Bilkent University

Department of Computer Engineering

SENIOR DESIGN PROJECT

Project Name: UniStud

Final Report

Supervisor:

Prof. H. Altay Güvenir

Jury Members:

Prof. Özgür Ulusoy

Prof. Uğur Güdükbay

Innovation Expert:

Melih Gezer

Presented by:

Aurel Hoxha

Albjon Gjuzi

Arba Hoxha

Eniselda Tusku

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project, course CS491/2.

1. Introduction

In the recent years, with the wide popularization of the web, the number of people who actively use different online platforms and applications has increased to an incredible rate, reaching 3.58 billion in 2017 [1]. Among these users' statistics show that the majority of them belong to a group age between 17-28 years old, making these people a very ideal target group for application developers. Students seem to use these platforms excessively but even though here are different applications that help them with several individual needs, there is a lack in the market for application that allow students to access several services in the same time, saving their time, money and energy.

If you are a student enrolled in a university or even an aspiring student, you have to navigate between several applications in order to access services related to the search of the opportunities, tutorials and online courses, finding books or even exploring nearby events. What we aim is to introduce an alternative that allows them to access these services in a single application in a simpler and more effective way.

UniStud will be offered as a web service and as an Android application, in this way reaching a higher number of users and operating better according to the demand of the users. What makes UniStud innovative, is the fact that a student is fed with relevant information without passing from link to another. UniStud will keep users updated with everything that is happening around them. The users can not miss the new opportunities and will easily engage into their educational path. In this report, detailed information about current platforms, proposed system and how it will be implemented will be provided.

1.1 Design Trade-Offs

1.1.1 Security vs. Cost

UniStud collects several data from users, which is composed of multimedia format data, videos or images and private information of different students. Thus, it is crucial for the system to ensure security and keep the information of the users secure. For security, we rely on encrypted databases. Relatedly, the security introduces monetary, time, and labour cost.

1.1.2 Space vs. Speed

The large amount of data introduces difficulties in saving and fetching the data from the servers. Such difficulties are the delays between the operations. The greatest space requirements are introduced by the continuous storage of the livestream tutorial components and its frequent update. This naturally causes an increment in the processing time and slows down the system. The communication between the server and the client for the majority of the data in the system will not slow down the system. However, in case of overloading, to achieve the same result with the livestream component, the quality of the videos will be decreased. All connections and data exchanges with the server will be handled in background threads. This way, we will be able to keep the system fast and highly responsive while managing lots of data at the same time.

1.2 Engineering Standards

In all documentations, UML [2] design principles are used in the description of class interfaces, diagrams, scenarios and use cases, subsystem compositions, and hardware-software components depiction. UML is a commonly used standard that allows simpler description of the components of a software project. With standard UML models, we were able to represent the system structure, software components, and functionalities.

1.3 Interface Documentation Guidelines

All the classes that are going to be used in the system will be documented following some conventions. Class, variables and method names will use ‘Camel-Case’ to diminish the difficulty in understanding the code. Every class will follow the same design pattern where the class name comes first, the attributes follow, and finally the methods are listed. The detailed outline is provided below:

1.4 Definitions, Keywords, Acronyms and Abbreviations

Career Opportunities: Browse section for different universities and internships.
UniTrade: The section where students will be able to sell/buy/loan study materials.
UniStream: The section where students will be able to watch or create tutorials.
UI: User Interface
API: Application Programming Interface
HTTP: Hypertext Transfer Protocol
TCP: Transmission Control Protocol
Client: The part of the system that user interacts with.
Server: The part of the system that responds to client’s requests. It is responsible for data management, API interactions and logical operations.

2. Final Architecture and Design

2.1 Subsystem Decomposition

2.1.1 Client

The client service corresponds to both mobile and web applications, because all functionalities will be implemented in both platforms. The client is the presentation layer of our system. Most of the requests that will be made will be sent to the server and data will be returned and displayed to the user accordingly. Client is responsible for managing users' operations on the system, presenting the data from the server to the user and also notifying the user when it is necessary. Client subsystem includes Presentation Tier and Control Tier.

2.1.1.1 Presentation Package

Presentation Tier is responsible for all of the user interface interactions and it uses Control Tier in order to communicate with the server.

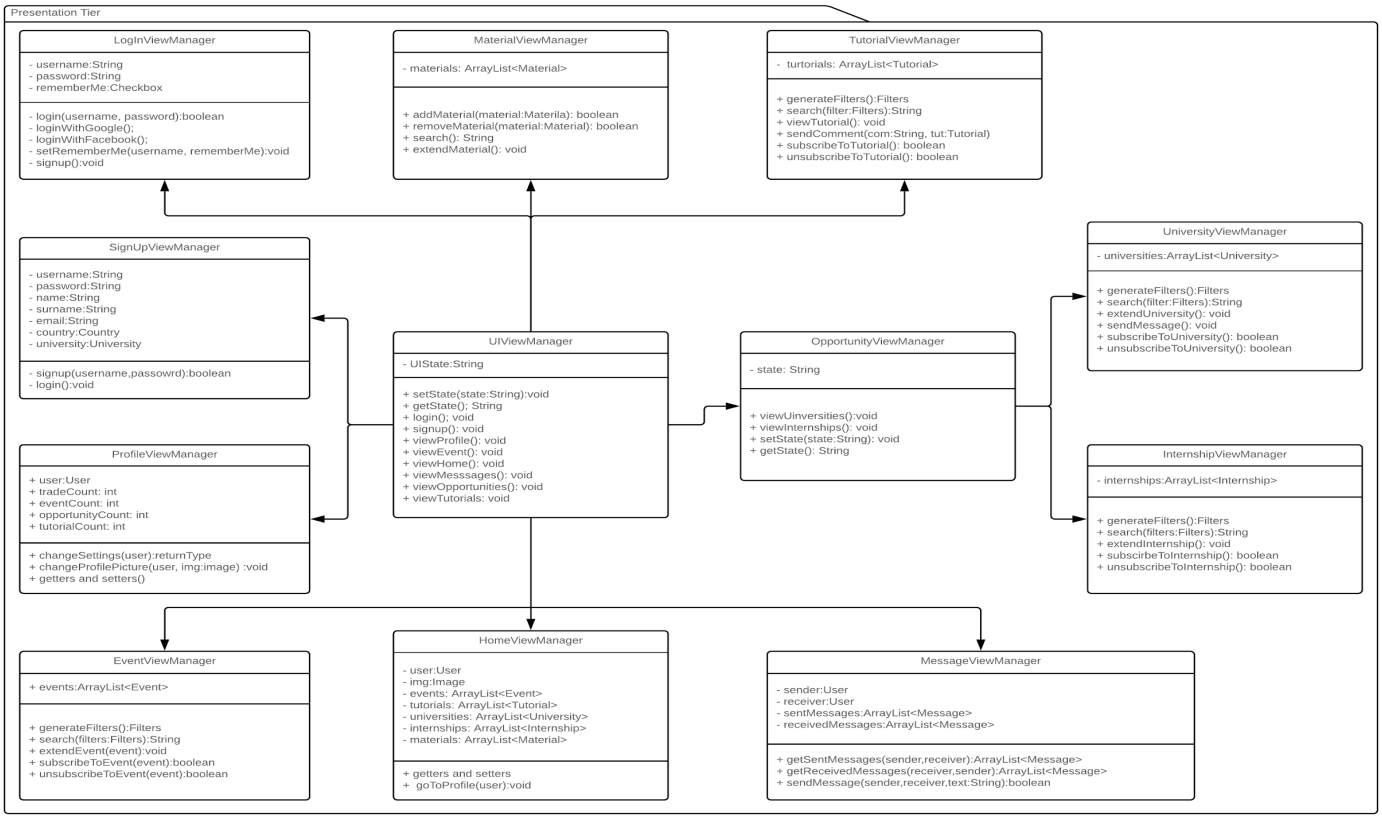


Figure 1 – Presentation Class Diagram

- **UIViewManager:** Class that control all views of the system and coordinates the change from one to another.
- **LoginViewManager:** This class handles the first page operations and UI that the user will see upon opening the system.
- **SignupViewManager:** This class handles the Sign-Up operations and UI for users that are not yet registered to our system.
- **ProfileViewManager:** This class handles all operations related to UI for a user's account and settings and the UI.
- **MessageViewManger:** This class handles all operation related to UI for a user sent and received messages.
- **EventViewManager:** This class handles all event related operations and the UI.
- **HomeViewManager:** This class handles all Home Page related operations and the UI.
- **MaterialViewManager:** This class handles all Study Material related operations and the UI.
- **OpportunityViewManager:** This class can direct user to either University or Internship view and coordinates movement between those two.
- **InternshipViewManger:** This class handles all Internship related operations and the UI.
- **UniversityViewManager:** This class handles all Universities related operations and the UI.
- **TutorialViewManager:** This class handles all Tutorial related operations and the UI.

