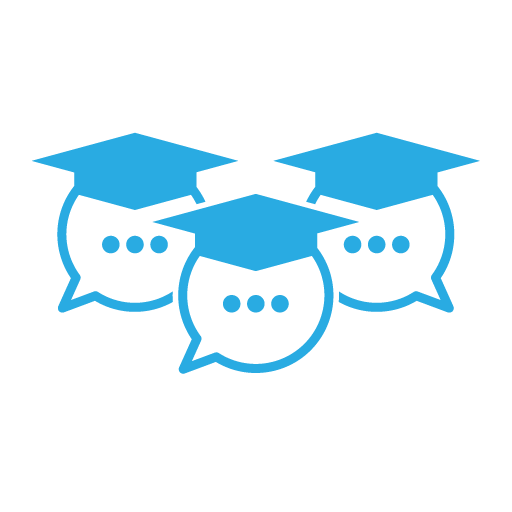
**Arab Open University- Riyadh**

Faculty of Computer Studies

Information Technology and Computing Department

**UNIVERSITY GROUP CHAT APPLICATION**

by

|  |  |
| --- | --- |
| Student ID | Student Name |
| 101401739 | Amar Mohammad Zieb Almotlek Almandil |
| 101300367 | Ali Najdat Almohammed Jawish |

TM471: Final Year Project, December 2019

Supervisors:

**Dr. Basil Kasasbeh  
Dr. Khaled Suwais**

# Abstract

Each day we care more about what information we share online and how other may use it against our will, the idea of this app was to solve two major issues students face every day, the need of a fast communication method with classmates and tutor, while not compromising privacy or share personal information.

Students still use unofficial instant messaging applications to stay updated about the course, while the official communication application that has been provided by the university does not offer the main feature students need which is fast chatting, push notification and mobility.

The Idea of the application is to create a mobile application that fills the gaps of current systems and make sure that the new system fits the student’s needs.

The system has been designed to provide all features current systems offer plus the feature that a student and an educational facility requires. This way we get all the features that help improve the learning and standardizes the communication method.

# Acknowledgments

        $reader = new Reader;

        if ($reader->type == 'admin') {

            return 'I could not have done this without your help and support';

        } else {

            return 'I would like to thank myself for doing this wonderful job';

        }

If words are considered a symbol of approval or a token of appreciation, then let the words play the heralding role in expressing my gratitude. Thank you for helping us to create a complete project that started with an idea.

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible and whose constant guidance and encouragement crown all efforts with success. We are grateful to our parents who stood by from the beginning, our project supervisors (Dr. Basel and Dr. Khalid) for their guidance, inspiration and constructive suggestions that were so helpful to us in the preparation of this project. We would also like to thank our colleagues who have helped in the successful completion of the project.

Contents

[Abstract III](#_Toc38055001)

[Acknowledgments IV](#_Toc38055002)

[Table of Figures VI](#_Toc38055003)

**CHAPTER ONE: INTRODUCTION**

[1.1 Overview 2](#_Toc38055004)

[1.2 Aims of The Project: 3](#_Toc38055005)

[1.3 Problem statement 3](#_Toc38055006)

[1.4 Project Scheduling 4](#_Toc38055007)

[1.4.1 Data Collecting 4](#_Toc38055008)

[1.4.2 Analyzing data 4](#_Toc38055009)

[1.4.3 Design 4](#_Toc38055010)

[1.4.4 Development 4](#_Toc38055011)

[1.4.5 Test 4](#_Toc38055012)

[1.5 Document Structure 6](#_Toc38055013)

**CHAPTER TWO: RELATED SYSTEMS & COMPARATIVE ANALYSIS**

[2.1 History of Messaging 8](#_Toc38055014)

[2.2 Comparing Current Systems 9](#_Toc38055015)

[2.2.1 WhatsApp 9](#_Toc38055016)

[2.2.2 Facebook Groups 9](#_Toc38055017)

[2.2.3 Telegram 10](#_Toc38055018)

[2.2.4 LMS Forum 10](#_Toc38055019)

[2.3 Stakeholders 11](#_Toc38055020)

[2.4 Applications 11](#_Toc38055021)

**CHAPTER THREE: REQUIREMENTS & ANALYSIS**

[3.1 Requirements and Features 13](#_Toc38055022)

[3.2 Functional and Non-Functional Requirements 13](#_Toc38055023)

[3.2.1 Functional Requirements 13](#_Toc38055024)

[3.2.2 Non-functional Requirements 14](#_Toc38055025)

[3.3 Project Diagrams and Charts 15](#_Toc38055026)

[3.3.1 Use-Case Table 15](#_Toc38055027)

[3.3.2 Use-Case Diagram 19](#_Toc38055028)

[3.3.3 Flowchart Diagram 20](#_Toc38055029)

**CHAPTER FOUR: DESIGN, IMPLEMENTATION, AND TESTING**

[4.1 Just an Idea! 22](#_Toc38055030)

[4.2 Scrum Framework 22](#_Toc38055031)

[4.2.1 Collecting Requirements 23](#_Toc38055032)

[4.2.2 Product Backlog 24](#_Toc38055033)

[4.2.3 Sprint Planning 25](#_Toc38055034)

[4.2.4 Daily Scrum 25](#_Toc38055035)

[4.2.5 Sprint Review 25](#_Toc38055036)

[4.2.6 Sprint Retrospective 25](#_Toc38055037)

[4.3 Implementation 26](#_Toc38055038)

[4.3.1 The Database 26](#_Toc38055039)

[4.3.2 Web-based Application 27](#_Toc38055040)

[4.3.3 Mobile Application 30](#_Toc38055041)

[4.3.4 Connecting it all together 43](#_Toc38055042)

[4.3.5 Alternatives and comparing 43](#_Toc38055043)

[4.4 Testing 44](#_Toc38055044)

[4.4.1 Testing login and register 44](#_Toc38055045)

[4.4.2 Testing Roles 46](#_Toc38055046)

[4.4.3 Testing API 46](#_Toc38055047)

[4.4.4 Testing group chat 46](#_Toc38055048)

**CHAPTER FIVE: RESULTS AND DISCUSSION**………………………………………….47

**CHAPTER SIX: CONCLUSIONS**……………………………………………………………50

[References 52](#_Toc38055049)

[Appendix: 54](#_Toc38055050)

# Table of Figures

[Figure 1 - Gantt Chart of the project time table with milestones periods of each task 5](#_Toc38056299)

[Figure 2 - Gantt Chart for Collect data & analyzing Data milestones 5](#_Toc38056300)

[Figure 3 - Gantt Chart for Design Milestone 5](#_Toc38056301)

[Figure 4 - Gantt View for Development Milestone 5](#_Toc38056302)

[Figure 5 - Gantt View for Testing Milestone 6](#_Toc38056303)

[Figure 6 - Use-Case Table of The System 15](#_Toc38056304)

[Figure 7 Use Case Diagram That Shows How The System Works 19](#_Toc38056305)

[Figure 8 – Flowchart diagram that shows how the system works 20](#_Toc38056306)

[Figure 9 A simplified overview of the events and cycles of scrum. 23](#_Toc38056307)

[Figure 10 Example of User Story Card 24](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056308)

[Figure 11 Empty User Story Card 24](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056309)

[Figure 12 Example of the product backlog 25](#_Toc38056310)

[Figure 13 Database Diagram 26](#_Toc38056311)

[Figure 14 the admin dashboard 28](#_Toc38056312)

[Figure 15 create group form 28](#_Toc38056313)

[Figure 16 Group list of students 29](#_Toc38056314)

[Figure 17 Students List 29](#_Toc38056315)

[Figure 18 welcome screen of the app 31](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056316)

[Figure 19 second welcome screen 32](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056317)

[Figure 20 register screen for new users 33](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056318)

[Figure 21 login screen 34](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056319)

[Figure 22 Verification screen 35](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056320)

[Figure 23 intro screen with information 36](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056321)

[Figure 24 list of groups screen 37](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056322)

[Figure 25 broadcast messages screen 38](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056323)

[Figure 26 in group chat screen 39](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056324)

[Figure 27 notification screen 40](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056325)

[Figure 28 side menu 41](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056326)

[Figure 29 profile screen 42](file:///C:\Users\Ali\Desktop\TM471-%20Report.docx#_Toc38056327)

[Figure 30 unauthorized attempt test 44](#_Toc38056328)

[Figure 31 empty field test 45](#_Toc38056329)

[Figure 32 password less than 8 charachters attempt 45](#_Toc38056330)

[Figure 33 password not match attempt 46](#_Toc38056331)

Chapter One: introduction

# 1.1 Overview

It was the beginning of a new semester, at the end of the first lecture our new tutor asked a guy to create a WhatsApp group so all students in that class could be into chat and send updates about the course, and the tutor said to the student to not include him in the WhatsApp Group because he does not want his mobile number to be available to all students and I had the same concept to not share my mobile number with others.

From this moment we had an idea, what if there was an application that offers the same features; instant messaging, media sharing and group chatting that WhatsApp offers, it was maintained by the university, the tutor is a member of this group and only the authorized people can join this group without the need to share our mobile number.

While the mobile number being too much exposed to privacy, we had to come up with a unique identifier, which lead us to choose from major standers like username, email, student ID.

By comparing each method, using random usernames (e.g. Red1991Dragon) does not follow the rule of being an official method of communication for an educational institute. Emails on the other hand is good to be a unique identifier, but some students may use personal emails to register or some unauthorized persons may join the chatting group.

This left us with the Student ID, which has the following characteristics, it is considered as an official identifier for the student, it is easy to remember, only the educational institute can create new student IDs and it does not compromise privacy.

