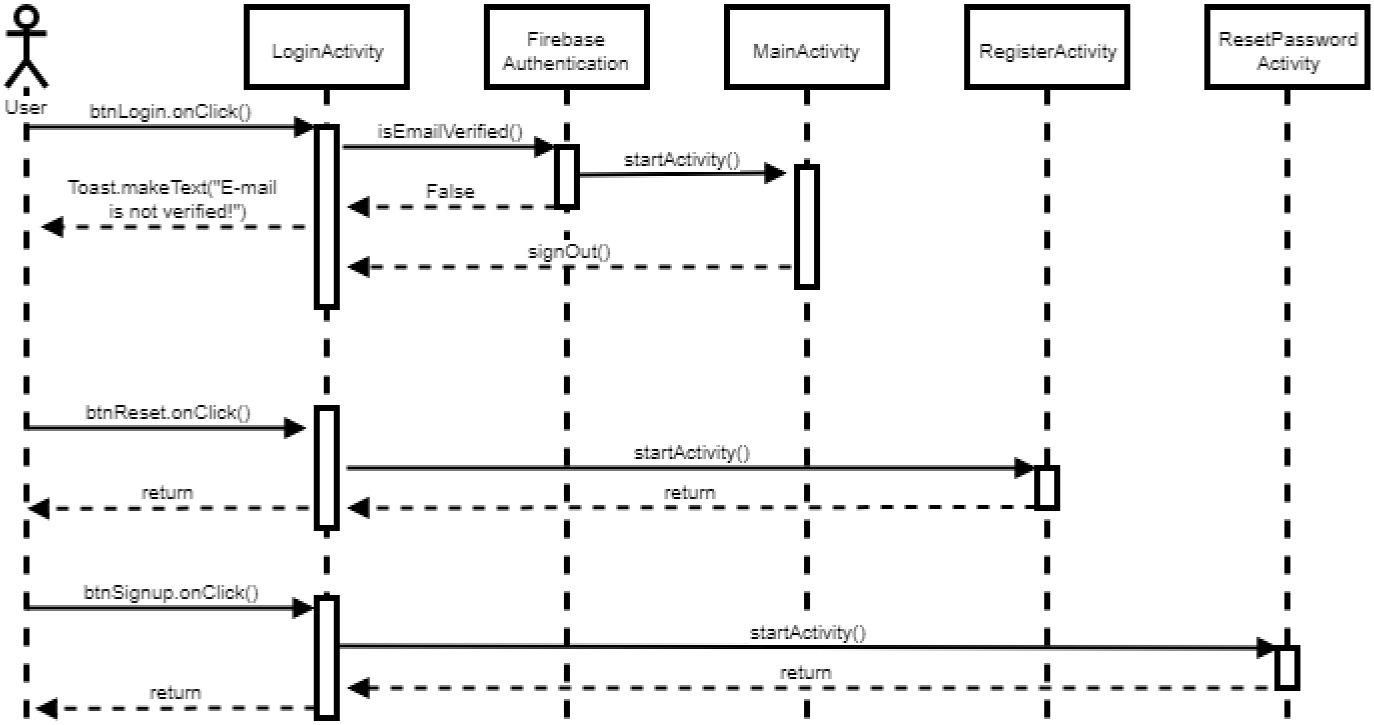


Figure 22: Login Activity Sequence Diagram

6.2 Register Activity

6.2.1 Classification

Register Activity is a class that extends Android Activity class, with attributes and methods.

6.2.2 Definition and Responsibilities

Provides registration to new users.

6.2.3 Constraints

Register needs valid IYTE e-mail and a password that is at least 6 characters long. Also, provided e-mail should not be already registered in the system.

6.2.4 Interactions

User is able to interact with this component through the GUI that is provided by the layout file associated with this activity. An example can be seen in the Screenshots section, Figure 5.

6.2.5 Processing

When Register button is clicked, inputs are first checked. If there are empty inputs, user is warned. Aside from that, entered e-mail has to be an IYTE mail that is not already registered. Passwords must be the same. If all these constraints are sufficed the registration is successful. When the registration is successful, an activation e-mail is sent to provided IYTE mail and user is returned to Login Activity. Account is only created after activation process is done.

6.2.6 Exports

Register new users to system. Users end up in Login Activity upon successful register.

