

**CS102**

**Spring 2020/21**

**Project  
Group:**

**G1C**

**Assistant:**

**Haya Shamim Khan Khattak**

## **~ University Social App ~**

**unION**

**Author Names:**

**Kutay Şenyiğit**

**Mert Ünlü**

**Yunus Eren Türkeri**

**Ömer Oktay Gültekin**

**Ege İpekci**

**Detailed-Design Report**

**(v2.0)**

**3 May 2021**

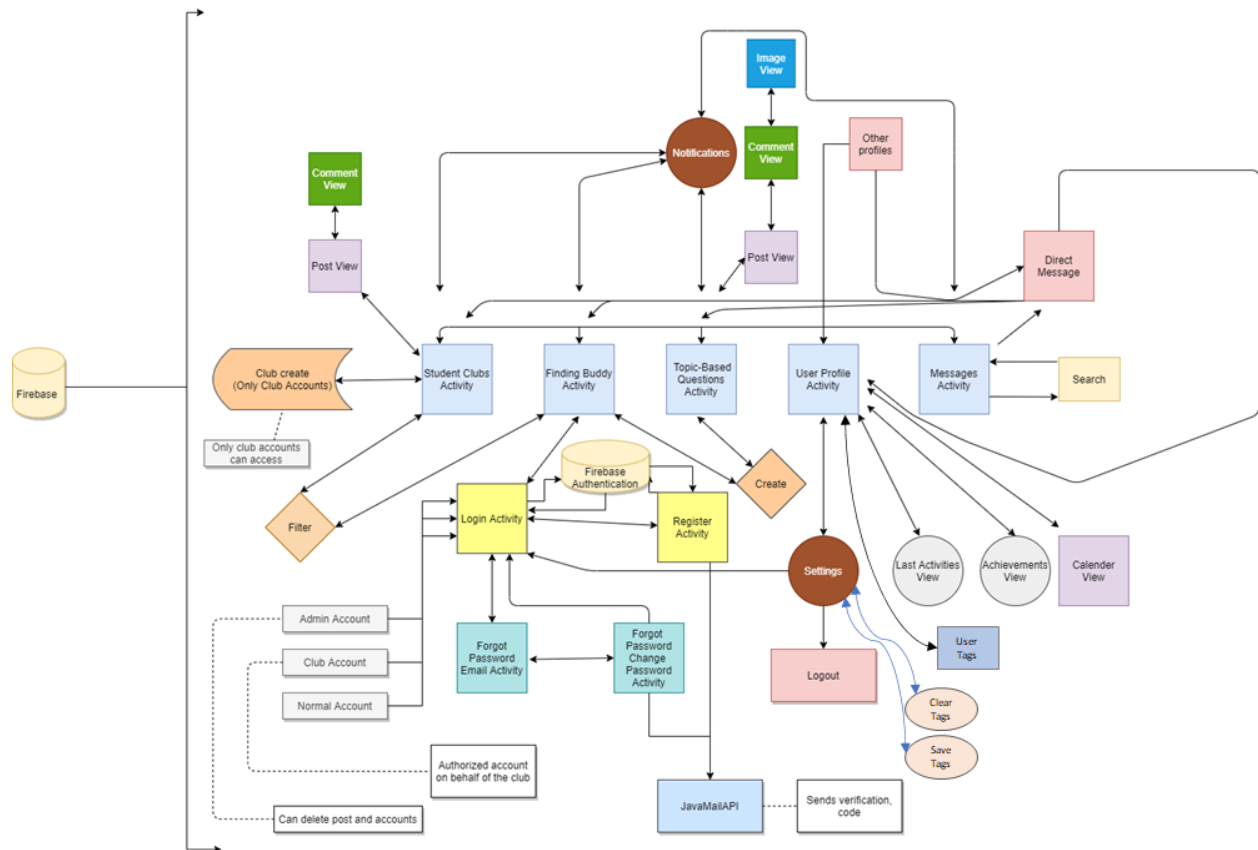
# 1. Introduction

University Social App is a mobile application for university students to interact with each other over their common questions and interests.