The project is a university group chatting application that is mainly focused to join students in groups, where each group represents a subject and all students who have been enrolled in this group can communicate together under the supervision of the educational institute administrators. The system is going to integrate with other systems to enable automatically joining students in the groups, by doing so this helps students engage with the course updates and news or discussions.

# 1.2 Aims of The Project:

This project aims to provide the following list of features that students may benefit from After completing the project:

* Develop an official platform for students and tutors to communicate in a group-based system.
* Allows students and tutors to communicate privately and securely via text messages and share pictures, voice notes, videos and documents.
* Keep students conveniently connected and updated on registered courses.
* Improve privacy and security by using student ID and the unique identifier instead of mobile number or email.
* Preventing unauthorized persons from joining the system.
* Monitoring the groups under the supervision of the university administrators.

# 1.3 Problem statement

One of the major difficulties we face as students is to choose between sharing your personal information by giving your mobile number or being left without knowing the updates and news of the class. another issue we as students face is using many platforms to communicate with each other or with the university tutors’ ether it was formal communication method such as email or informal method such as WhatsApp.

The multisource of information leads the student to waste time searching for it instead of focusing on learning, by providing one source of news and updates also helps reduce the duplication of data or missing information.

The existence of external unauthorized parties who claim to help students by guiding them, answering their questions, share un-trustable study materials, may lead the student to plagiarism or even worst.

# 1.4 Project Scheduling

This section describes the methodology and the planned time for the completion of each milestone. The project has been divided into milestones; each milestone contains small tasks that can be done in a small period.

|  |  |
| --- | --- |
| Order | Milestone |
| 1 | Data Collecting |
| 2 | Analyzing data |
| 3 | Design |
| 4 | Development |
| 5 | Test |

Table - a table that shows a list of milestones

## 1.4.1 Data Collecting

In this phase we visit the Arab open university and interview some students, and some of the training staff by taking their views and their problems that they suffer.

## 1.4.2 Analyzing data

At this stage we compile the data which has been collected in the previous phase by making it in a list of problems. we categorize it into two parts, critical problem, realistic problem.

## 1.4.3 Design

After the analysis, we think about the software domain and choose the type of database that we will work on and determine what programming language will fit our needs. The design of the user interface should be clear, easy and simple to use, to make it easier for students and tutors to achieve the maximum benefit with the least time and effort.

## 1.4.4 Development

At this stage we begin the initial programming of the program.

## 1.4.5 Test

In this phase we test the program completely in each step of the previous stages in terms of use for students and tutors.

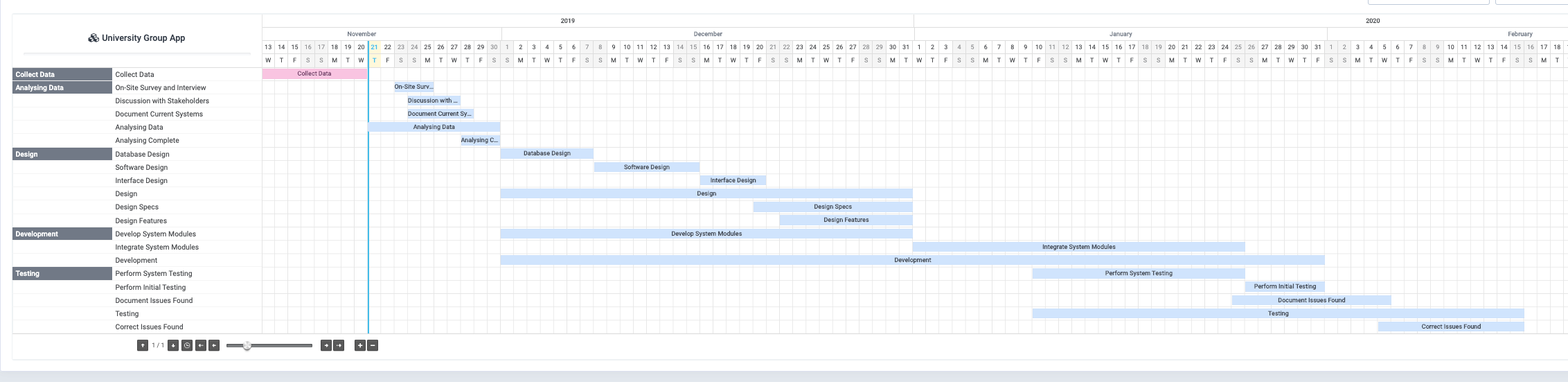


Figure - Gantt Chart of the project time table with milestones periods of each task

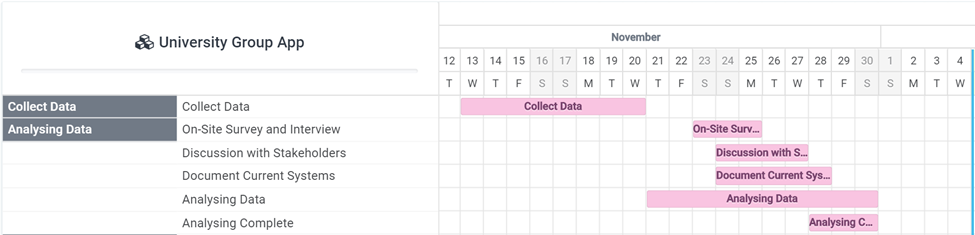


Figure - Gantt Chart for Collect data & analyzing Data milestones

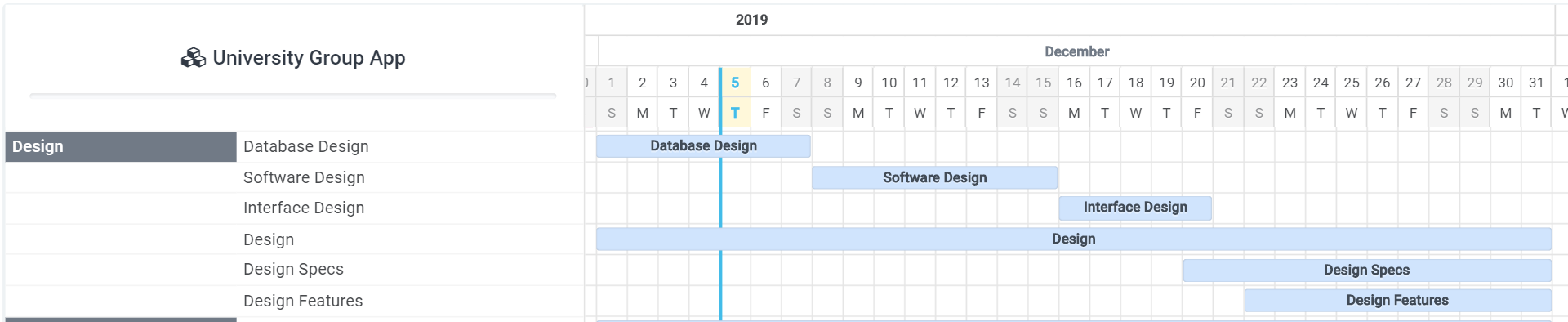
****

Figure - Gantt Chart for Design Milestone

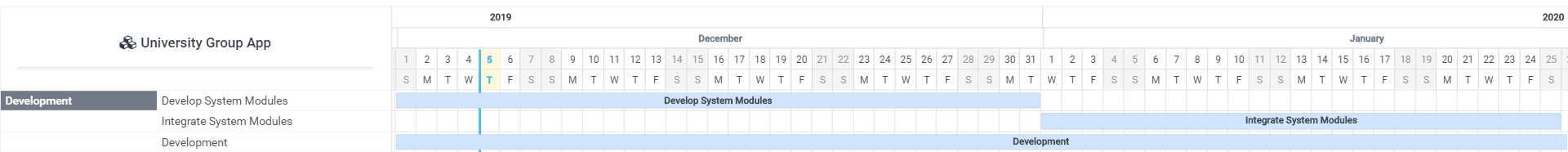
****

Figure - Gantt View for Development Milestone

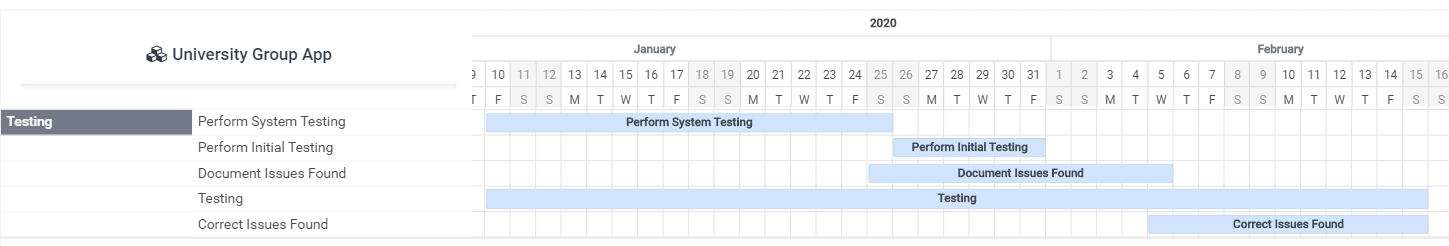
****

Figure - Gantt View for Testing Milestone

# 1.5 Document Structure

This documentation is organized by chapters, albeit not explicitly indicated in some of the sections.

In the following chapters we are going to start by discussing and comparing the current systems and what makes the suggested systems a good solution for the current circumstances. Next, we are going to provide some citations and results of reports. Followed by stating the requirements and analysis which includes the charts, diagrams and drawings that illustrate how the system should work.

Chapter Two:

Related systems & comparative analysis

# 2.1 History of Messaging

The internet has simplified our life in a way no one could have imagined, it changed how people communicate and express feelings, we used to sit together face to face and chat for hours. Nowadays, the internet made it possible for us to chat, make an audio call or even a video call with someone that is hundreds of miles away.