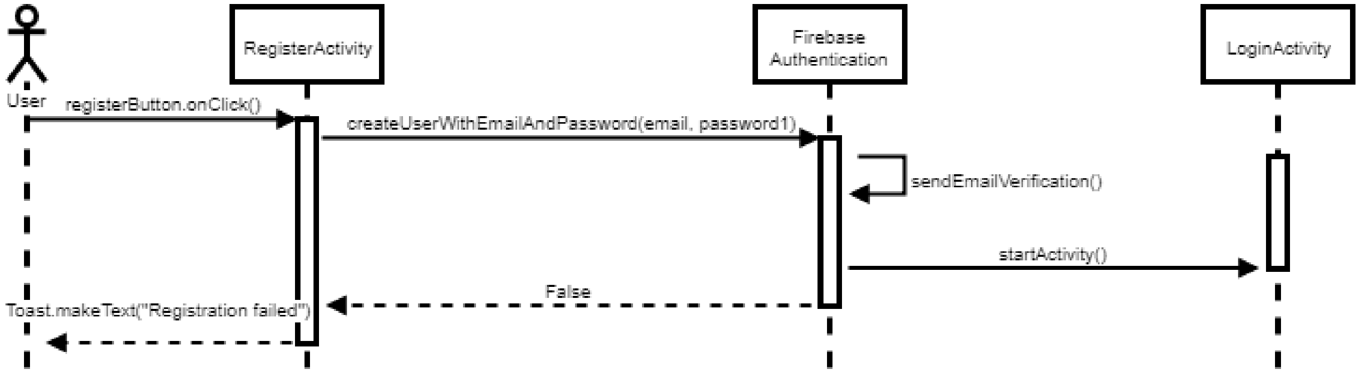


Figure 23: Register Activity Sequence Diagram

6.3 Reset Password Activity

6.3.1 Classification

Reset Password Activity is a class that extends Android Activity class, with attributes and methods.

6.3.2 Definition and Responsibilities

Lets users reset their password if they forget their current ones.

6.3.3 Constraints

An already registered and active account's e-mail has to be provided.

6.3.4 Interactions

User is able to interact with this component through the GUI that is provided by the layout file associated with this activity. An example can be seen in the Screenshots section, Figure 6.

6.3.5 Processing

When Reset button is clicked, input is first checked. If it is empty user is warned. Entered e-mail has to be an IYTE mail that is already registered. Upon success, an e-mail is sent to users which provides a temporary link that allows users to set their new password. User is then sent back to Login Activity.

6.3.6 Exports

Registered users are able to reset their passwords. Users end up in Login Activity upon successful reset.