## 2. System Overview

Using the MVC Pattern, an online application for the [Android Platform](#) will be provided. The tool [Android Studio](#). [XML](#) will be used to develop the app unIOn, in order to create the layout. For the database, we will use [Firebase Authentication](#), [Firebase Storage](#), [Firebase Realtime Database](#). [JavaMailAPI](#) will be used for the verification of the user email and user password. [Picasso](#) is going to be used to set Images.

### 1. Architecture Diagram:



## 2. Model classes

- ModelAchievements
- ModelAchievementsScores
- ModelBuddyAndClubPost
- ModelCalendar
- ModelChat
- ModelChatlist
- ModelComment
- ModelLastActivities
- ModelNotification
- ModelStackPost
- ModelUsers

## 3. View Classes

- BuddyFragment
- ChatActivity
- ClubsFragment
- CreateAnAccountActivity
- LoginActivity
- MainActivity
- MessageFragment
- OtherProfile
- PostActivity
- ProfileFragment
- RecoverAccountActivity
- StackFragment
- VerifyAccountActivity

## 4. Layout Classes

- activity\_chat.xml
- activity\_create\_an\_account.xml
- activity\_login.xml
- activity\_main.xml
- activity\_other\_profile.xml
- activity\_post.xml
- activity\_recover\_account.xml
- activity\_test.xml
- activity\_verify\_account.xml
- custom\_buddy\_filter.xml
- custom\_club\_filter.xml
- custom\_create\_club\_post.xml
- custom\_create\_post\_buddy\_popup.xml
- custom\_feed\_card.xml
- custom\_notification\_popup.xml
- custom\_popup.xml
- custom\_popup\_achievements.xml
- custom\_settings.xml
- custom\_stack\_createpost\_popup.xml
- custom\_stack\_filter.xml
- custom\_stack\_over\_flow\_feed\_cards.xml
- custom\_user\_calendar\_popup.xml
- custom\_view\_club\_post\_popup.xml
- custom\_view\_stack\_post\_popup.xml
- fragment\_buddy.xml
- fragment\_clubs.xml
- fragment\_message.xml
- fragment\_profile.xml
- fragment\_stack.xml
- last\_acts\_list\_layout.xml
- last\_acts\_list\_layout\_pu.xml
- login\_signup\_button.xml
- row\_achievements.xml
- row\_calendar.xml
- row\_chat\_left.xml
- row\_chat\_right.xml
- row\_chatlist.xml
- row\_comment.xml
- row\_lastachi.xml
- row\_notification.xml
- row\_users.xml
- stack\_bigger\_comment.xml
- tag\_button.xml
- text\_view.xml
- user\_pp.xml