2.1.1.2 Controller Package

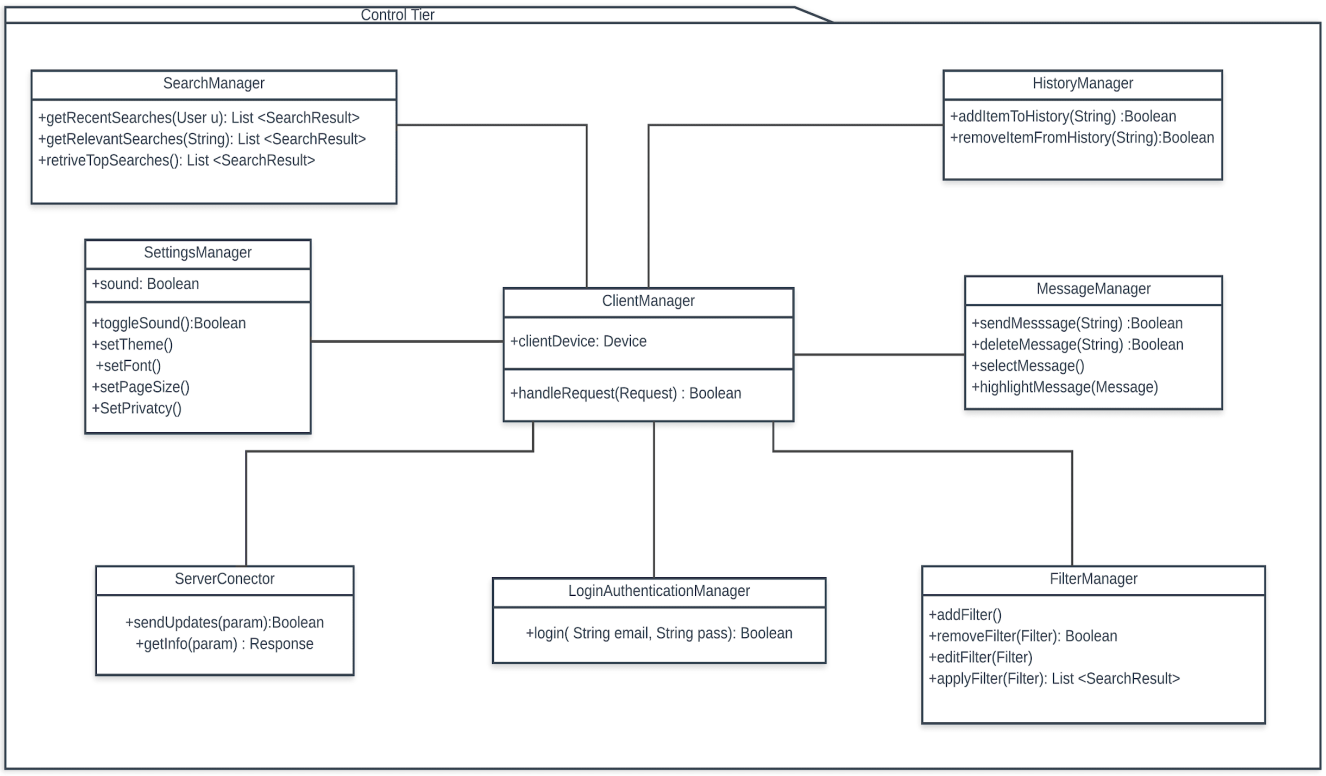


Figure 2 – Controller Class Diagram

- **ServerConnector:** This class handles the communication between the client and the servers.
- **SettingsManager:** This class updates the settings of the user according to their changes and preferences.
- **SearchManager:** Class that stores the previous searches in order to help the user and also directs the search to the server to get the results.
- **LoginAuthenticationManager:** Class that handles login operations.
- **FilterManager:** Class that handles filters for all types of searches and applies the search according to them.
- **MessageManager:** Class that handles message sending and receiving operations.
- **HistoryManager:** Class that saves and updates the history according to user operations and interaction with the system.

2.1.2 Server

Server is a crucial part of our system that will be responsible for the all the interactions that the user will have with the system. The tutorials that will be offered will be recorded on the client side and delivered to the server. The server will receive this data and will make it accessible for the other users in the platform. It will also handle the information that will be entered regarding the other categories such as opportunities, trading and events and will make the necessary adjustments of this information accordingly.

2.1.2.1 Logic Tier

The Logic tier is the application layer responsible for the control of the flow of information between presentation layer and data layer. It accommodates all the heavy operations the application needs to handle.