The first text message that has been sent over the network was sent in 1992, after that the methods of chatting have changed. In 1996, a software called ICQ has been introduced to the market as a group chat platform, it allowed many users to chat with each other at the same time and share files and documents over the internet. (ERICKSON, 2012)

In the 2000s, mobile phones made it even simpler by allowing users to install their favorite applications and start chatting with their friends on the go, without the need to use a computer. The rise of social media like Facebook, Twitter, and video conferences application helped people to communicate, sharing media and start messaging each other even more.

In any educational institute, students need to stay in touch with each other and interact, new platforms like LMS helped in a way to keep all information in one source. But student’s lifestyle has changed, students need to access information she/he requires anytime and anywhere by using a device that is between her/his hands.

Nowadays, all of us can reach information anywhere using a mobile phone and the need for a computer became less important.

# 2.2 Comparing Current Systems

Before comparing with current systems, it is important to note that the comparison is based on some of the main characteristics of the group chat feature, and not the 1-1 chatting systems. We are going to start with the most popular application that students use the most, and state the pros and cons of each system.

## 2.2.1 WhatsApp

WhatsApp is an instant messaging (IM) application that has been created to connect people using the mobile number as the unique identifier and the user’s contacts are the ones stored in the phone’s memory. The main purpose of the application is to replace SMS service. It is the most popular application on any mobile application store. (WhatsApp, 2020)

|  |  |
| --- | --- |
| Pros | Cons |
| Groups can have many admins | Groups are limited to only 256 members only |
| Authentication is simple and fast | Can only use mobile number |
| Only contacts see your profile | Anyone can send you a message |
| Can share media files | File size is limited to 100MB |
| Can send voice notes | Only web version for desktop |
| Can send pictures directly from the camera | New members cant see history |
| Can make voice calls & video calls | Unofficial |
| END-TO-END ENCRYPTION |  |
| Available on Android, iOS, and a web version |  |
| Only installed one device |  |

## 2.2.2 Facebook Groups

With over 1 billion users worldwide Facebook is the social network platform that helps users to stay in touch with friends and family members. Facebook Groups can be created by individual users it allows members of that group to interact with posts, question links, pictures, video, and comments. (Facebook, 2020)

|  |  |
| --- | --- |
| Pros | Cons |
| Simple to use | Requires Facebook account |
| Access from multiple devices | Post based |
| Ability to share media and links | Anyone can join the group |
| Multi admin with permissions | Personal data is available for anyone |
| User can see group history | Search of information is hard |

## 2.2.3 Telegram

Telegram is also an instant messaging application that has been introduced to the market in 2013, ever since telegram gained his popularity by enabling features that other IM applications did not provide such as channels, large groups, AI bot large media sharing and many more. (Telegram, 2020)

|  |  |
| --- | --- |
| Pros | Cons |
| Messages are encrypted | Can access |
| Access from multiple devices | Not popular |
| groups of up to 200,000 members | People can find you via username |
| Media sharing | Unofficial |
| Uses username and mobile number |  |
| Replies, mentions, hashtags |  |
| Pinned messages inside the group |  |
| admins have privileges |  |
| Group permissions |  |
| File sharing up to 1.5 GB |  |
| New members can see history |  |

## 2.2.4 LMS Forum

Each University has its own Learning Management System (LMS) which is full of many features that many students may not know. One of the features is that each course in the LMS has its group where students can chat, ask and interact with each other. (Quigley, 2018)

|  |  |
| --- | --- |
| Pros | Cons |
| Official communication method | Post based |
| Moderated by university | File size sharing is limited |
| Media sharing | Late replies |
| Only current students can chat | Available only on web |
| Only student name and ID is available | Forum and not a group chat |

# 2.3 Stakeholders

The application aims to help students and teachers to communicate with each other without the need for external applications that are not related to the educational institution. And to achieve the most possible benefits from the new technologies through instant messaging without the need to compromise privacy which is a major interest of the parties.

List of stakeholders:

* Students
* Tutors
* University administrators

# 2.4 Applications

The application can be used in any educational institution, universities, academies and colleges that have a student ID and emails for all students, and where the educational institution needs to control the sharing of information.

Chapter Three: Requirements & analysis

# 3.1 Requirements and Features

The hardware and software that makes the application run on a smartphone should have the following requirements:

* Mobile phone that runs on Android OS or iOS.
* A working touch screen or a mouse to input data.
* Virtual keyboard or physical keyboard.
* Ability to connect Wi-Fi or mobile plans.
* Ability to exchange data over the network.
* Ability to use the phone’s microphone.
* Ability to use the phone’s camera.
* Ability to access the phone’s storage.
* At least 500MHz processor speed.
* A minimum of 512 MB RAM.

# 3.2 Functional and Non-Functional Requirements

In this section we are going to state the functional and non-functional requirements that define the characteristics of the application.

## 3.2.1 Functional Requirements

* **User Registration**: a user must be able to register for the application using his/her student ID after installing the application, then receive an authentication message on his/her university email.
* **Joining Groups**: the application must be able to join the student to groups which he/she has enrolled in using the university system.
* **Sending messages**: a user must be able to send and receive chat messages and share multimedia files (emoji, audio, photo, video, documents, etc.).
* **Broadcast Message**: administrators and tutors must have the ability to send broadcast messages to all members.

## 3.2.2 Non-functional Requirements

* **Scalability**: The application should be able to provide an instant messaging service to a large number of users at the same time, and can create new groups by system administrators.
* **Privacy**: Messages shared between users should be encrypted and should not show the user’s first name or any other private information except his student ID.
* **Performance**: The application should be lightweight, be able to send messages instantly, and not consume large amounts of memory or storage, concerning speed and other functional requirements.
* **Robustness**: In case of losing the mobile or any other issue, all chat history is stored in the server database to enable recoverability.
* **Integrity**: Addresses consistency of data; the ability to prevent unauthorized modification or deletion of data.
* **Security**: Using two-factor authentication upon registration helps prevent unauthorized access to the user’s account.

# 3.3 Project Diagrams and Charts

## 3.3.1 Use-Case Table

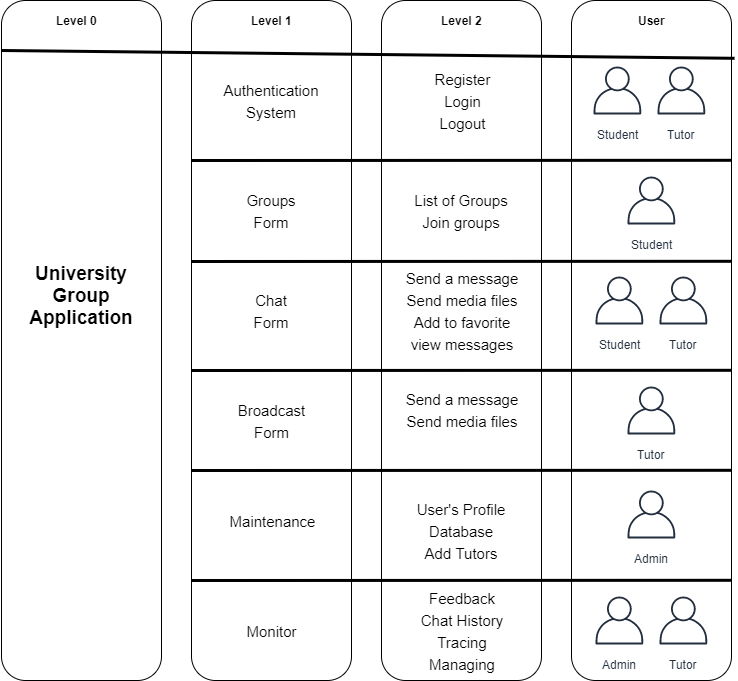
****

Figure - Use-Case Table of The System

|  |  |
| --- | --- |
| Use Case Name | User Registration |
| Unique Identifier | UC01 |
| Actors | Student |
| Pre-Conditions | Application must be installed on the mobile |
| Basic Path | The user enters his/her student ID and a registration to the system is made |
| Alternative Path | User already have an account and he/she has been redirected to the login page |
| Post-Conditions | The user is registered in the students table |

Table User Registration Use-case Table

|  |  |
| --- | --- |
| Use Case Name | Tutor Registration |
| Unique Identifier | UC02 |
| Actors | Admin |
| Pre-Conditions | Tutor name and email must be provided, and the email must be valid |
| Basic Path | The admin registers the tutor via the admin panel of the system |
| Alternative Path | User already have an account and he/she has been redirected to the login page |
| Post-Conditions | the tutor is registered in the tutors table |

Table Tutor Registration Use-case Table

|  |  |
| --- | --- |
| Use Case Name | Joining Groups |
| Unique Identifier | UC03 |
| Actors | Student |
| Pre-Conditions | Student must be registered, has logged in to the application and has enrolled in at least one course |
| Basic Path | The registered courses groups show up and the user can chat |
| Alternative Path | A message shows the student that he has not been enrolled in any course |
| Post-Conditions | The student can chat and interact with the application. |

Table Joining Groups Use-case Table

|  |  |
| --- | --- |
| Use Case Name | Send a massage |
| Unique Identifier | UC04 |
| Actors | Student / Tutor |
| Pre-Conditions | 1) User must be logged in to the application and has been enrolled in at least one course.  2) an internet connection is established. |
| Basic Path | The message has been sent to the group |
| Alternative Path | 1) The user must be enrolled in at least one course.  2) The message is pending until a good internet connection is established. |
| Post-Conditions | The student can chat and interact with the application. |

Table - Send a Message Use Case Table

|  |  |
| --- | --- |
| Use Case Name | Broadcast |
| Unique Identifier | UC05 |
| Actors | Admin / Tutor |
| Pre-Conditions | 1) User must be logged in to the application.  2) an internet connection is established. |
| Basic Path | The message has been sent to all users |
| Alternative Path | 1) The message is pending until a good internet connection is established. |
| Post-Conditions | A list of all users who received the broadcast |