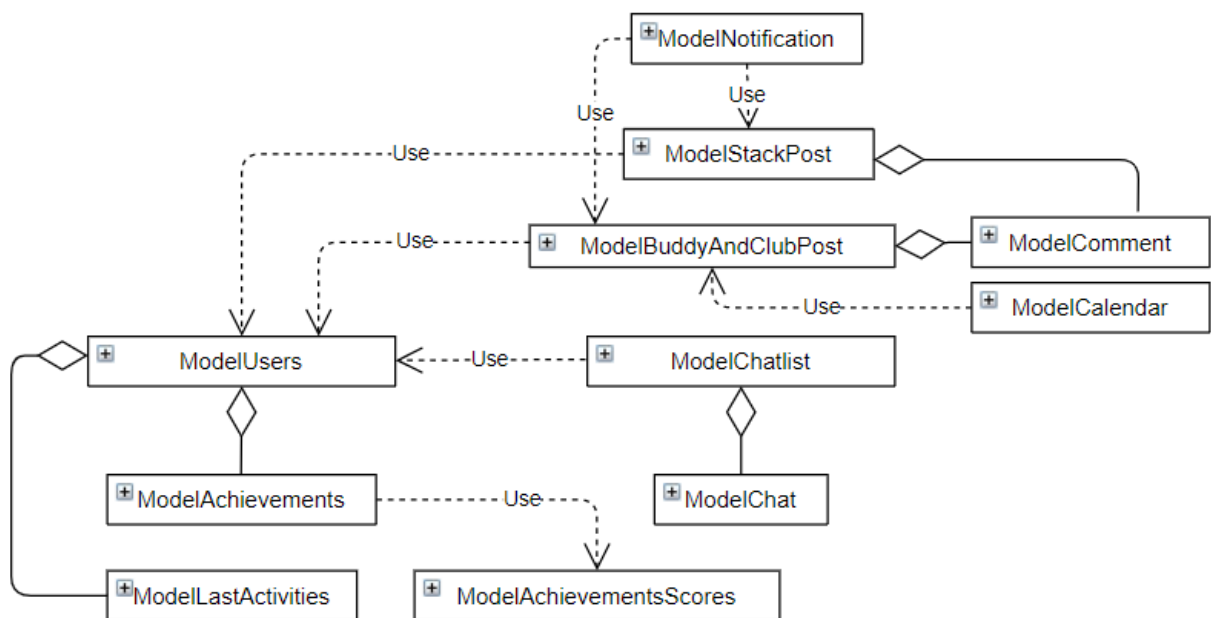
## 5. Controller Classes

- controllers
  - notifications
    - IService
    - Client
    - Data
    - FirebaseMessaging
    - FirebaseService
    - OreoAndAboveNotification
    - Response
    - Sender
    - Token
  - AdapterAchievements
  - AdapterBuddyPosts
  - AdapterCalendar
  - AdapterChat
  - AdapterChatlist
  - AdapterClubPosts
  - AdapterComment
  - AdapterLastActivities
  - AdapterNotification
  - AdapterSearchProfile
  - AdapterStackPosts
  - AdapterUsers
  - JavaMailAPI
  - KeyboardUtils
  - LocalDataManager
  - Utils

### 3. Core Design Details

Note: In this section, the demonstrations for the UML schematics can be seen. Due to the huge sizes of the classes' contents, we wanted to present the simplified version for the UML diagram in section 1, and in section 2, the contents of each class can be viewed as separate tables. In section 3, just for the ease of examining the extended versions of the classes with UML notations, we provided an open link. There, page 1 is the simplified version, page 2 is the extended version, and page 3 is the table-by-table view of the classes.

#### 1. UML Diagram (Simplified Version):



#### 2. UML Diagram (Extended Tables Version):

ModelUsers
<p>accountState: String</p> <p>accountType: String</p> <p>achievements: String</p> <p>email: String</p> <p>pp: String</p> <p>tags: String</p> <p>uid: String</p> <p>username: String</p>
<p>ModelUsers( accountState: String, accountType: String, achievements: String, email: String, pp: String, tags: String, uid: String, username: String): void</p> <p>ModelUsers(): void</p> <p>getAccountState(): String</p> <p>setAccountState( accountState: String): void</p> <p>getAccountType(): String</p> <p>setAccountType( accountType: String): void</p> <p>getAchievements(): String</p> <p>setAchievements( achievements: String): void</p> <p>getEmail(): String</p> <p>setEmail( email: String): void</p> <p>getPp(): String</p> <p>setPp( pp: String): void</p> <p>getTags(): String</p> <p>setTags( tags: String): void</p> <p>getUid(): String</p> <p>setUid( uid: String): void</p> <p>getUsername(): String</p> <p>setUsername( username: String): void</p>

ModelNotification
<p>pId: String</p> <p>timestamp: String</p> <p>pUid: String</p> <p>notification: String</p> <p>sUid: String</p> <p>sName: String</p> <p>sEmail: String</p> <p>sImage: String</p> <p>sTag: String</p>
<p>ModelNotification( pId: String, timestamp: String, pUid: String, notification: String, sUid: String, sName: String, sImage: String, sTag: String): void</p> <p>ModelNotification(): void</p> <p>getEmail(): String</p> <p>setEmail( sEmail: String): void</p> <p>getTag(): String</p> <p>setTag( sTag: String): void</p> <p>getpId(): String</p> <p>setpId( pId: String): void</p> <p>getTimestamp(): String</p> <p>setTimestamp( timestamp: String): void</p> <p>getpUid(): String</p> <p>setpUid( pUid: String): void</p> <p>getNotification(): String</p> <p>setNotification( notification: String): void</p> <p>getsUid(): String</p> <p>setsUid( sUid: String): void</p> <p>getName(): String</p> <p>setName( sName: String): void</p> <p>getImage(): String</p> <p>setImage( sImage: String): void</p>