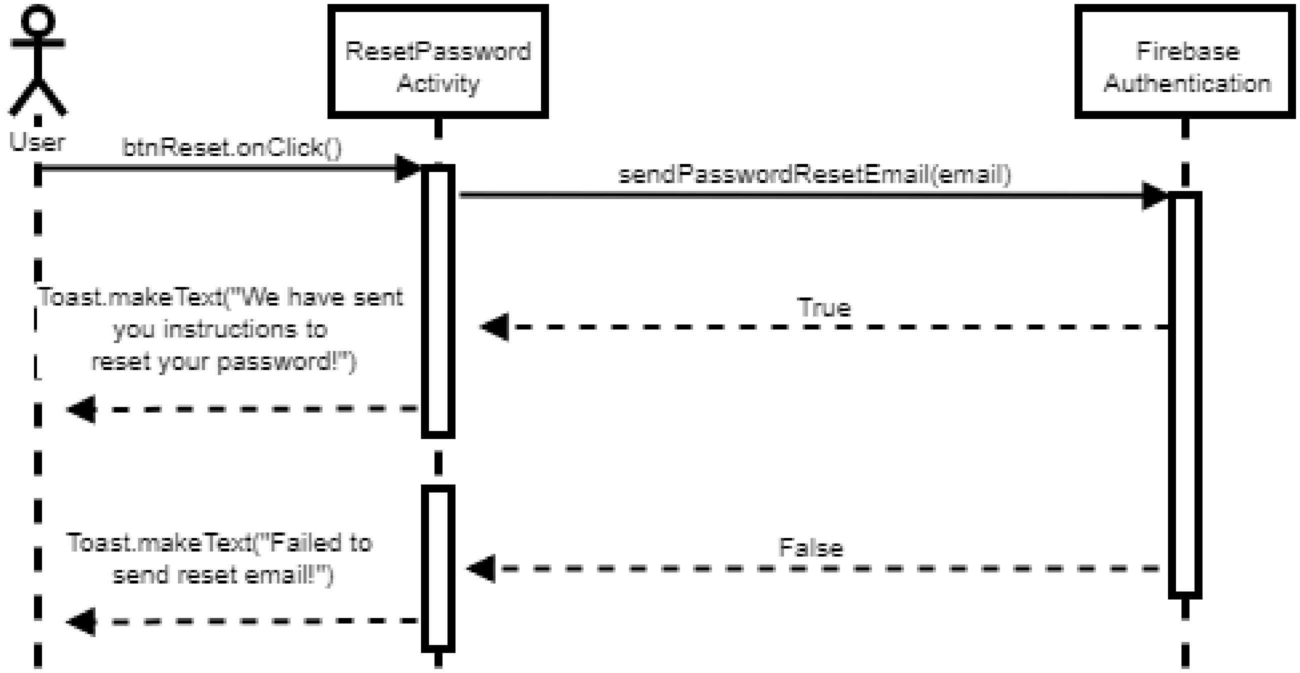

User
ResetPassword Activity
Firebase Authentication
btnReset.onClick()
sendPasswordResetEmail(email)
True
Toast.makeText("We have sent you instructions to reset your password!")
False
Toast.makeText("Failed to send reset email!")

Figure 24: Reset Password Activity Sequence Diagram

6.4 Club List Activity

6.4.1 Classification

Club List Activity is a class that extends Android Activity class, with attributes and methods.

6.4.2 Definition and Responsibilities

Let's users reset their password if they forget their current ones. The primary role of this activity is to show all the clubs that are stored in our database as a scroll-able and click-able list to inform the user about the contents.

6.4.3 Interactions

User is able to interact with this component through the GUI that is provided by the layout file associated with this activity. An example can be seen in the Screenshots section, Figure 10.

User can access this component by clicking ”Club Lists” option in the toolbar which is accessible from almost any activity page.

This activity also houses a toolbar at the top-right corner for easy navigation.

6.4.4 Processing

By clicking on one of the clubs from the list, user is sent to Club Details Activity with information on the selected club.

6.4.5 Exports

Lists all the clubs stored in database.
Allow users to access the club details.

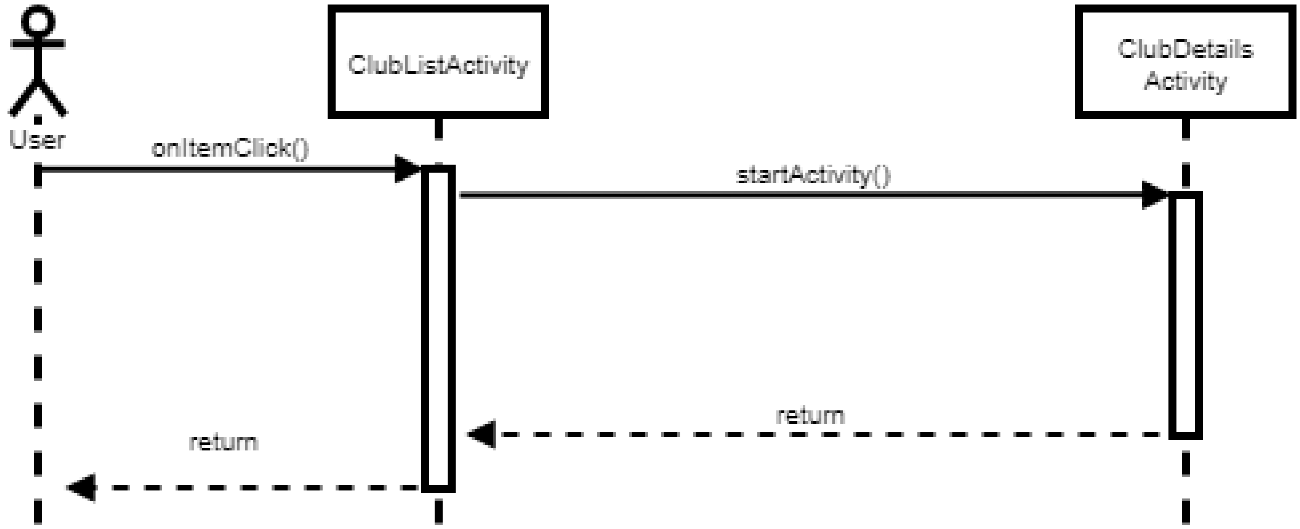


Figure 25: Club List Activity Sequence Diagram

6.5 My Clubs Activity

This activity is exactly the same with Club List Activity except that this activity lists the clubs that user is currently subscribed to. Same features provided with Club List Activity.