Table - Broadcast Use-Case Table

## 3.3.2 Use-Case Diagram

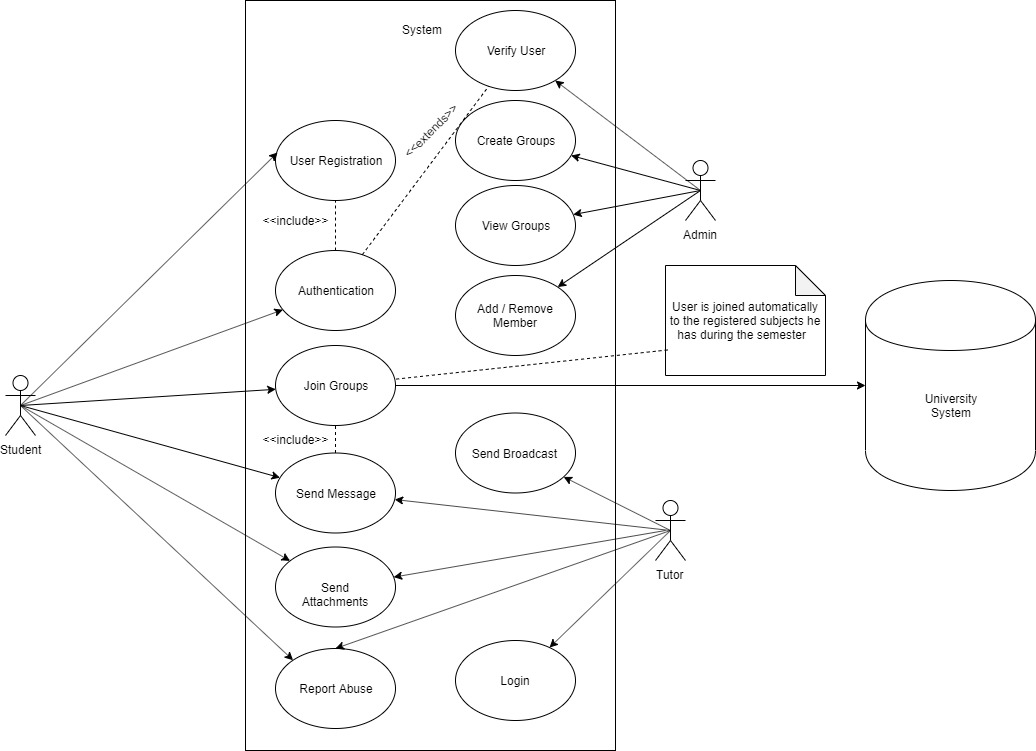


Figure Use Case Diagram That Shows How The System Works

## 3.3.3 Flowchart Diagram

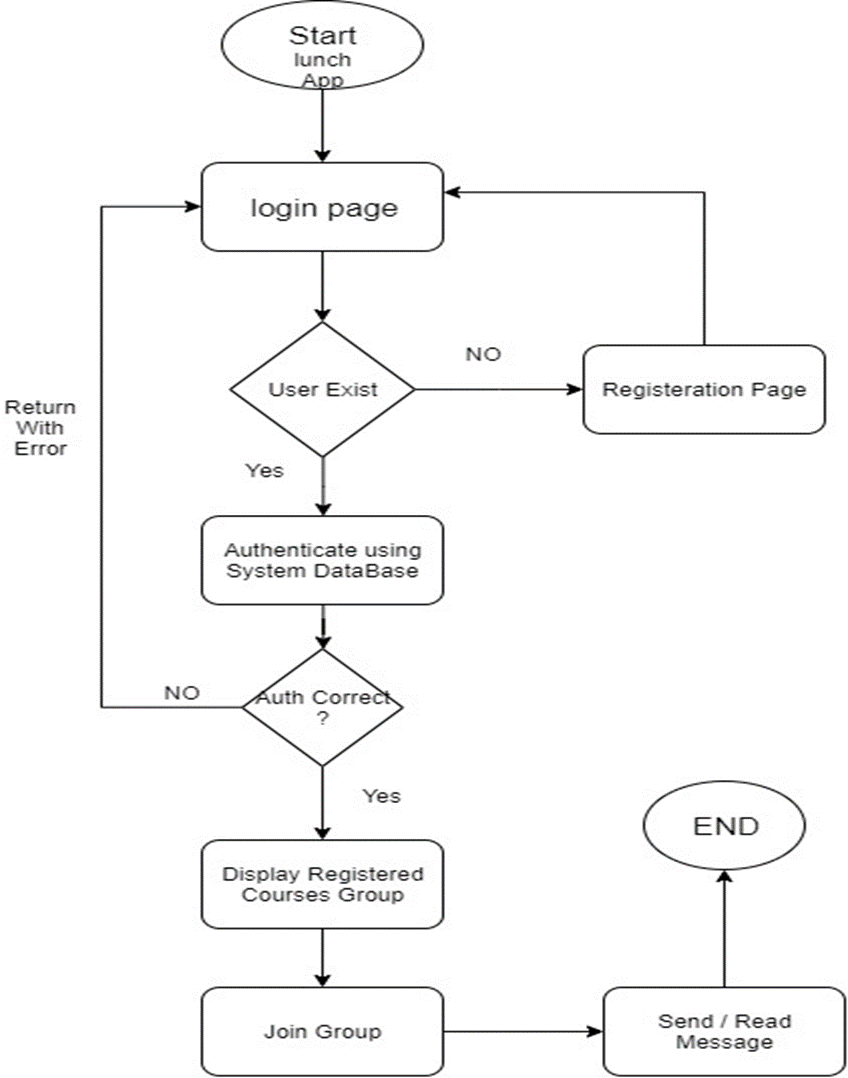


Figure – Flowchart diagram that shows how the system works

Chapter four:  
Design, Implementation,  
and testing

# 4.1 Just an Idea!

All great products started from just an idea, some found reached success and others were doomed to fail. While many reports suggest that 90% of startup projects fail due to many reasons; No market need, not the right team, get outcompeted, poor product, ignore customers, lose of focus, and many more other reasons. If we think of the reasons for a second, we can notice that most of them are related in a way or another to the way the project has been executed.

And to stay on track, deliver on time and control the workflow of the project we need to apply a method for managing the project, while there are many good project management approaches, each of it has its characteristics, requirements, benefits, and drawbacks.

After discussing the project idea with the team, researching, and asking friends with previous experience in managing software projects, we have selected to use Scrum as our approach to developing this project,

# 4.2 Scrum Framework

Scrum is an agile project management framework with proven results in decreeing time, improving product quality and heightening customer satisfaction all while lowering costs of time and money. or as its creators define it “A Better Way of Building Products”. (Scrum.org, 2020)

Scrum is easy to explain, but the application and mastery are often difficult. While scrum does not fix anything, it depends mainly on the team members to do everything. It also requires good teamwork which makes it essential. If the team members do not cooperate well, the project will face failure.

The scrum framework implements the idea of breaking the project into sprints, each sprint is to deliver a function but before releasing the feature to the customer each sprint of the project has to go through these 7 steps:

1. Requirement elaboration
2. Design
3. Development
4. Comprehensive testing
5. Integration
6. Documentation
7. Approval

The seven steps to fully build the scope of each requirement are performed for every item. Every requirement has taken on during a sprint no matter how small or large it is fully built, tested, and approved or rejected.

What is good about Scrum is that it is an incremental and iterative approach to achieve the best results possible with the minimum time required to achieve that goal. It consists of 7 stages:

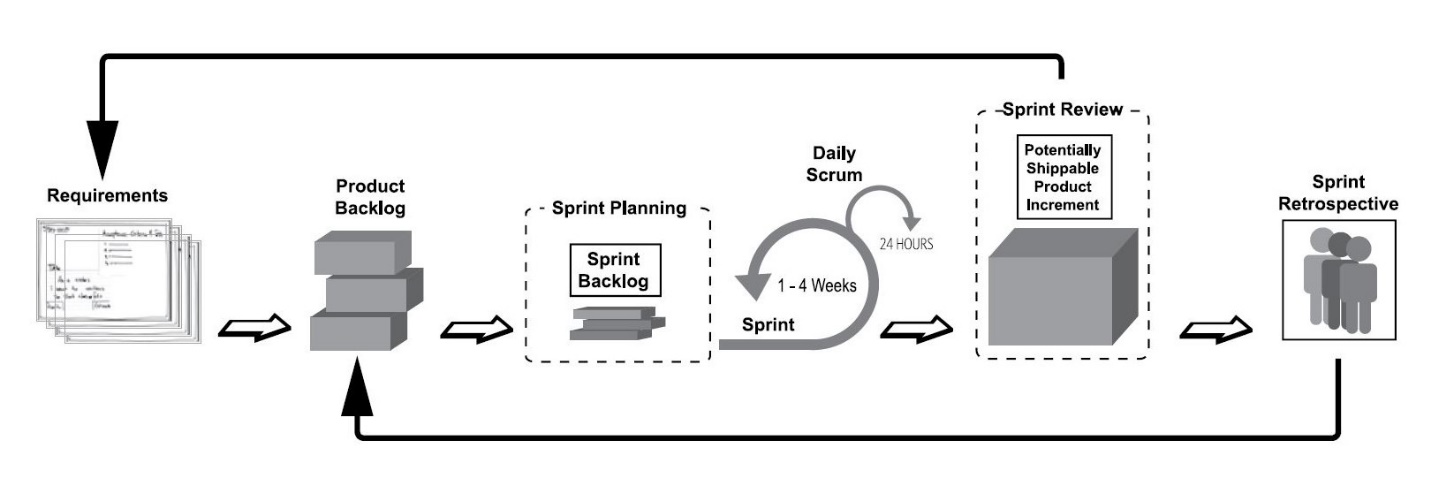


Figure A simplified overview of the events and cycles of scrum.