ModelChat	ModelAchievements Scores
<p>message : String</p> <p>receiver : String</p> <p>sender : String</p> <p>timestamp : String</p> <p>isSeen : boolean</p>	<p>math : String</p> <p>career : String</p> <p>sport : String</p> <p>technology : String</p> <p>social : String</p> <p>english : String</p> <p>turkish : String</p> <p>study : String</p>
<p>ModelChat( message : String, receiver : String, sender : String, timestamp : String, isSeen : boolean ) : void</p> <p>ModelChat() : void</p> <p>getMessage() : String</p> <p>setMessage( message : String ) : void</p> <p>getReceiver() : String</p> <p>setReceiver( receiver: String ) : void</p> <p>getSender() : String</p> <p>setSender( sender : String ) : void</p> <p>getTimestamp() : String</p> <p>setTimestamp( timestamp : String ) : void</p> <p>isSeen() : boolean</p> <p>setSeen( seen : boolean ) : void</p>	<p>ModelAchievementsScores( career : String, english : String, math : String, social : String, sport : String, study : String, technology : String, turkish : String ) : void</p> <p>ModelAchievementsScores() : void</p> <p>getMath() : String</p> <p>setMath( math : String ) : void</p> <p>getCareer() : String</p> <p>setCareer( career : String ) : void</p> <p>getSport() : String</p> <p>setSport( sport : String ) : void</p> <p>getTechnology() : String</p> <p>setTechnology( technology : String ) : void</p> <p>getSocial() : String</p> <p>setSocial( social : String ) : void</p> <p>getEnglish() : String</p> <p>setEnglish( english : String ) : void</p> <p>getTurkish() : String</p> <p>setTurkish( turkish : String ) : void</p> <p>getStudy() : String</p> <p>setStudy( study : String ) : void</p>

ModelCalendar
<p>pDate: String</p> <p>pHour: String</p> <p>pId: String</p> <p>pTitle: String</p> <p>pType: String</p> <p>username: String</p>
<p>ModelCalendar( pDate: String, pHour: String, pId: String, pTitle: String, pType: String, username: String): void</p> <p>ModelCalendar(): void</p> <p>getpDate(): String</p> <p>setpDate( pDate: String): void</p> <p>getpHour(): String</p> <p>setpHour( pHour: String): void</p> <p>getpId(): String</p> <p>setpId( pId: String): void</p> <p>getpTitle(): String</p> <p>setpTitle( pTitle: String): void</p> <p>getpType(): String</p> <p>setpType( pType: String): void</p> <p>getUsername(): String</p> <p>setUsername( username: String): void</p>

ModelChatlist
<p>id : String</p>
<p>ModelChatlist( id : String ) : void</p> <p>ModelChatlist(): void</p> <p>getId(): String</p> <p>getId( id : String ) : void</p>