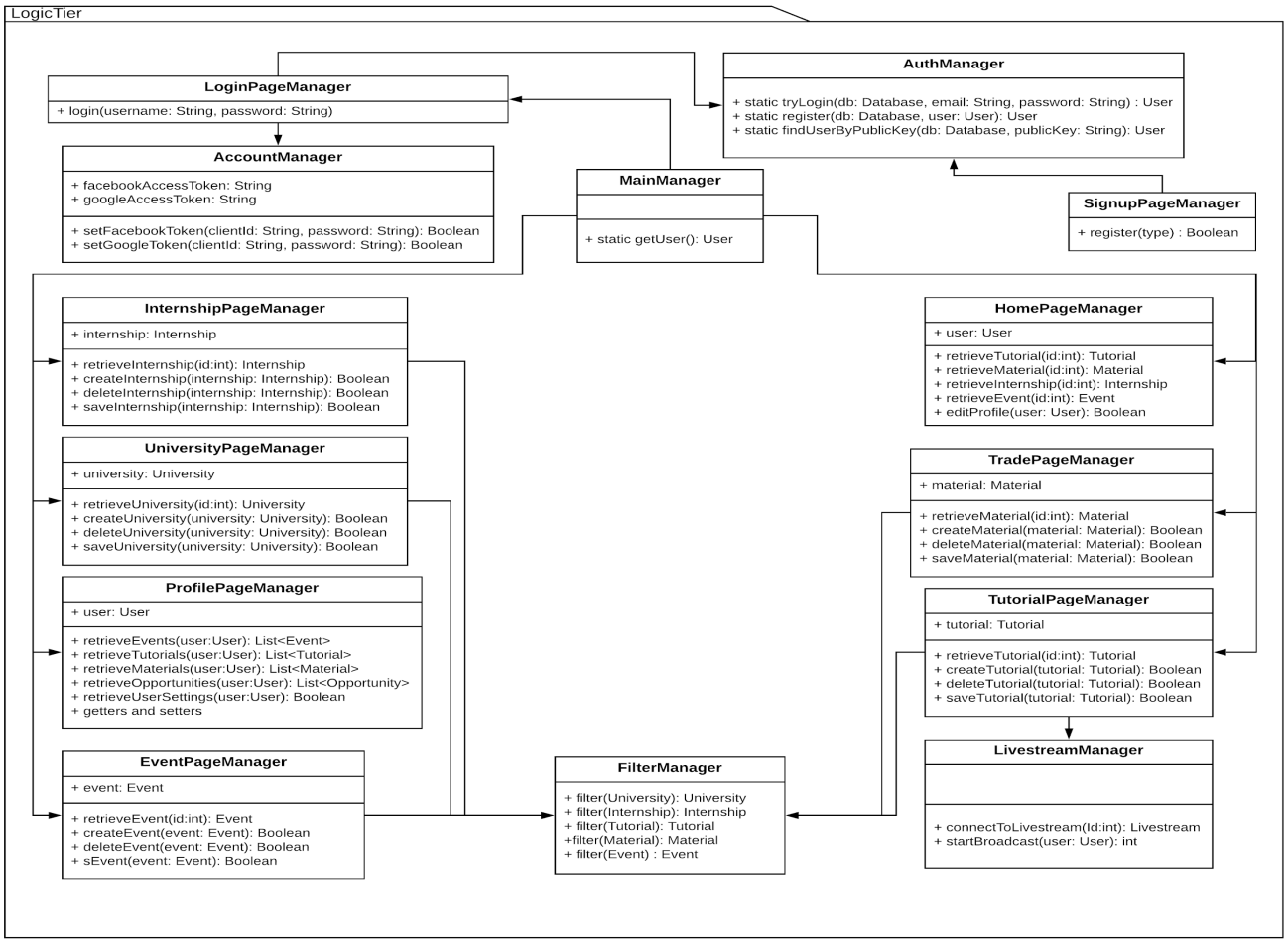


Figure 3 – Logic Class Diagram

- **AccountManager:** The main function of this class is to maintain the users' account.
- **AuthManager:** This class is used to save, retrieve and update information about the users from the database.
- **MainManager:** Fundamental class that will be responsible for accessing the services provided by the application. It will gather all the information and deal with the interactions of the user leading to each of these individual services.
- **HomePageManager:** Class that will be responsible for storing information and directing the user to specific services.
- **TutorialPageManager:** This class is responsible for the tutorials and will have all the necessary information regarding them. It will be responsible for the interaction of the user with the tutorials both in offering and accessing.
- **InternshipPageManager:** This class is responsible for the internships and will have all the necessary information regarding them. It will be responsible for the interaction of the user with the internships.
- **UniversityPageManager:** This class is responsible for the university related search and will have all the necessary information regarding them.
- **EventPageManager:** This class is responsible for the events related and will have all the necessary information regarding them. It will deal with the interactions on both offering and accessing an event.
- **TradePageManager:** This class that is responsible for the displaying and trading of all of the academic materials and will be responsible for the interactions of the user with these materials.
- **LogicPageManager:** This class is responsible for handling the Login process of the user and dealing with the provided data.
- **SignUpPageManager:** This class is responsible for handling the SignUp process of the user, dealing with the provided data and managing the interactions to make the user part of the platform.
- **ProfilePageManager:** This class is responsible for the user information and will store all the necessary data.
- **FilterManager:** This class is responsible for filtering all the data content that is found in the system. It will use the data according to the particular type.

2.1.2.2 Data Tier

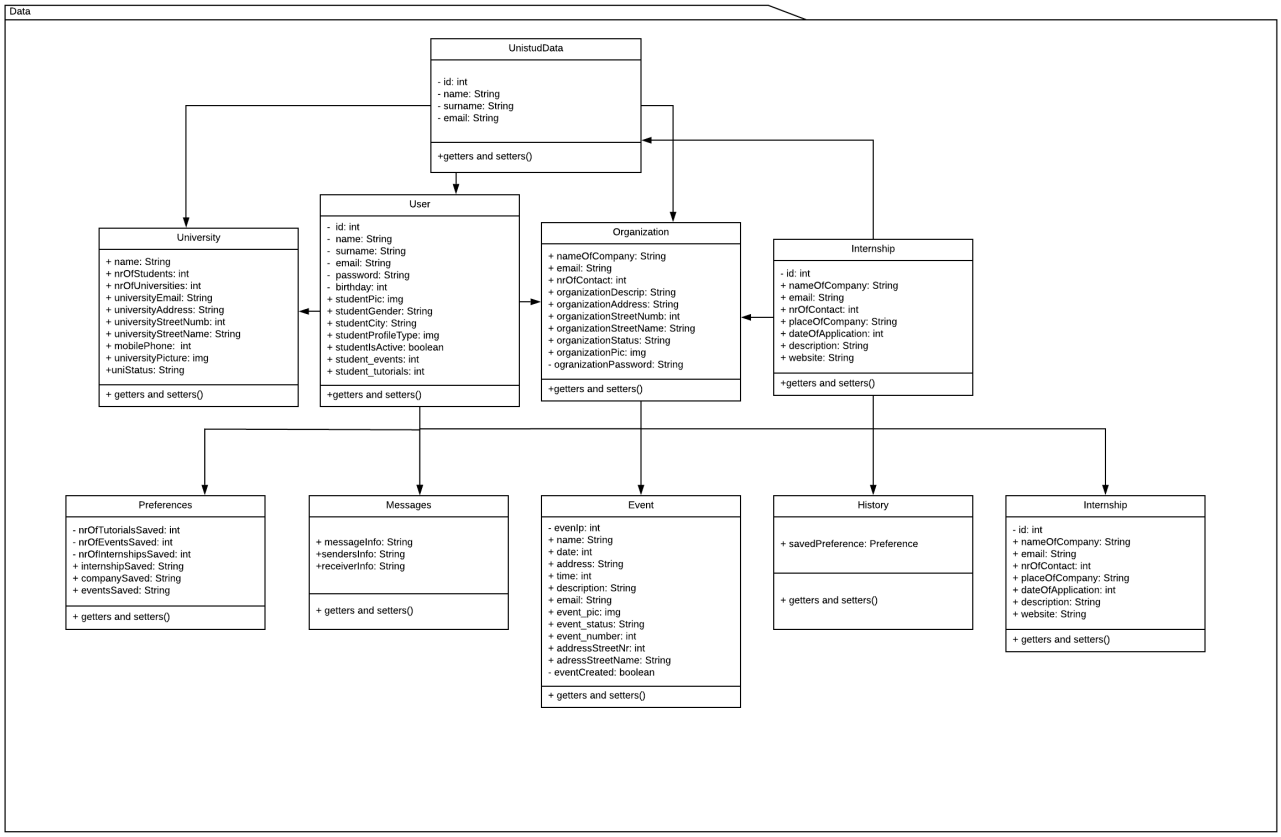


Figure 4 – Data Class Diagram

- **User:** Data class representing users.
- **University:** Data class representing all the universities in UniStud.
- **Internship:** This class contains the information about the internships offered by the companies in the platform.
- **Organization:** This data class concerns with the information connecting the other data classes such as preferences, messages, event, history and tutorial.
- **Preferences:** This class handles all the user's preferences by the filters provided in the platform.
- **Messages:** This class represents the data concerning the messages between the users of the platform.
- **Event:** This data class represents all the events provided and advertised in the platform.
- **History:** The data class which handles the information of the user's history.
- **Tutorial:** This data class represents all the available tutorials in the platform.

2.2 Hardware Software Mapping

2.3 App Design

2.3.1 Live Broadcasting

UniStud is a platform that allows users to stream live. Broadcasting will be used with the sole purpose of streaming tutorials for different courses or topics. The app will use the Wowza GoCoderSDK in order to be able to stream for every android mobile device. WowzaStreamingEngine will be used for the live streams and one of the student's laptop will function as the server for incoming and outgoing livestreams. The streaming engine provides integration with the SDK in order to provide low latency live streams.

2.3.2 Livestream and Video Playing

Unistud also allows the users to save the tutorials live stream for latter usage. Students who miss the live stream can follow it later. ExoPlayer library is used for both livestream and video playing. The only difference is in the media source they use. Live stream uses Apple HLS playback and therefore requires HlsMediaSource, meanwhile the .mp4 file formats of saved videos require ExtractorMediaSource.

2.3.3 Find Universities, Internships, Events and Items

The main aim of UniStud is to help students find all necessary components in a single application. Therefore, we have provided them the chances of:

1. Searching for universities and contact the students already enrolled there to get real information about life in university, difficulty of course and life costs.
2. Searching for internships.
3. Searching for nearby events organized by companies of clubs.
4. Searching for items they want to buy, or even selling some items they possess like books, notes, devices or any kind of item they want to.