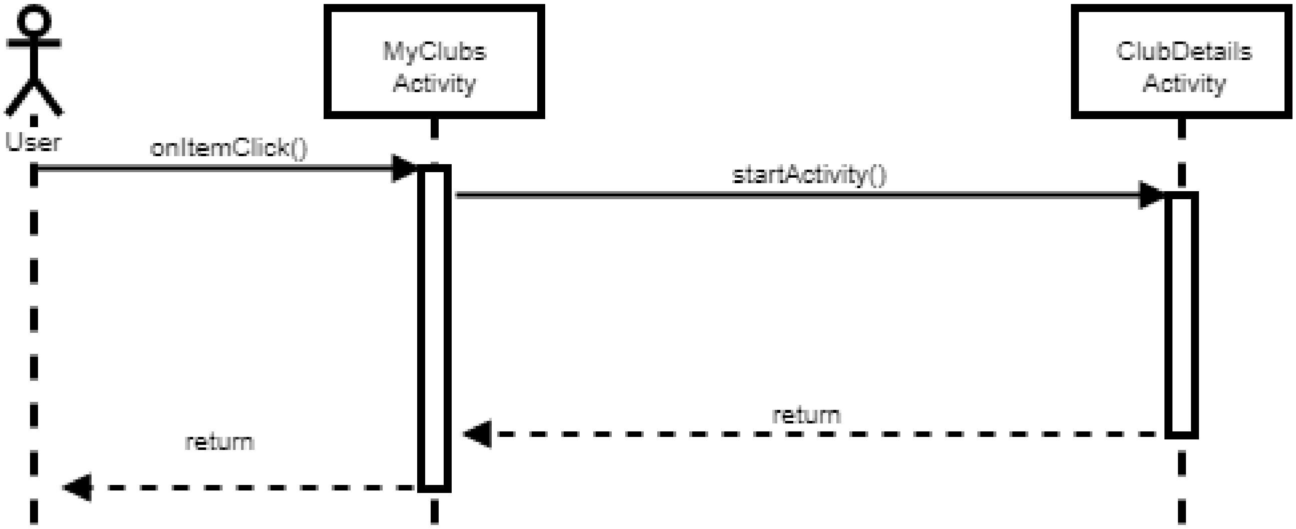


Figure 26: My Clubs Activity Sequence Diagram

6.6 Profile Activity

6.6.1 Classification

Profile Activity is a class that extends Android Activity class, with attributes and methods.

6.6.2 Definition and Responsibilities

Users are able to change their password through their profiles.

6.6.3 Constraints

The length of the password shouldn't be lower than six characters, and two input fields should be the same in order to reset the password.

6.6.4 Interactions

User is able to interact with this component through the GUI that is provided by the layout file associated with this activity. An example can be seen in the Screenshots section, Figure 15.

User can access this component by clicking "Profile" option in the toolbar which is accessible from almost any activity page.

This activity also houses a toolbar at the top-right corner for easy navigation.

6.6.5 Processing

User types in a new password and verifies it by typing for the second time and then clicks the Reset Password button. Inputs are verified and if successful, user is sent to Login Activity to re-log again with new password.

6.6.6 Exports

Allows users to update their passwords.

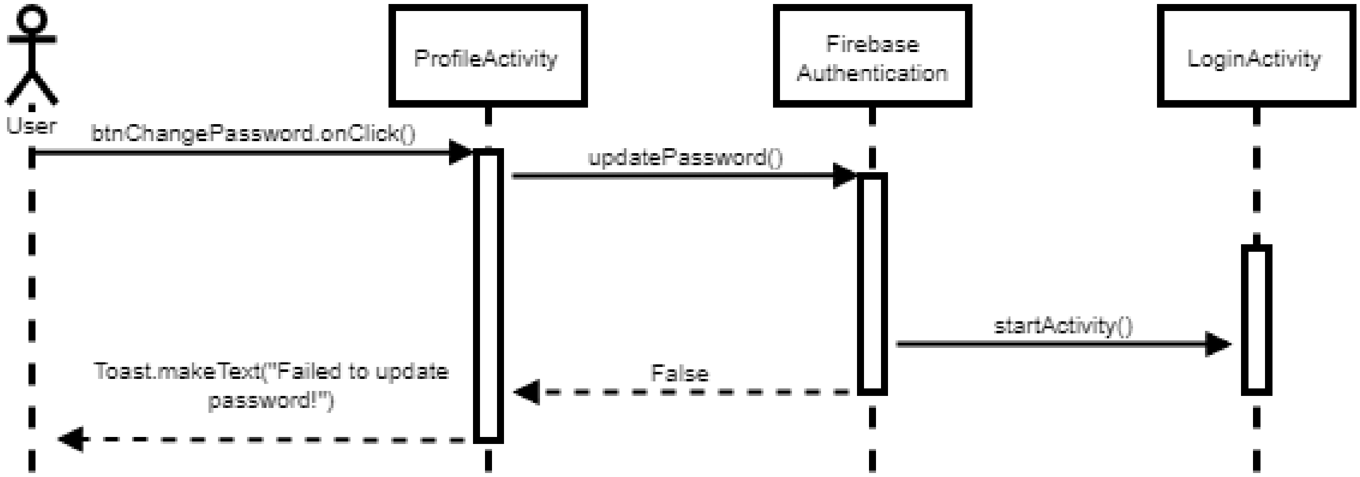


Figure 27: Profile Activity Sequence Diagram

6.7 Club Details Activity

6.7.1 Classification

Club Details Activity is a class that extends Android Activity class, with attributes and methods.