ModelStackPost
<p>pId : String</p> <p>pTitle : String</p> <p>pDetails : String</p> <p>pImage : String</p> <p>pTime : String</p> <p>uid : String</p> <p>uEmail : String</p> <p>pUpVoteNumber : int</p> <p>pTagIndex : String</p> <p>username : String</p> <p>pAnon : String</p>
<p>ModelStackPost( pId : String, pTitle : String, pDetails : String, pImage : String, pTime : String, uid : String, uEmail : String, pAnon : String, username : String, pUpVoteNumber : Integer, pTagIndex : String, pUpUsers : List&lt;String&gt; ) : void</p> <p>ModelStackPost(): void</p> <p>getPId(): String</p> <p>getpUpUsers(): List&lt;String&gt;</p> <p>setpTagIndex( pTagIndex : String ) : void</p> <p>getpTagIndex(): String</p> <p>setUsername( username : String ) : void</p> <p>getUsername(): String</p> <p>setPAnon( pAnon : String ) : void</p> <p>getPAnon(): String</p> <p>setpUpVoteNumber( pUpVoteNumber : int ) : void</p> <p>getpUpVoteNumber(): int</p> <p>setUEmail( uEmail : String ) : void</p> <p>getUEmail(): String</p> <p>setUid( uid: String ) : void</p> <p>getUid(): String</p> <p>setPTime( pTime : String ) : void</p> <p>getPTime(): String</p> <p>setPImage( pImage : String ) : void</p> <p>getPImage(): String</p> <p>setPDetails( pDetails : String ) : void</p> <p>getPDetails(): String</p> <p>setPTitle( pTitle : String ) : void</p> <p>getPTitle(): String</p> <p>setPId( pId : String ) : void</p>

 ModelComment	 ModelBuddyAndClubPost
<p> cId : String  comment : String  timeStamp : String  uid : String  cAnon : String  cPhoto : String  uName : String  upNumber : int  cUpUsers List&lt;String&gt; </p>	<p> pId: String  pTitle: String  pDate: String  pHour: String  pLocation: String  pQuota: String  pDetails: String  pImage: String  pTime: String  uid: String  uEmail: String  pAnon: String  username: String  pTags: String  pGender: String  uPp: String </p>
<p> ModelComment( cId : String, comment :String, timeStamp : String, upNumber : Integer, uid : String, cAnon :String, cPhoto :String, uName :String ,List&lt;String&gt; cUpUsers ) : void  ModelComment() : void  getCId() : String  setCId( cId : String ) : void  getComment() : String  setComment( comment : String ) : void  getTimeStamp() : String  setTimeStamp( timeStamp : String ) : void  getUpNumber() : int  setUpNumber( upNumber : int ) : void  getUid() : String  setUid( uid : String ) : void  getCAnon() : String  setCAnon( cAnon : String ) : void  getCPhoto() : String  setCPhoto( cPhoto : String ) : void  getUName() : String  setUName( uName: String ) :void  getUpUsers() : List&lt;String&gt;  setUpUsers( cUpUsers : List&lt;String&gt; ) : void </p>	<p> ModelBuddyAndClubPost( pId: String, pTitle: String, pDate: String, pHour: String, pLocation: String, pQuota: String, pDetails: String, pImage: String, pTime: String, uid: String, uEmail: String, pAnon: String, username: String, pTags: String, pGender: String, uPp: String): void  ModelBuddyAndClubPost(): void  getpId(): String  setpId( pId: String): void  getpTitle(): String  setpTitle( pTitle: String): void  getpDate(): String  setpDate( pDate: String): void  getpHour(): String  setpHour( pHour: String): void  getpLocation(): String  setpLocation( pLocation: String): void  getpQuota(): String  setpQuota( pQuota: String): void  getpDetails(): String  setpDetails( pDetails: String): void </p>



```
getImage():  
setImage( pImage: String): void  
getTime(): String  
setpTime( pTime: String): void  
getUid(): String  
setUid( uid: String): void  
getEmail(): String  
setuEmail( uEmail: String): void  
getpAnon(): String  
setpAnon( pAnon: String): void  
getUsername(): String  
getpTags(): String  
setpTags( pTags: String): void  
setUsername( username: String): void  
getpGender(): String  
setpGender( pGender: String): void  
getuPp(): String  
setuPp( uPp: String): void
```

ModelLastActivities
<p> pld: String  timestamp: String  pUid: String  notification: String  sUid: String  sName: String  sEmail: String  sImage: String  sTag: String  type: String </p>
<p> ModelLastActivities( pld: String, timestamp: String, sUid: String, sName: String, sTag: String, type: String): void  ModelLastActivities(): void  getType(): String  setType( type: String): void  getEmail(): String  setEmail( sEmail: String): void  getTag(): String  setTag( sTag: String): void  getpId(): String  setpId( pld: String): void  getTimestamp(): String  setTimestamp( timestamp: String): void  getpUid(): String  setpUid( pUid: String): void  getNotification(): String  setNotification( notification: String): void  getsUid(): String  setsUid( sUid: String): void  getName(): String  setName( sName: String): void  getImage(): String  setImage( sImage: String): void </p>