## 4.2.1 Collecting Requirements

At this stage we collected the requirements from the stakeholders (students & tutors) about the product and what are the thing they expect to see in the app. We used a *user story* method that enables us to gather the requirements then be transferred into sprints then split into smaller tasks that are easier to achieve, such as “I want to be able to login using my student ID so that I can hide my phone number from others”.

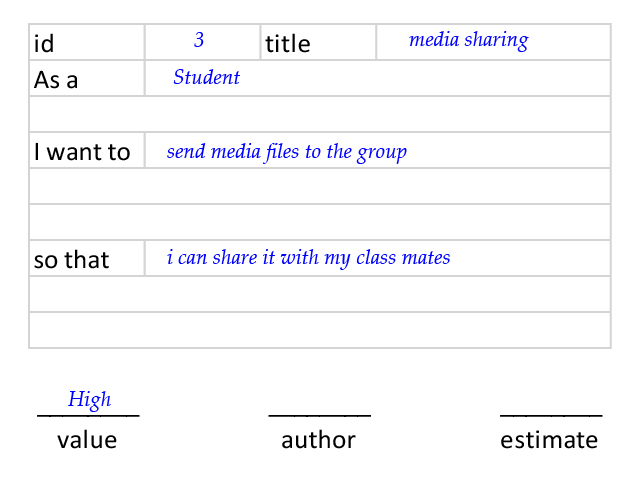
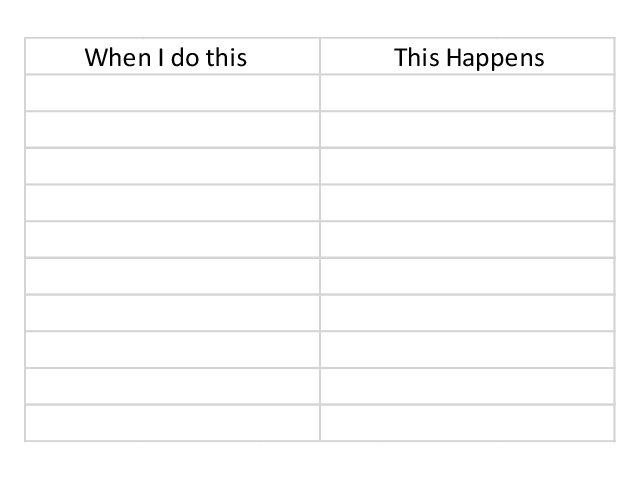
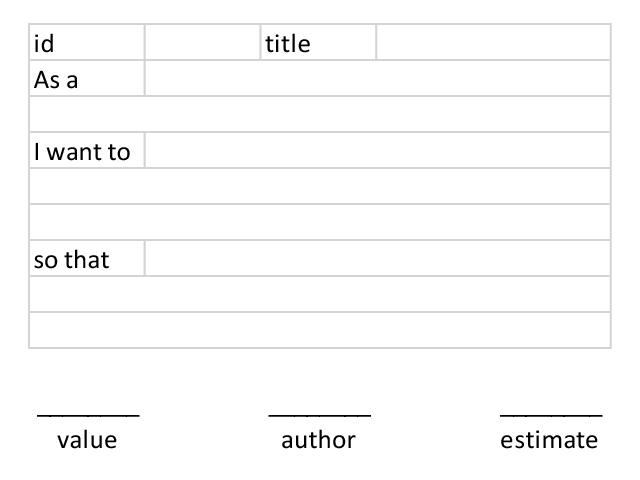


Figure Example of User Story Card

Figure Empty User Story Card



We used cards to write user stories, which makes it easy for us to transfer the conversation into rich requirements before the start of development.

## 4.2.2 Product Backlog

After collecting the requirements, refining it with the development team and renegotiate it with the users, we write everything in the product backlog to be the core of the project where everything is transparent and visible for all team members and only the product owner can add to it.

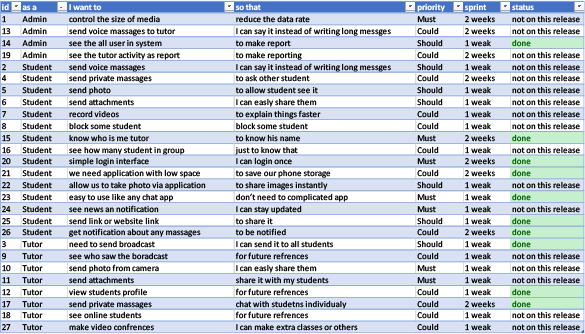


Figure Example of the product backlog

## 4.2.3 Sprint Planning

Unlike other methodologies for project management, sprint does not require the development team to write technical specifications of what exactly needs to be done, but only the users need then the team figures out how to implement that need in the best possible way.

## 4.2.4 Daily Scrum

Each day we meet or chat with each other about what is done and discuss the progress, and identify and solve the issues. For no more than 2 minutes we report to each other “what I did yesterday”, “what I’ll do today”, “what’s in my way”.

## 4.2.5 Sprint Review

In this stage, at the end of the sprint, we review our finished work and decide if it is shippable or not. If not, how many hours does it takes to be done?

## 4.2.6 Sprint Retrospective

This stage we test and check if the finished work is done right or it can be improved.

# 4.3 Implementation

After defining what features need to be done using Scrum, we started the phase of transferring an idea from written words into a working code.

As the project consists of two parts, one is a web-based application and the other is a mobile application, we are going to explain each part separately then discuss the method we worked on to connected the two platforms together. Then we dig into the features each platform has to offer for its users, and compare the other alternative options we might have used to develop the project and why we choose those methods, programming languages and frameworks for this specific project.

## 4.3.1 The Database

The core of the application is the ability to store messages, users, details about groups and its settings. All two platforms should connect in a way or another to one database that can handle the system, easy to maintain and upgradeable for future versions if there is a need for it. After investigation of the requirements we found that users, roles, groups, messages are required for a basic chat application. With help from another table group-user to connect a user to groups. So, we used MySQL as our choice for a database to create the above-mentioned tables.

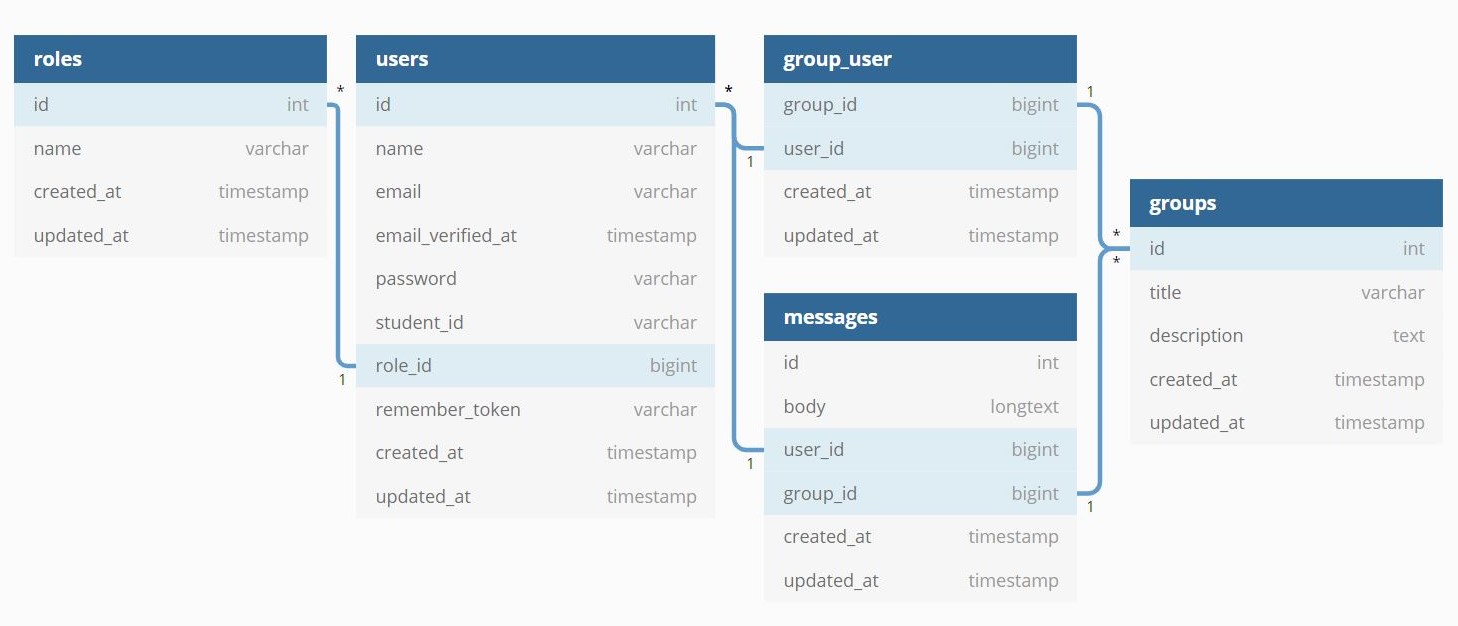


Figure Database Diagram

## 4.3.2 Web-based Application

As the product backlog defines the need of the users with the role of “Admin” for an ability to manage the system, import and export data, send a broadcast, and other features. The mobile application alone is not very efficient in that way, so we had to create and web-based application that enables admins to do these functions.

For the web application to able to be upgradeable and easily maintainable we had to use an architectural pattern known as model-view-controller (MVC) which divides the logic of the software into three connected parts.