6.7.2 Definition and Responsibilities

This activity acts differently depending on the users properties. First of all, if user is not already subscribed to the club, it shows the ”Subscribe” button as shown in Figure 13 of Screenshots section. But if they are, instead the ”Unsubscribe” button is shown like in Figure 12. On top of that, if a user is subscribed and is also an admin of the club, it provides 2 more options, namely ”Make Announcement” and ”Make Event”, which can be seen in Figure 14.

6.7.3 Interactions

User is able to interact with this component through the GUI that is provided by the layout file associated with this activity. An example can be seen in the Screenshots section, Figure 12, 13 and 14.

User can access this component through My Clubs Activity or Club List Activity only.

This activity also houses a toolbar at the top-right corner for easy navigation.

6.7.4 Processing

Whenever user clicks on ”Subscribe” they are set to receive notifications about that Club’s Events and Announcements. They will also see these on the Main Activity. ”Unsubscribe” will remove these. Creating Event/Announcement will register these in the database, which in turn will execute Cloud Functions for Notification delivery.

6.7.5 Exports

Let user subscribe to a Club that is not already subscribed.
Let user unsubscribe from a Club that is already subscribed.
Let admins of the club make Event/Announcement.

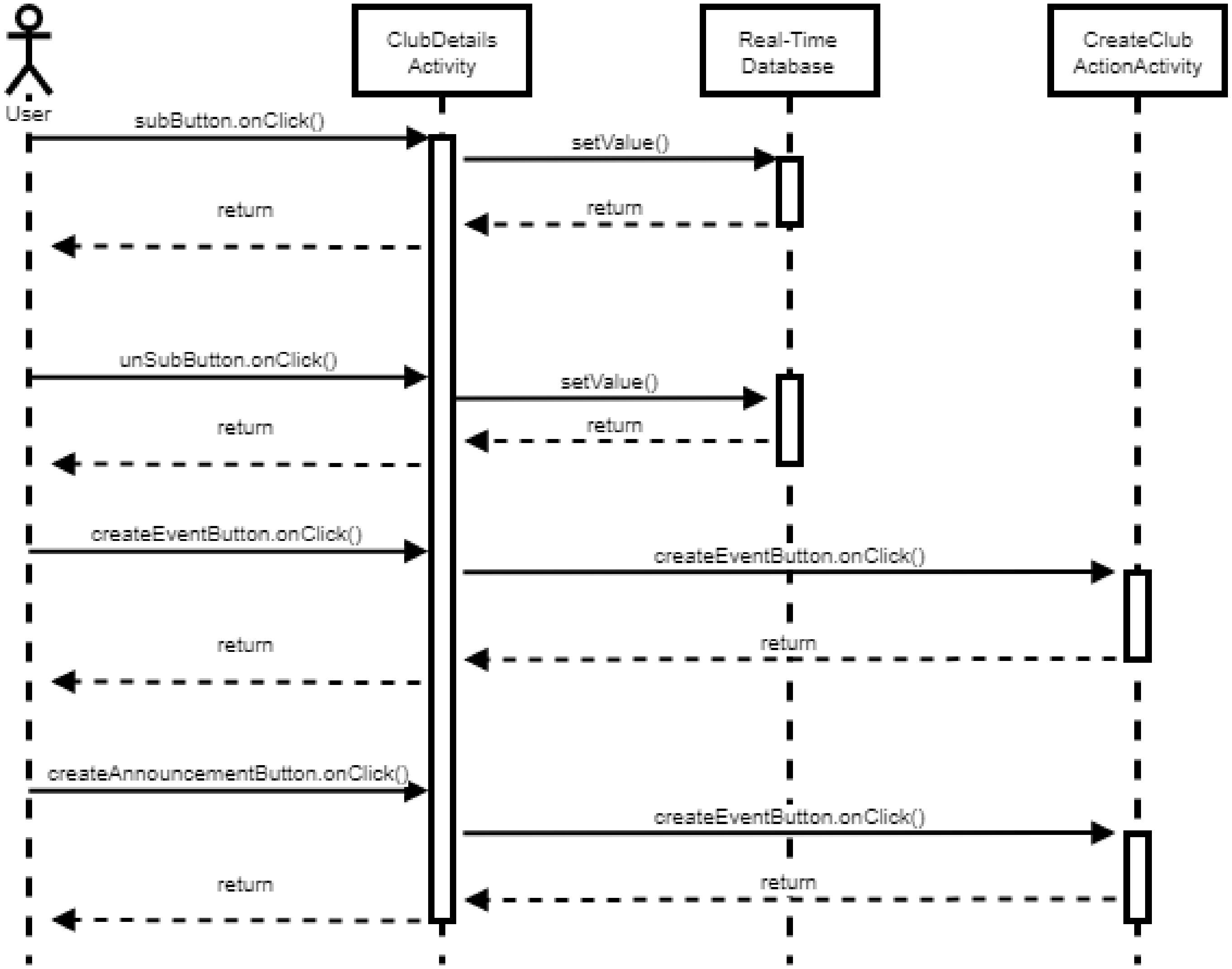


Figure 28: Club Details Activity Sequence Diagram

6.8 Create Club Action Activity

6.8.1 Classification

Create Club Action Activity is a class that extends Android Activity class, with attributes and methods.

6.8.2 Definition and Responsibilities