3. Algorithms

3.1 Video Quality Adjusting

For the live stream tutorials, we collect some information from the user in real time in order to check their internet connection and its speed. Based on these data we change adjust the quality of the live stream or of the video tutorial that is saved in the database.

3.2 Garbage Data Check

Our application is constantly getting inputs from the users, starting from the registration up to selling a book. In most of them we are using different algorithms to check if the data is valid or not. For example, we are checking if the email is valid or the password contains at least 8 characters and is strong enough, which means it contains at least one upper case, one lower case, one digit and one special character. We know this might be annoying to our users, but this it totally for their own data safety.

3.3 Suggested User Ranking

In the list of universities, each university also has a list of the students from the platform that are enrolled there. Students are displayed according to a ranking of number of users they reply to and how fast they reply to them. The faster and the more students one replies to, the higher he appears in that list.

4. Impact of Engineering Solution

4.1 Global Impact

Unistud offers a very useful and collaborative platform in the market that assists the student's needs in multiple areas. By its usage students are able to search for different internships and universities, search for events around them, buy and sell different items and offer and receive tutorials. Having such an integrated platform will have a large impact on the lives of students making their lives easier and making the accessing of services efficient.

4.2 Social Impact

The social impact of UniStud is concentrated in the target group of the students and it offers an opportunity for them not only to increase their network but also to obtain different services. The aspect of the application that provides a search platform for different opportunities and specifically for the universities is associated with a list of the students who have attended the same university and want to share their feedback with others. In this way, the students have a way to get real time information from people all around the world. Even though UniStud is not a social platform, using it has a social impact on its users because it brings them closer together. Also, the Tutorials part of the application builds an interaction bridge between all the users in the system. Since every tutorial is shared, each viewer can join and discuss related to a specific topic. In this way affecting the participants socially and at the same time professionally. The other aspects of UniStud such as trading and events also contribute in an overall social impact since it allows students to connect with others in order to buy and sell items or to be part of the events around them. In this way, it is possible for students to strengthen their community, communicate, share and help each other more. In this way UniStud becomes a real social influencer for the students all around the world and it changes the way in which their communities function.

5. Tools and Technologies Used

This section is decided to explanation of the tools and technologies we have used, what they are and how we used them in our project.

5.1 Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. [3]

5.2 Google Cloud

Platform, powered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. We use the cloud to save the videos generated from the live stream tutorials. [4]

5.3 Firebase

Firebase is a mobile and web application development platform. Because of the ease of integration and usage, neat interface and speed it was used as UniStud database. [5]

5.4 Android

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. [6]

5.5 Wowza Streaming Engine

Wowza Stream Engine is a unified streaming media server software developed by Wowza Media Systems. The server is used for streaming of live and on-demand video, audio, and rich Internet applications over IP networks to desktop, laptop, and tablet computers, mobile devices, IPTV set-

top boxes, internet-connected TV sets, game consoles, and other network-connected devices. The server is a Java application deployable on most operating systems. [7]

5.6 Wowza GoCoder SDK

It is a cross-platform, extensible API for capturing and streaming live audio and video in iOS and Android mobile apps. The SDK supports a wide range of iOS and Android devices as well as 4K video resolution and other customizable encoder settings for low latency streaming of videos. [8]

5.7 Picasso

A powerful image downloading and caching library for Android Introduction Images add much-needed context and visual flair to Android applications. Picasso allows for hassle-free image loading in your application, often in one line of code! [9]

5.8 Facebook and Google SDK

We integrated Facebook and Google API's to ease the process of signing up by getting the information we need directly from user's Facebook or Gmail accounts. [10]

5.9 ExoPlayer

ExoPlayer is an application level media player for Android. It provides an alternative to Android's MediaPlayer API for playing audio and video both locally and over the Internet. ExoPlayer supports features not currently supported by Android's MediaPlayer API, including DASH and SmoothStreaming adaptive playbacks. [X + 8]

5.10 Apache Web Server

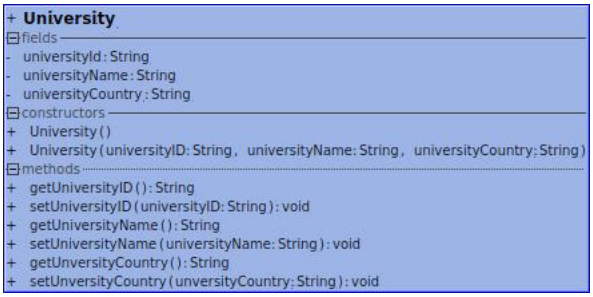
The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. We have used this tool to make the live stream server laptop also the cloud for the videos. As videos are saved in the laptop, they can be played latter on by using Apache Web Server. [X + 9]