MVC (Model-View-Controller) is a popular design pattern with the following components, Model which is the central component that manages data, logic, and rules of the application, View that displays the information to the user and Controller which connects Model with its View by accepting inputs and converts it to commands to the Model or View. (Kumar, 2020)

The web application should be also divided into two parts, one is the backend that holds the core of the application and the other is the frontend that is responsible for the user interface. With the help of the MVC design pattern it was easy to do so.

For developing the backend, there are many available programming languages to helps us develop it, but we choose PHP with Laravel framework version 7.0 to develop it for many reasons we will explain at the end of this chapter.

For the frontend, there are more frameworks than we can count, and to keep things simple we used HTML, CSS, JS with the Bootstrap framework.

The web application has the following features:

* Login page and register page for new users (students).
* Admin dashboard.
* List groups, and the ability to create groups.
* Ability to change the title or the description of a group.
* List of students, the ability to create and manage the student account.
* List of tutors, the ability to create and manage the tutor account.
* List of admins, the ability to create and manage the admin account.
* Ability to import data.
* Ability to assign and unassign tutors and students to groups.

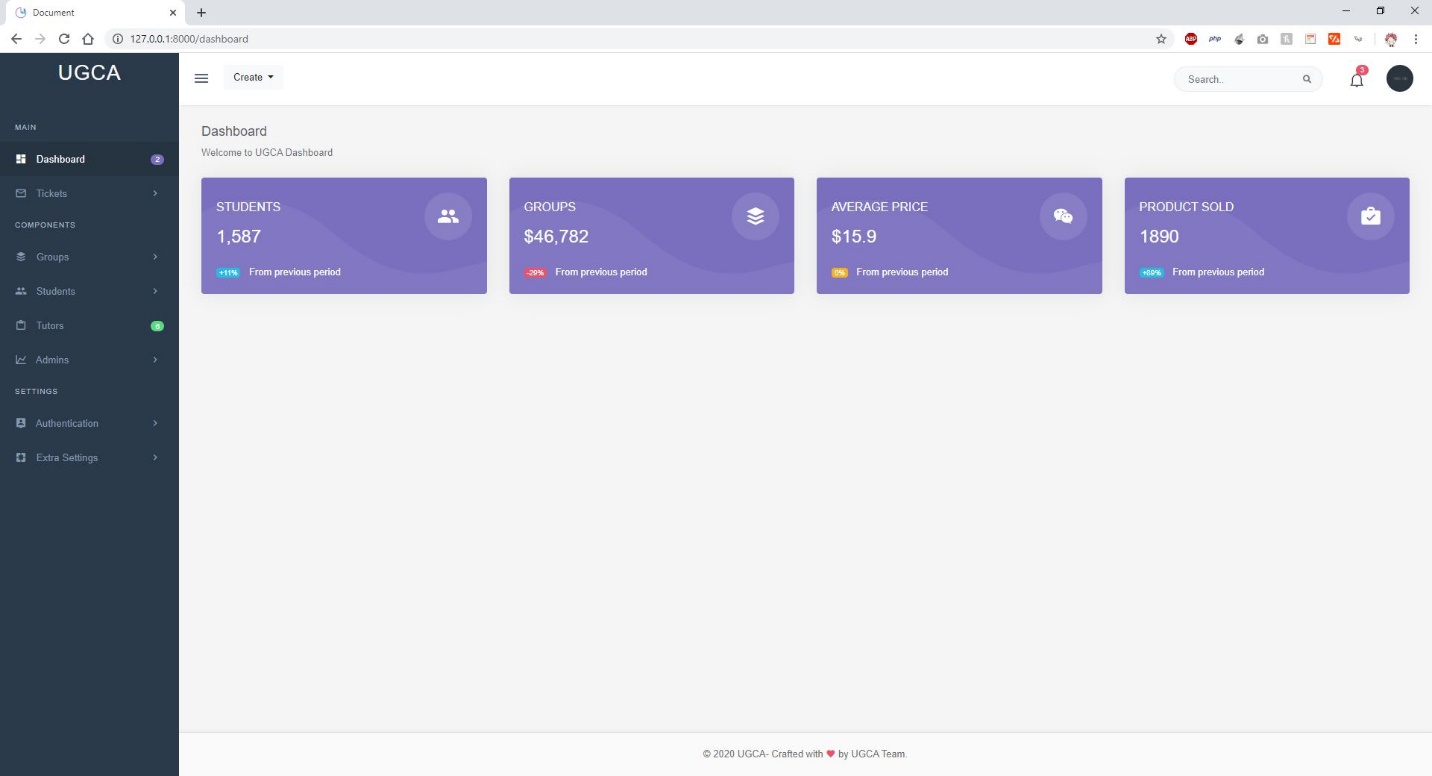


Figure the admin dashboard

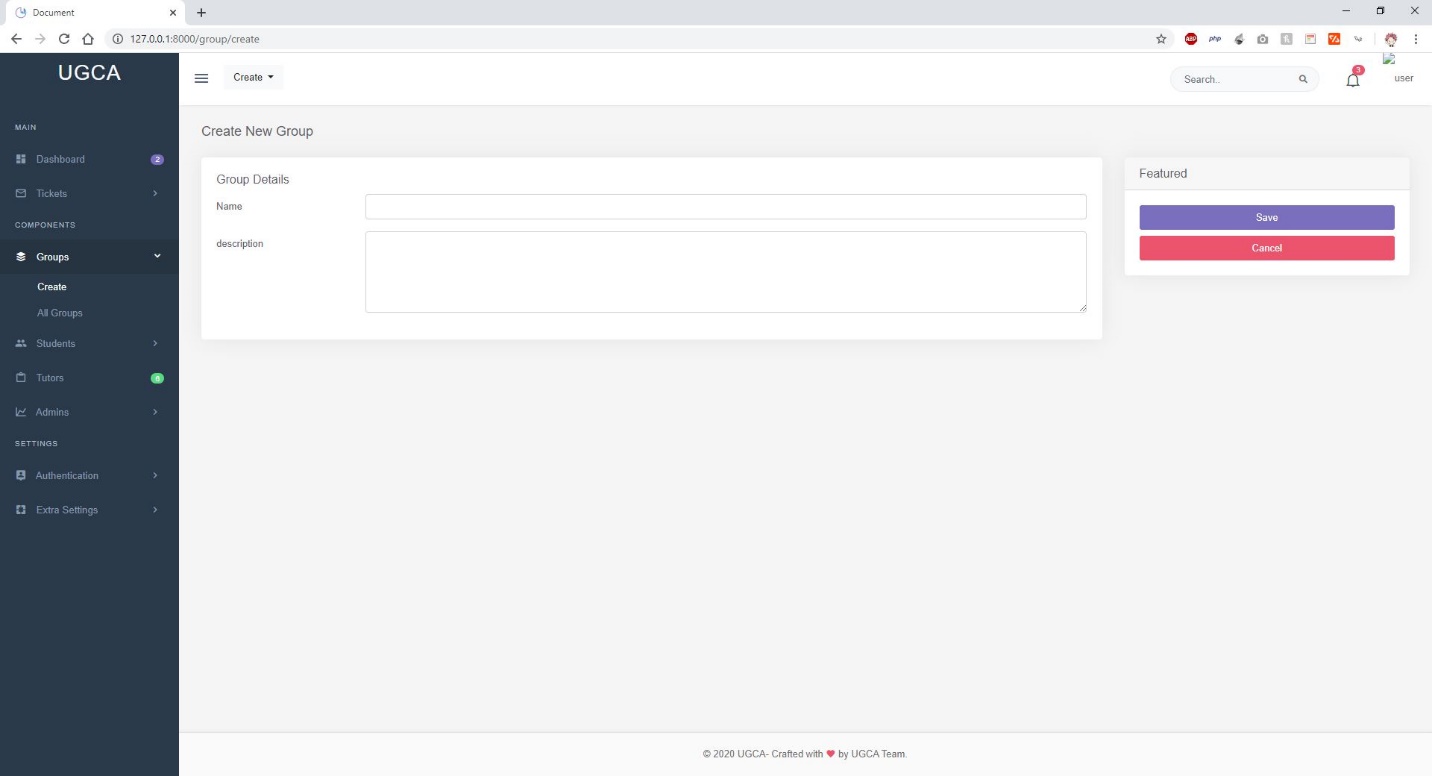


Figure create group form

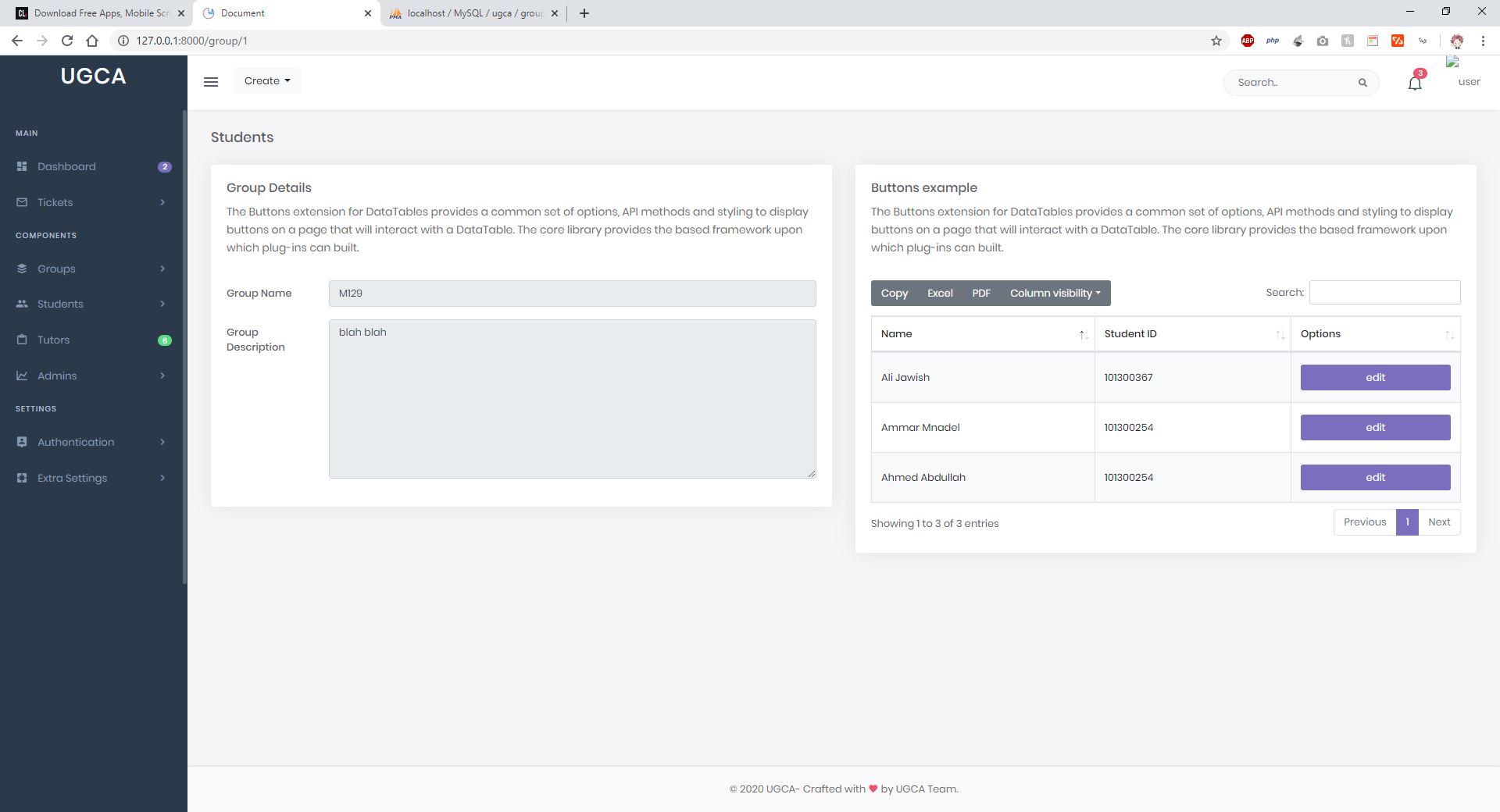


Figure Group list of students