This activity deals with the creation of Events/Announcements and also is responsible from adding them to Database. It also houses PlacePickerDialog as a built in function.

6.8.3 Interactions

User is able to interact with this component through the GUI that is provided by the layout file associated with this activity. An example can be seen in the Screenshots section, Figure 16 and 17.

User can access this component through clicking on one of the ”Make Announcement” or ”Make Event” buttons, which can be found in Club Details Activity.

This activity also houses a toolbar at the top-right corner for easy navigation.

6.8.4 Processing

For announcements, Subject and Description fields must be filled, if not user is warned. If successful, user is notified and returned to Main Activity.

For events, additionally Time, Date and Place fields must be set through the click-able buttons beside them. Clicking on these buttons will call for TimePickerFragment to show Time Picker Dialog, DatePickerFragment to show Date Picker Dialog and built-in PlacePickerDialog to show Google Maps respectfully.

6.8.5 Exports

Let admins create announcement/events.

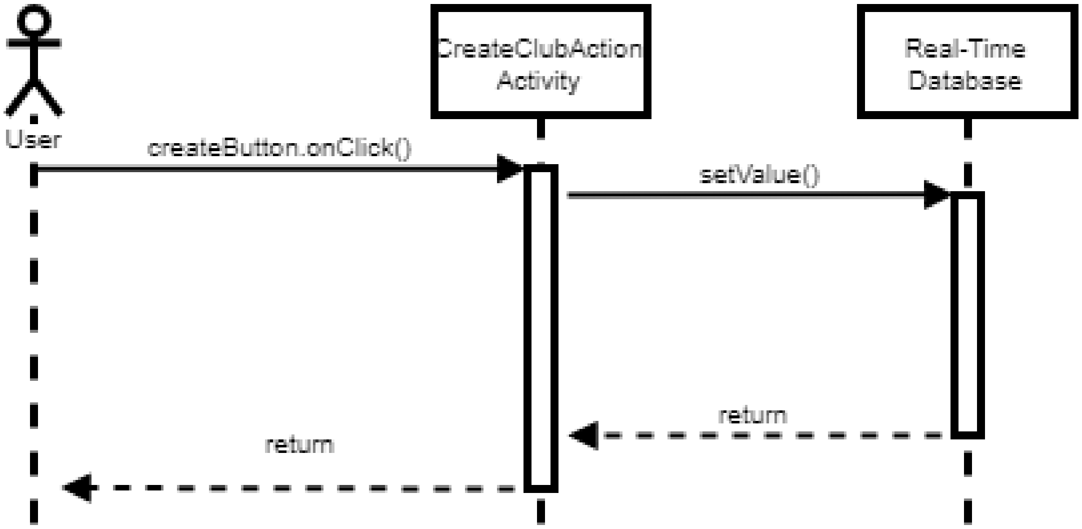


Figure 29: Create Club Action Activity Sequence Diagram

6.9 Main Activity

6.9.1 Classification

Main Activity is a class that extends Android Activity class, with attributes and methods.

6.9.2 Definition and Responsibilities

This activity is responsible from showing the user Events/Announcements of the clubs they are subscribed to.

6.9.3 Interactions

User is able to interact with this component through the GUI that is provided by the layout file associated with this activity. An example can be seen in the Screenshots section, Figure 7 and 8.