6. Class Diagram

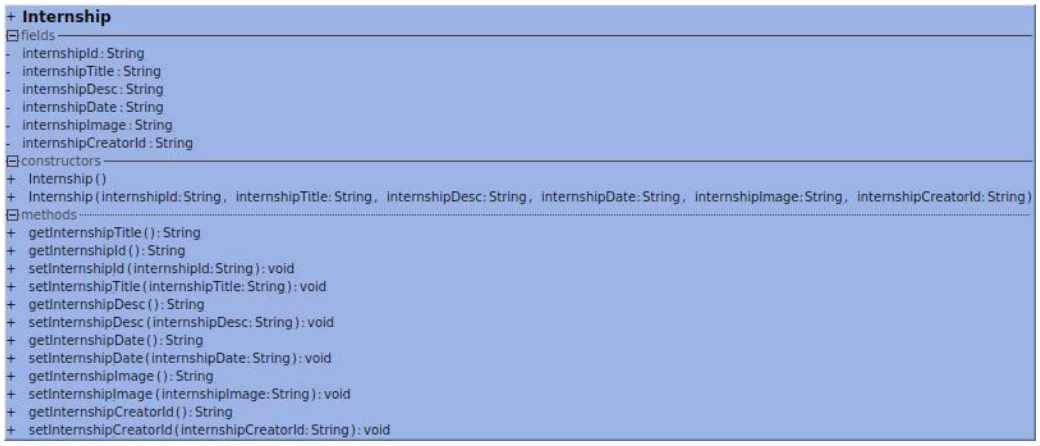
6.1 Student

6.2 Organization

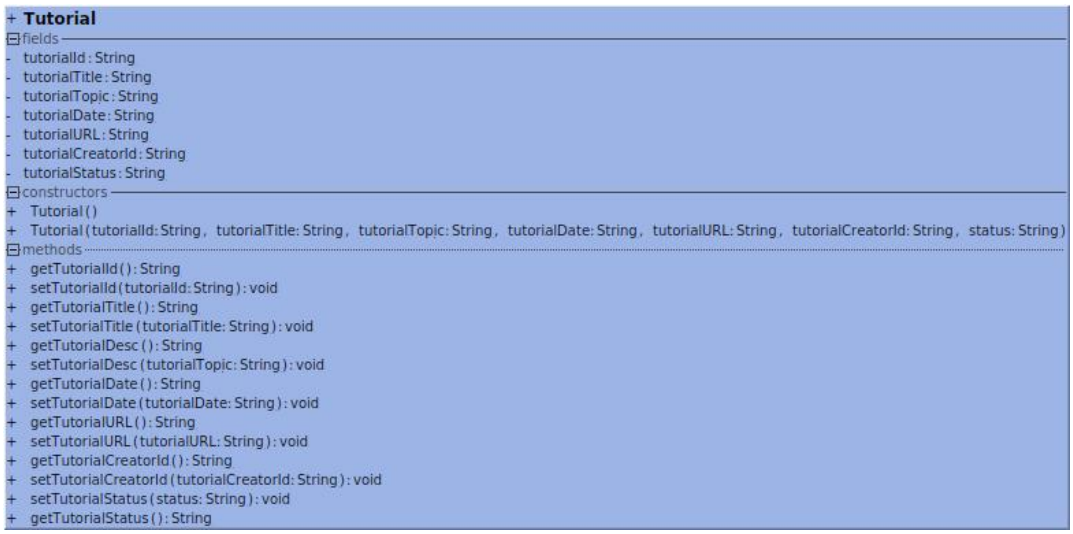
6.3 University



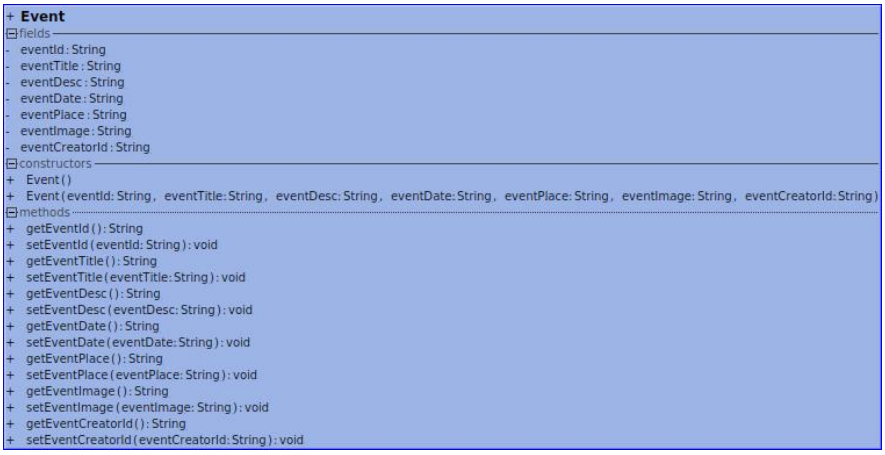
6.4 Internship



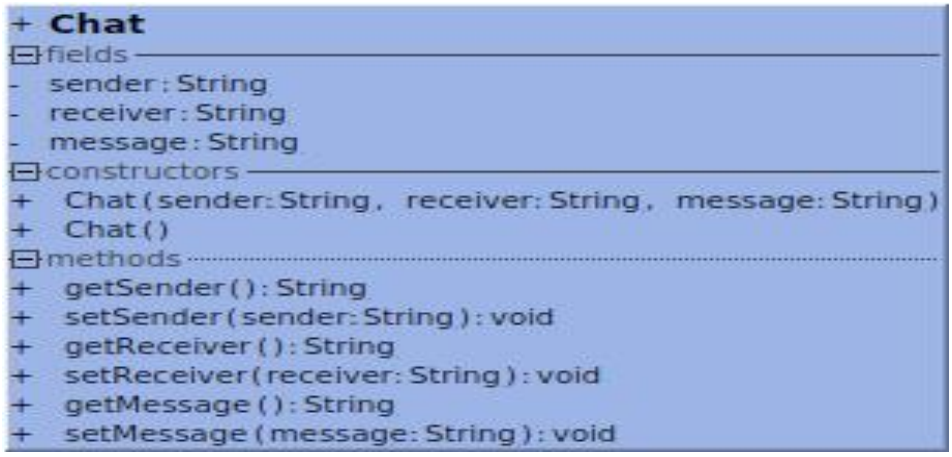
6.5 Tutorial



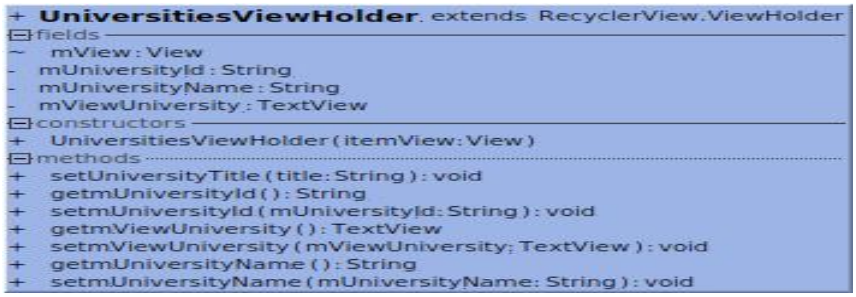
6.6 Event



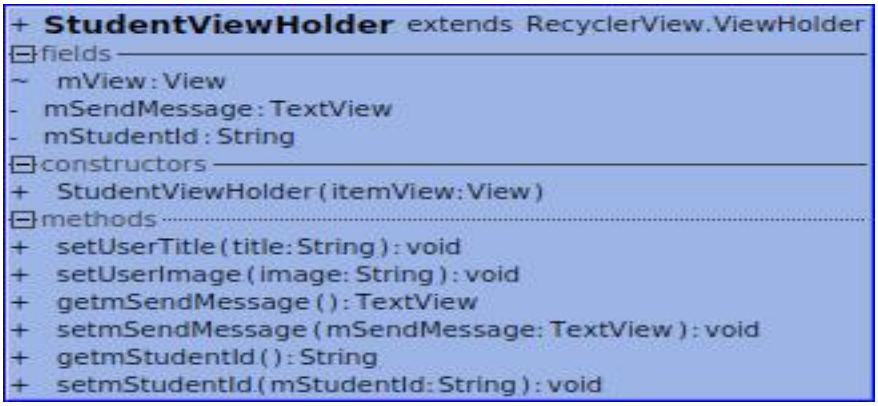
6.7 Chat



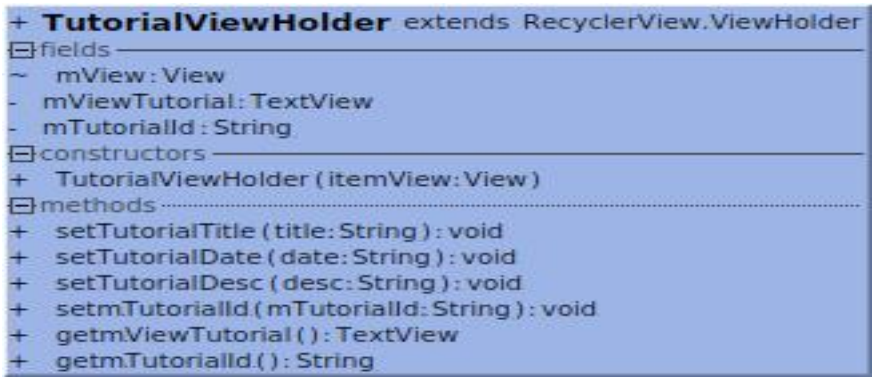
6.8 UniversityViewHolder



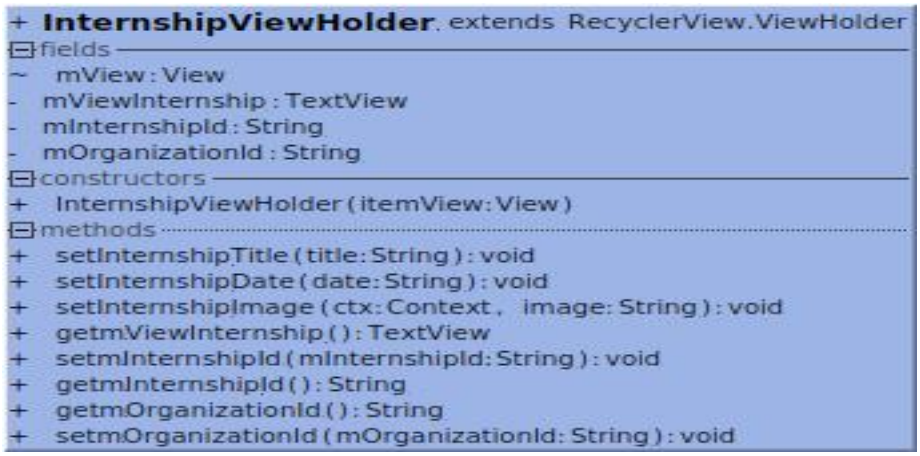
6.9 StudentViewHolder



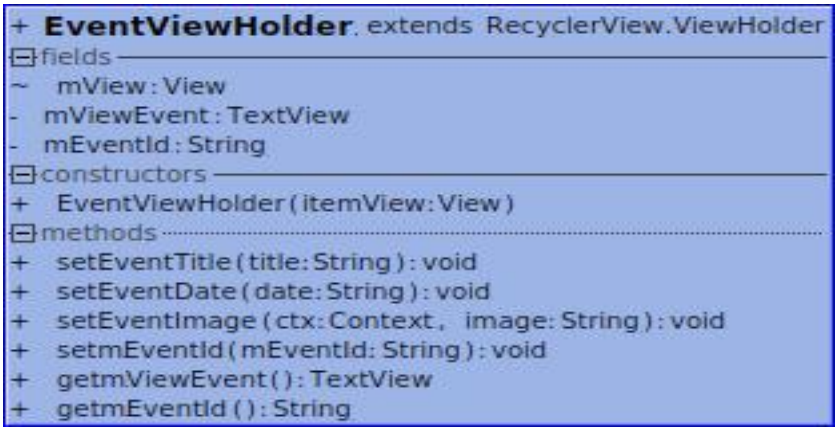
6.10 TutorialViewHolder



6.11 InternshipViewHolder



6.12 EventViewHolder

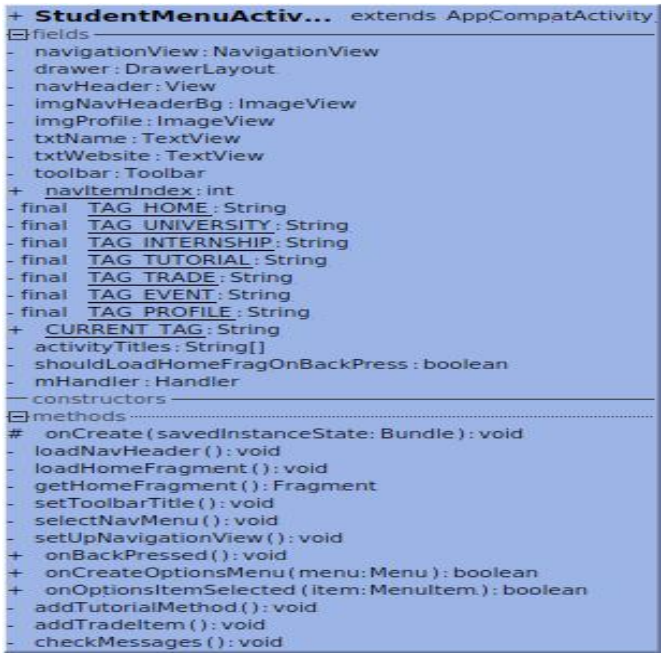


6.13 User Adapter

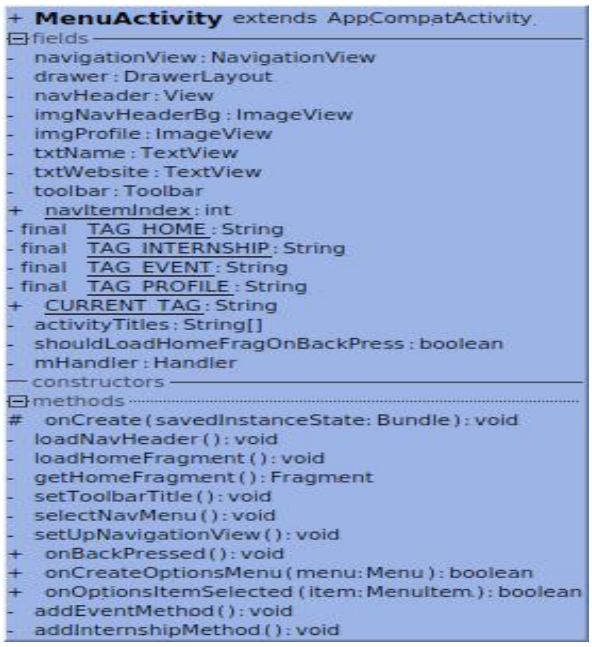
6.14 Message Adapter