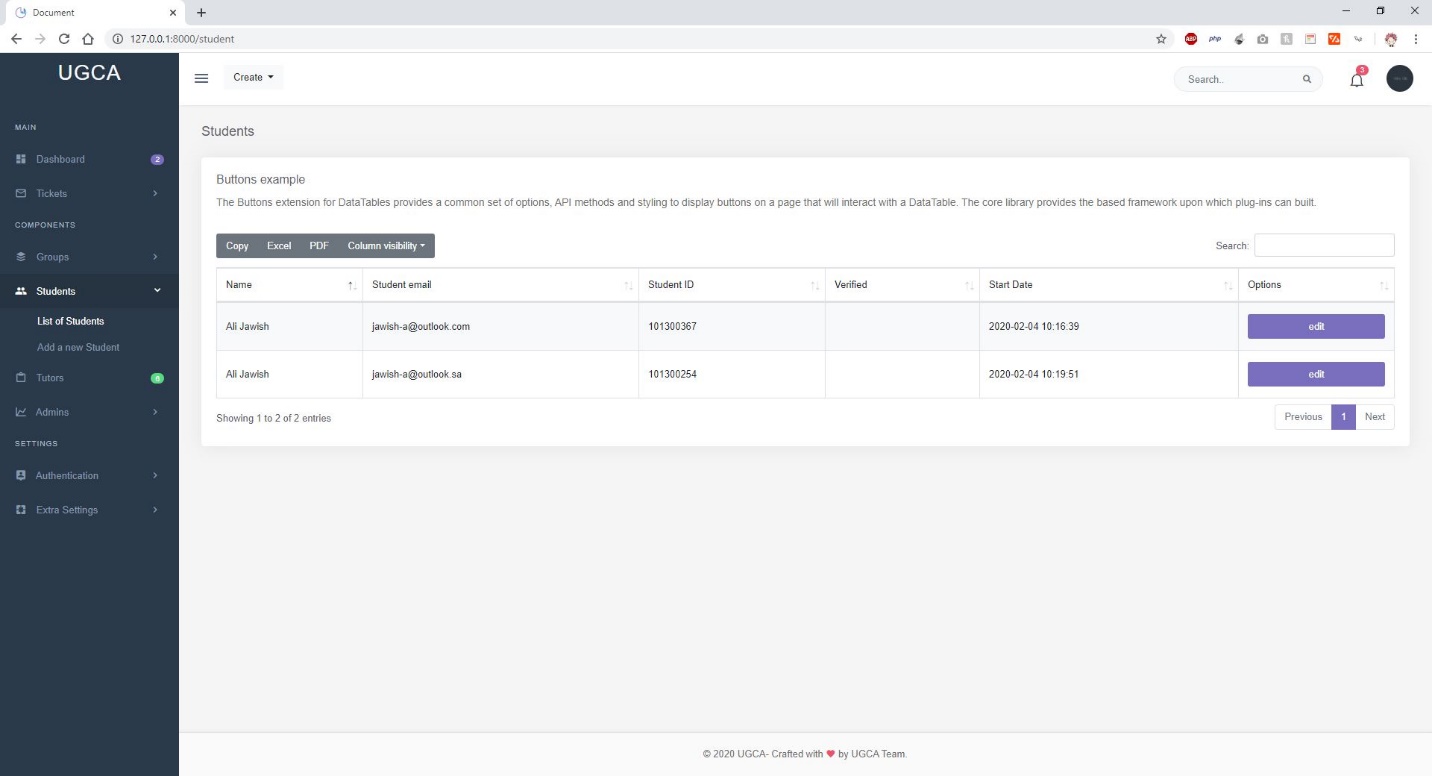


Figure Students List

## 4.3.3 Mobile Application

For the students to be able to use the full functionality of the system and due to their requirements, a mobile application might be the best solution due to its nature that enables students to chat whenever and where ever they need.

To give the student the same look and feel over both platform, Android & iOS we had to do a custom design of the UI/UX. At first, we draw some sketches using Adobe XD to imagine how the mobile app would look like for the students. Then we asked some students for feedback, do some editing and changes to the sketches, and after that we transferred it to a code.

The first challenge was to do create an app for both platforms at the same time, By using Dart with Flutter framework as the programming language with its cross-platform technology, it helped us overcome this challenge.

The second challenge was to connect the mobile app to the database, either to fetch data or to post new data. This issue can be solved by using API (Application Programming Interface) calls to integrate the mobile app with the web application that is already connected to the database.

The third challenge is to keep the chat in real-time between users. To do so, both the client and the server must keep an open connection channel to push and retrieve messages. Sockets or WebSockets made this possible, which is as described in the MDN web docs website “an advanced technology that makes it possible to open a two-way interactive communication session between the user’s browser and a server” (Mozilla, 2020)

Again, we are going to discuss the reasons and benefits of our choices at the end of this chapter.