User can access this activity through "IYTE Clubsy" option in the toolbar, which is accessible from almost anywhere.

This activity also features tabs, to separate between announcements and events.

This activity also houses a toolbar at the top-right corner for easy navigation.

6.9.4 Processing

Data is retrieved from Database and shown on the screen inside the List using adapters and listener objects. On the events section, items are also click-able and will open up Google Maps with the specified coordinates to show the Event's location to user.

6.9.5 Exports

Let user see announcements/events of the clubs they are subscribed to. Let user see place of events by clicking on them.

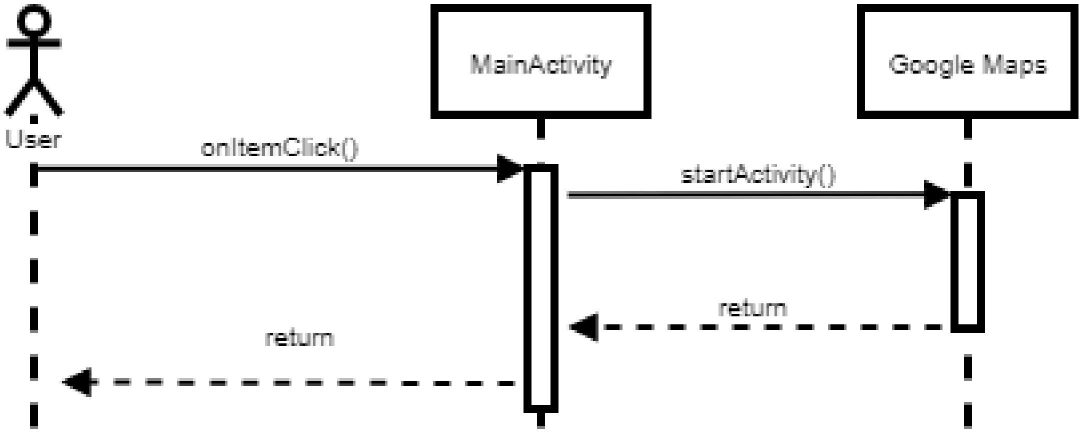


Figure 30: Main Activity Sequence Diagram

6.10 Time Picker Fragment

6.10.1 Classification

Time Picker Fragment is a class that extends Dialog Fragment class that is provided by Android, with attributes and methods.

6.10.2 Definition and Responsibilities

This fragment is responsible from letting user select a time in 24h format and set it to Events time input.

6.10.3 Interactions

User is able to interact with this component through the GUI that is provided by the Android Operating System associated with this fragment. An example can be seen in the Screenshots section, Figure 19.

User can access this dialog by clicking on the symbol beside "Time" input in Create Club Action Activity.

6.10.4 Processing

Selected time is set to input field for further processing by the parent activity.

6.10.5 Exports

Let users set the time of event.

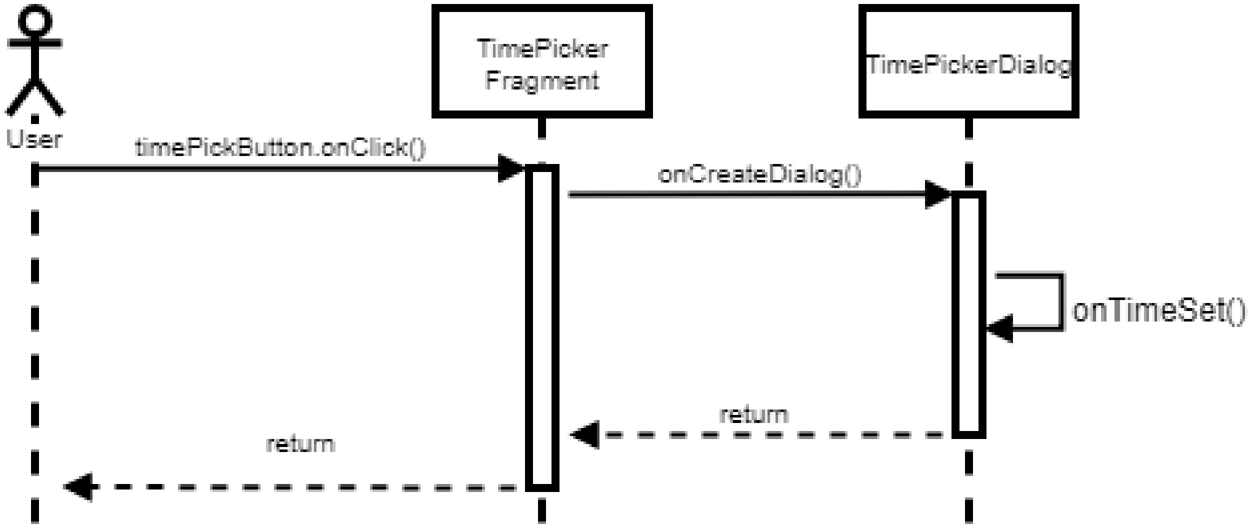


Figure 31: Time Picker Fragment Sequence Diagram

6.11 Date Picker Fragment

6.11.1 Classification

Date Picker Fragment is a class that extends Dialog Fragment class that is provided by Android, with attributes and methods.