6.15 Main Activity

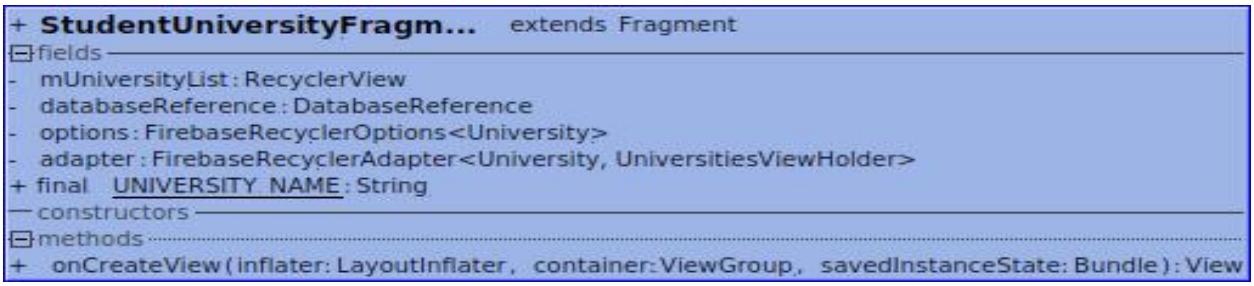
6.16 Student Menu Activity



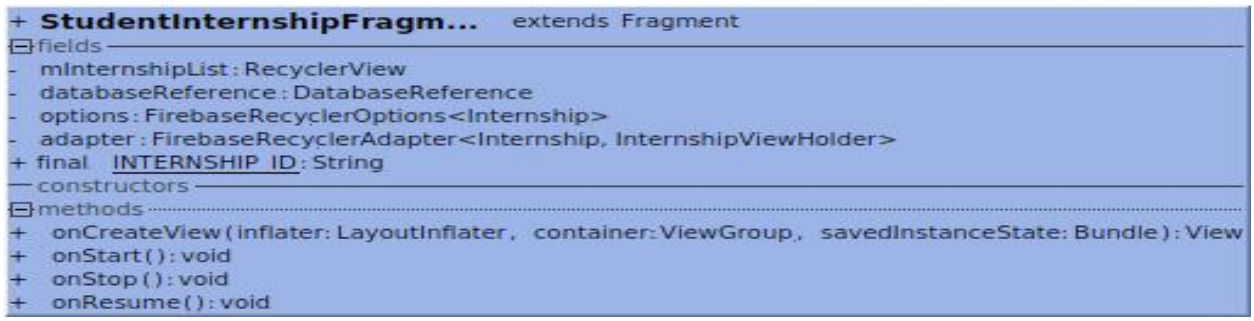
6.17 Organization Menu Activity



6.18 Student University Fragment

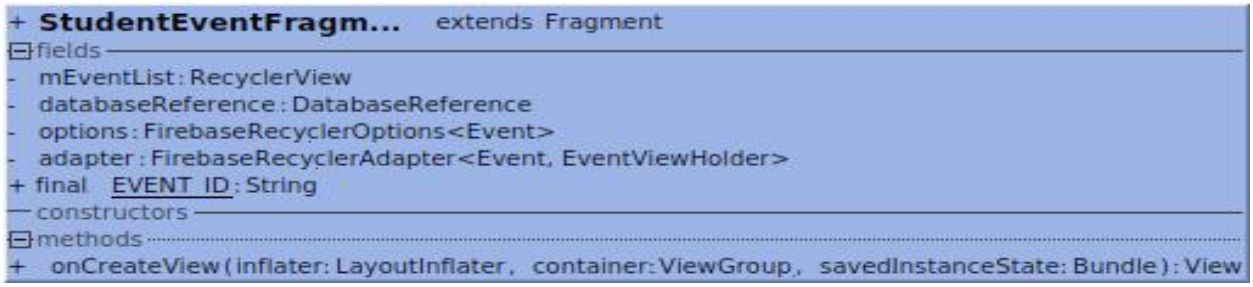


6.19 Student Internship Fragment

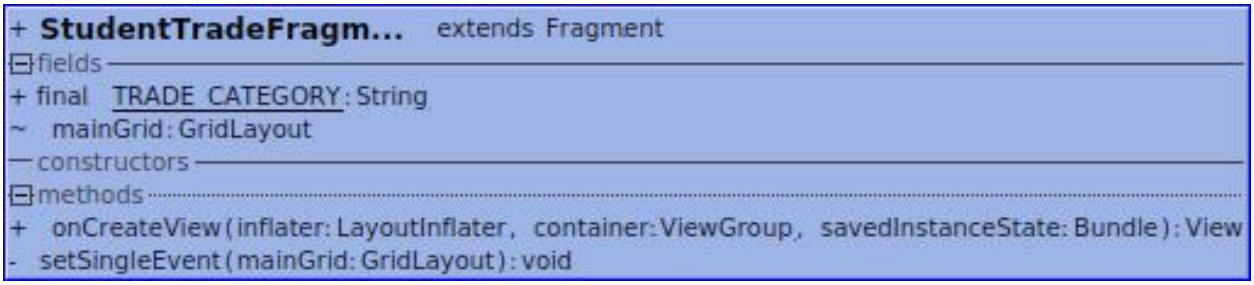


6.20 Student Tutorial Fragment

6.21 Student Event Fragment



6.22 Student UniTrade Fragment



6.23 Student Profile Fragment

6.24 Organization Internship Fragment

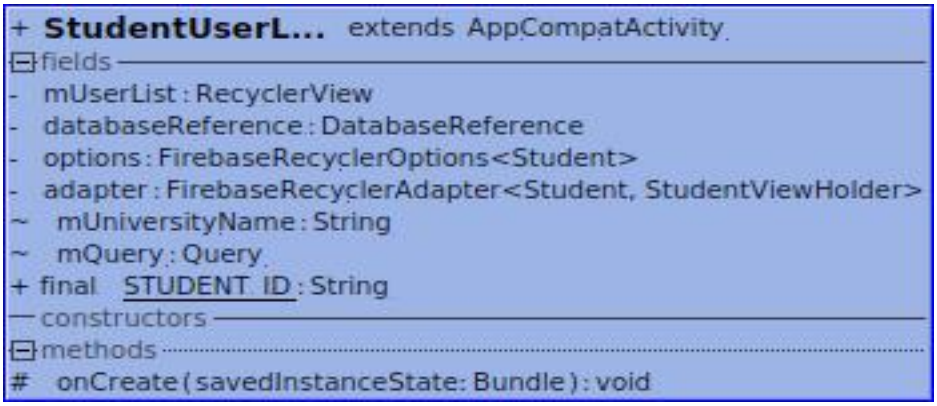
6.25 Organization Events Fragment

6.26 Forget Password Activity

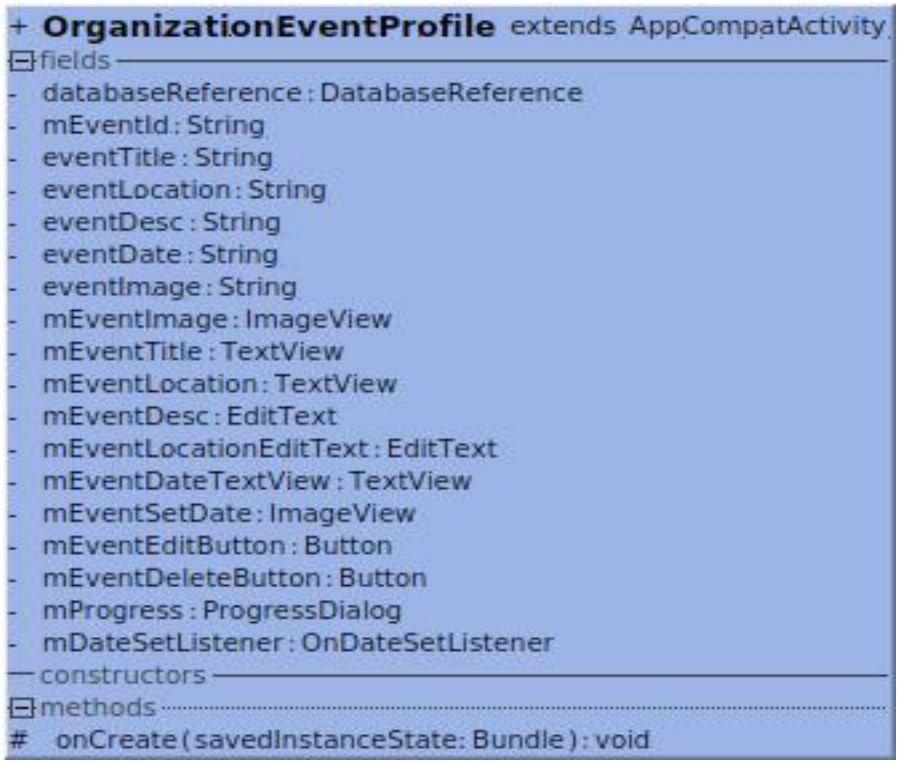
6.27 Sign Up Activity

6.28 Organization Register Activity

6.29 Student User List

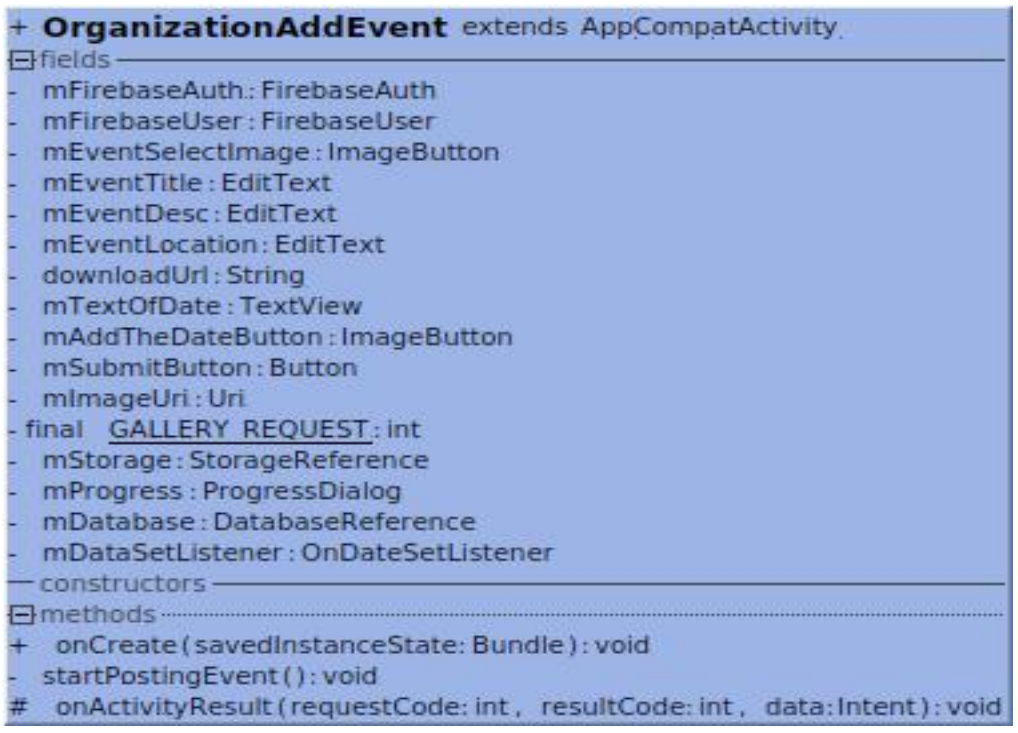