ModelAchievements
<p> title: String  description: String  point: String  genre: String  level: String  nId: String </p>
<p> ModelAchievements( title: String, description: String, point: String, genre: String, level: String, nId: String): void  ModelAchievements(): void  getnId(): String  setnId( nId: String): String  getLevel(): String  setLevel( level: String): void  getTitle(): String  setTitle( title: String): void  getDescription(): String  setDescription( description: String): void  getPoint(): String  setPoint( point: String): void  getGenre(): String  setGenre( genre: String): void </p>

### 3. UML Diagram (Extended Tables & Notationed Version):

The following link includes 3 pages:

Page-1 : The simplified version of the UML diagram with notations, but missing content.

Page-2 : The extended version of the UML diagram with notations and class contents.

Page-3 : The extended version of the UML diagram with class contents, but missing notations.

>Click here [https://app.diagrams.net/#G1mu8srHInQAK5VSvc\\_MwWVZHEMReMJ0yY](https://app.diagrams.net/#G1mu8srHInQAK5VSvc_MwWVZHEMReMJ0yY) <

>If the link above doesn't work, try this (don't forget to click the button "Open with diagrams.net" on above when you click the link)

[https://drive.google.com/file/d/1mu8srHInQAK5VSvc\\_MwWVZHEMReMJ0yY/view?usp=sharing](https://drive.google.com/file/d/1mu8srHInQAK5VSvc_MwWVZHEMReMJ0yY/view?usp=sharing) <

## 4. Task Assignment

Group members & task distribution:

|| **Kutay Şenyiğit** (Firebase / Recover Activity / Login Activity)

- "Login", "signup" and "create an account" activities using Firebase methods
- XML classes for the layout of main activity and messaging activity
- Implementation of message feature
- Sending notification to the phone and also showing them in the app
- Implementation of how other users profiles will be seen by current user
- Chat list feature
- Searching user for sending message using search bar (Implementation and Xml)
- Searching user for review others profile using search bar (Implementation and Xml)
- Showing last activities of users (Implementation and showing them in RecyclerView)
- Showing achievements of users (Implementation and showing them in RecyclerView)

|| **Ömer Oktay Gültekin** (Calendar Activity / Comment Activity/ Create Post Activity / Notifications)

- The settings required for Firebase
- Verify your account and Forgot your passwords activities implementation
- Login optimization and remember me feature

- Revising post view Xml
- Club, Buddy and Stack fragment showing posts and comments (Implementation and revised Xml's)
- In profile, getting user photo and name from database
- In profile settings, changing user photo, changing user password
- Retrieving server time and apply this accurate time on all necessary places
- Bug fixing
- Revising post view activities for buddy and club activities
- Uploading photo feature when user create post or make a comment and showing them in their place

**|| Yunus Eren Türkeri** (XML Classes For User Profile Activities / AppCompatActivity Templates)

- XML classes for main activities
- Button and view templates
- Profile Xml
- In Profile, last activities and achievements window transition
- ListViews for user comments for club and buddy activities
- Medal image designs of achievements

**|| Mert Ünlü** (Creating Post / Filtering Posts / SignUp Activity)

- XML classes for fragment activities
- Custom pop-up menus
- Saving tags, creating posts, filtering posts (Implementation and Xml's)
- Database operations for the actions above
- Revising some Xml's
- Fragment Navigation (Bottom Bar)
- Showing achievements of the user and updating his achievements according to his actions on app

**|| Ege Ipekci** (Comment Details Activity / Direct Messages Activity / FindBuddy Activity)

- XML classes for main activities
- Login-Signup animations
- Showing users tags on user profile
- Saving achievements data on database
- View others tag on their profile
- Adding activities to user calendar