6.11.2 Definition and Responsibilities

This fragment is responsible from letting user select a date and set it to Events date input.

6.11.3 Interactions

User is able to interact with this component through the GUI that is provided by the Android Operating System associated with this fragment. An example can be seen in the Screenshots section, Figure 18.

User can access this dialog by clicking on the symbol beside ”Date” input in Create Club Action Activity.

6.11.4 Processing

Selected date is set to input field for further processing by the parent activity.

6.11.5 Exports

Let users set the date of event.

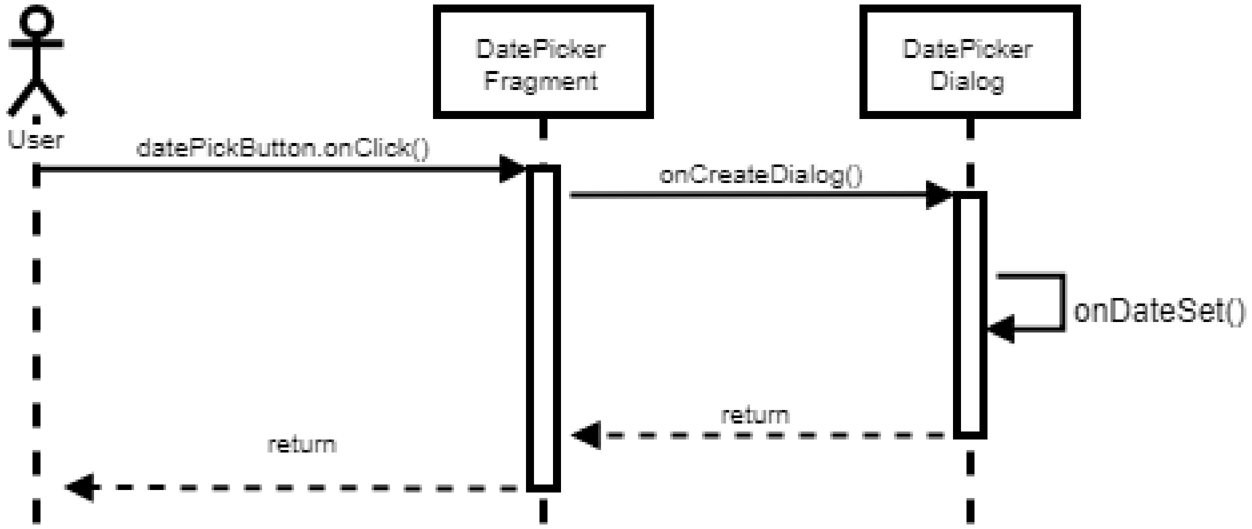


Figure 32: Date Picker Fragment Sequence Diagram

7 Technologies and Tools

7.1 Android Studio

Android Studio is a complete and de facto IDE for Android development. Through the use of packages we have integrated Git and Kotlin support. Git and GitHub support came pretty handy developing and testing as a team. Even though it has built in emulation for testing, it does not support this feature in AMD based systems, therefore we have used another virtualization program called Genymotion alongside it.

7.2 Genymotion

Installed as a standalone program but also integrated inside the Android Studio, is the third-party Android emulation and virtualization program. Used for testing and debugging of our application.

7.3 Google Firebase

Google's platform that provides Database, Authentication and Cloud Functions among many other things to many different platforms among Android. Firebase API was heavily used throughout the program to communicate with the Real-Time Database and Authentication systems.

Cloud functions were written in TypeScript language inside Node.js environment and uploaded to Google Firebase servers for execution. They were used for reliable notification delivery to devices.

8 Conclusion

This Software Design Document provides a complete description of IYTE Clubsy Project, a design project of IYTE Computer Engineering Department Software Engineering course. It tries to explain the design and implementation of the application, constraints and dependencies in full detail. It is a complete product with ability to maintain and extend upon easily. It tries to provide a simple and minimalistic application yet easy to use and user friendly fully featured packet.