6.30 Organization Event Profile

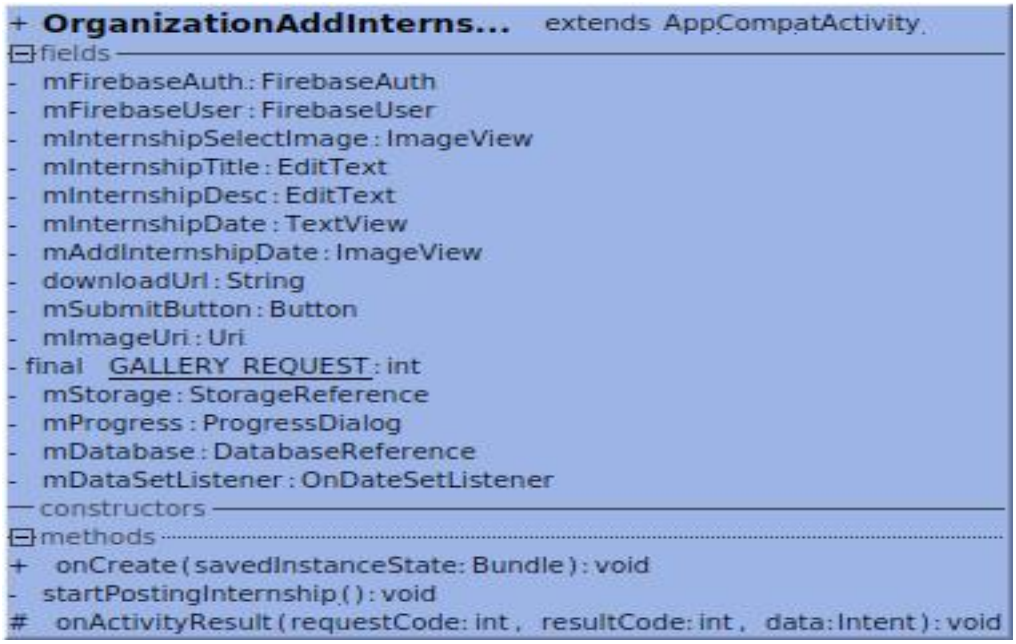


6.31 Organization Internship Profile

6.32 Organization Add Event

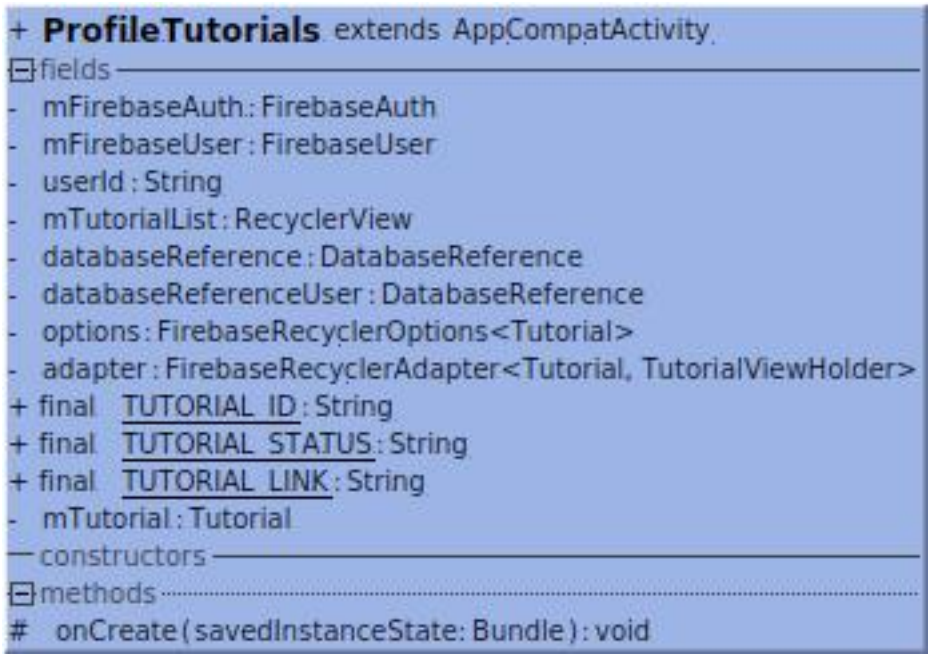


6.33 Organization Add Internship



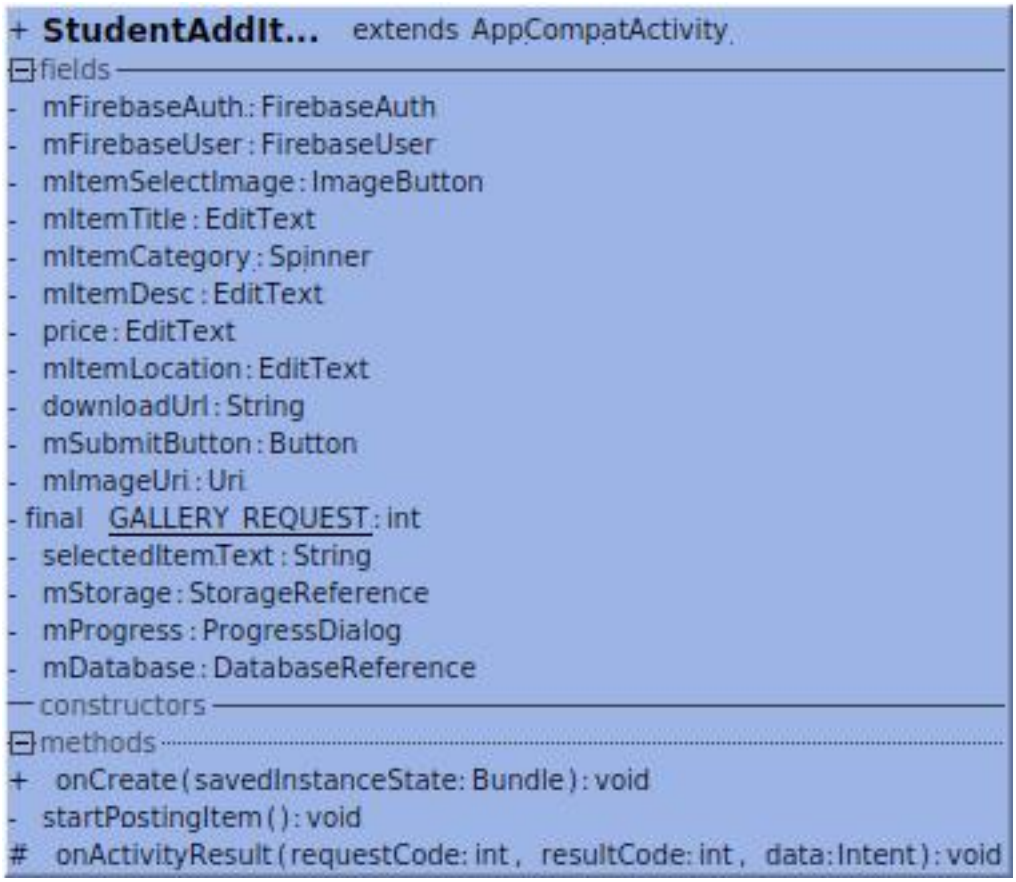
6.34 Profile Events

6.35 Profile Tutorials



6.36 Profile Items

6.37 Student Add Item



6.38 Student Trade Window

6.39 Trade Adapter

6.40 Message Activity

6.41 My Messages

6.42 Object Adapter

6.43 Tutorial Live Stream

6.44 Play Video

7. References

[1] INTERNET USAGE STATISTICS The Internet Big Picture. (n.d.). 2018 https://www.internetworldstats.com/stats.htm. Accessed: 2018-10-14.

[2] “UML - Basics, ibm. http://www.ibm.com/developerworks/rational/ library/769.html.” Accessed: 2018-11-04.

[3] Git. https://git-scm.com/. Accessed: 2019-05-05.

[4] Google cloud. https://cloud.google.com/. Accessed: 2019-05-05.

[5] Firebase. https://firebase.google.com/. Accessed: 2019-05-05.

[6] Android. https://www.android.com/. Accessed: 2019-05-05.

[7] Wowza streaming engine. https://www.wowza.com/. Accessed: 2019-05-05.

[8] Wowza GoCoder SDK. https://www.wowza.com/. Accessed: 2019-05-05.

[9] Picasso. https://square.github.io/picasso/ . Accessed 2019-05-05.

[10] Facebook/Google API. https://facebook.com/, https://google.com/. Accessed 2019-05-05.

[11] ExoPlayer. https://github.com/google/ExoPlayer. Accessed 2019-05-05.

[12] Apache Web Server. https://httpd.apache.org/. Accessed 2019-05-05.