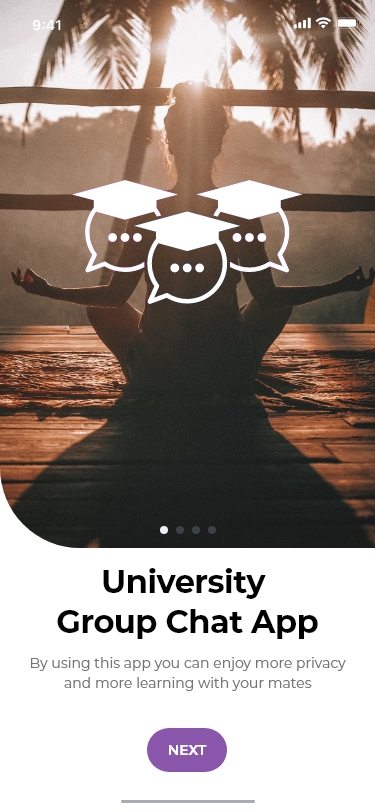
****

Figure welcome screen of the app

****

Figure second welcome screen

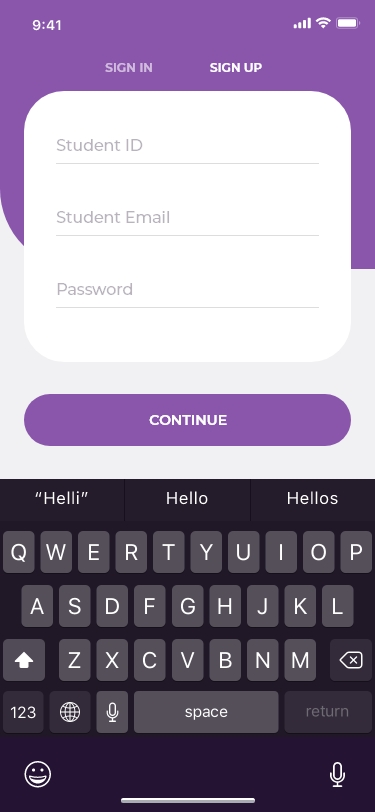
****

Figure register screen for new users

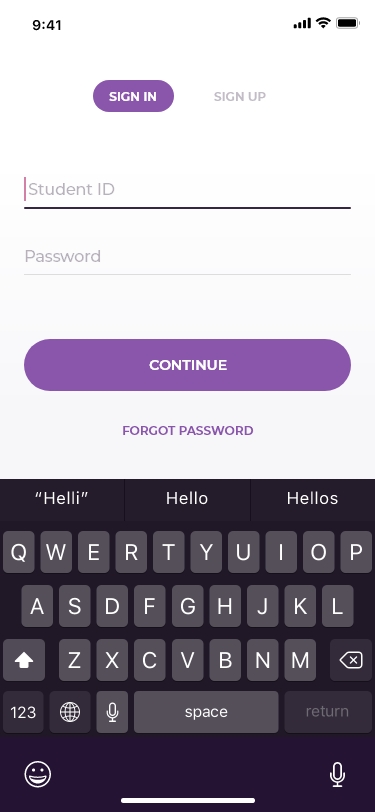
****

Figure login screen

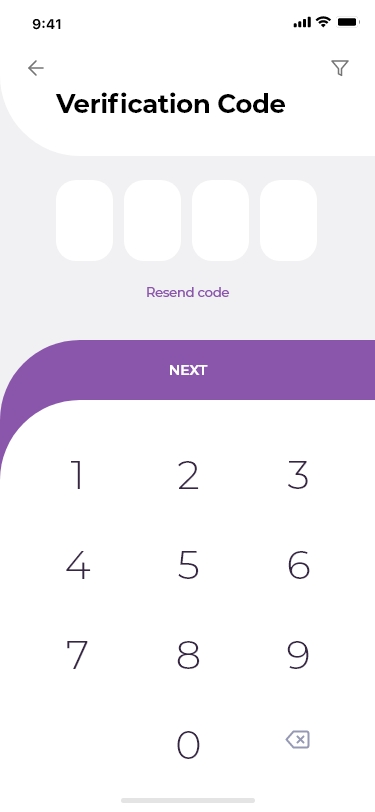
****

Figure Verification screen

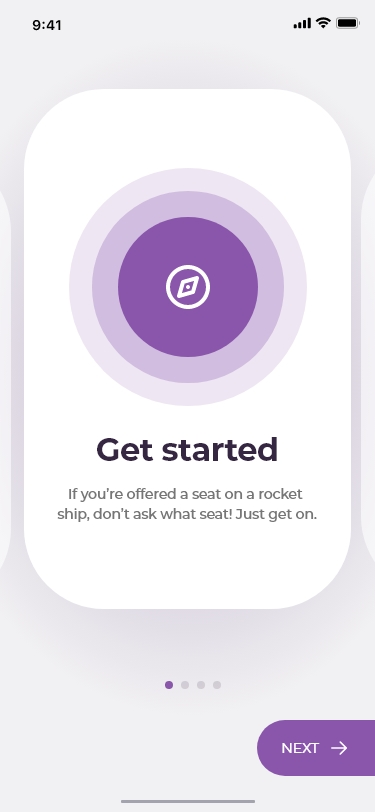
****

Figure intro screen with information

****

Figure list of groups screen

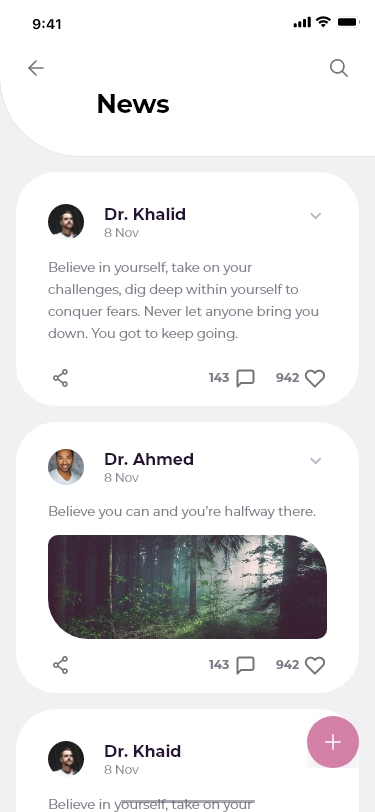
****

Figure broadcast messages screen

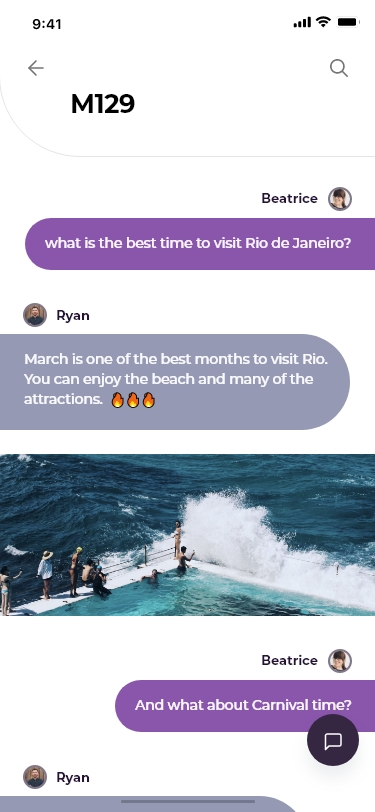
****

Figure in group chat screen

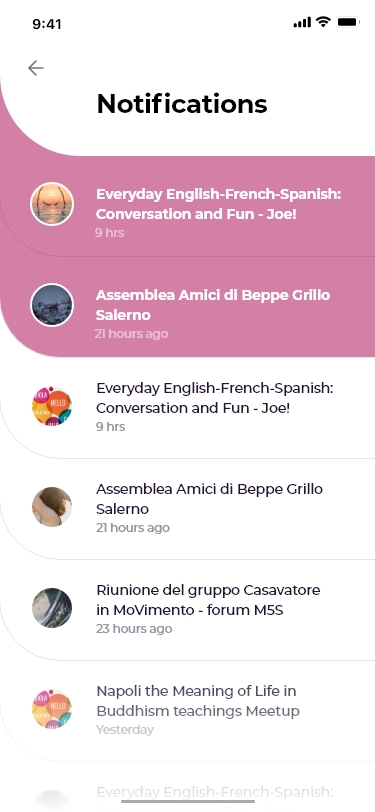
****

Figure notification screen

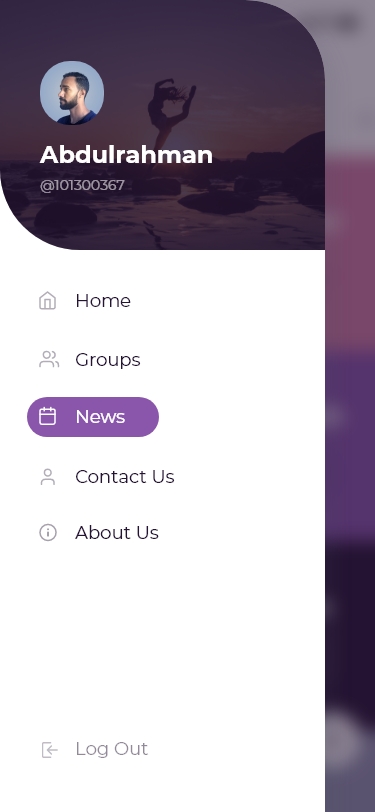
****

Figure side menu

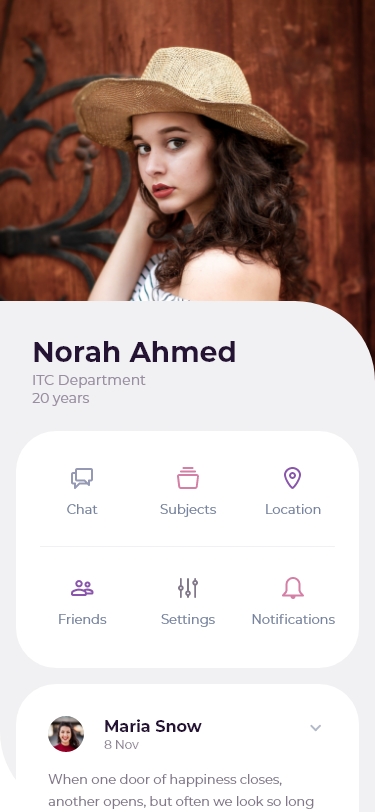
****

Figure profile screen

## 4.3.4 Connecting it all together

To summer up all the implementation described above, we created a web app that is connected to the database, provided API routes for other systems to connect to it, created a mobile app that is connected to the database via an API connection.

For chat functionality, we use a web-socket to connect clients.

## 4.3.5 Alternatives and comparing

In this section we will list alternative methods of implementation of the project and explain the reasons behind our choices.

For the database, there are two major technologies for structuring the database, SQL and NoSQL. While SQL uses tables and relationships between tables to store and retrieve data, NoSQL is non-relation and each entry holds all its data. (Smallcombe, 2017)

NoSQL is more used if there is no clear structure of the data, but the structure of the is clear so SQL is the best for this kind of project.

For the web application, we could have used one of the three major programming languages used to develop websites, Asp.net, Node.JS or PHP. Due to Asp.net server requirements and its costs, we had to look at the other options, Node.JS is mostly used with NoSQL and MongoDB. That leaves us with PHP, nearly 70% of the web runs on it for its high number of developers and clear and rich documentations. (W3Techs, 2020)

For mobile app development, there are three main technologies, native, cross-platform and hybrid. Hybrid app is not suitable for this type of application, native app that uses Kotlin and Java for Android and Swift for iOS are kind of expensive to develop and they are time-consuming. Cross-platform solution way is suitable for the requirements and perfectly fits our needs. In this area we have another 3 options to choose from, Xamarin, React Native or Flutter, they all perform the same and offer lots of features and documentation. (Alferd, 2019)

# 4.4 Testing

Due to the nature of the Scrum project management approach, testing is done multiple times during the development process, to ensure the best possible delivery of the project.

The first test is done after each sprint, to test if the unit meets the sprint description, second is test is done after integrating the unit with the system, third is after delivering the project to get the feedback from the user to get acceptance.

## 4.4.1 Testing login and register

We will test the following, unauthorized users, password less than 8 characters, wrong password match, and empty field for the name.

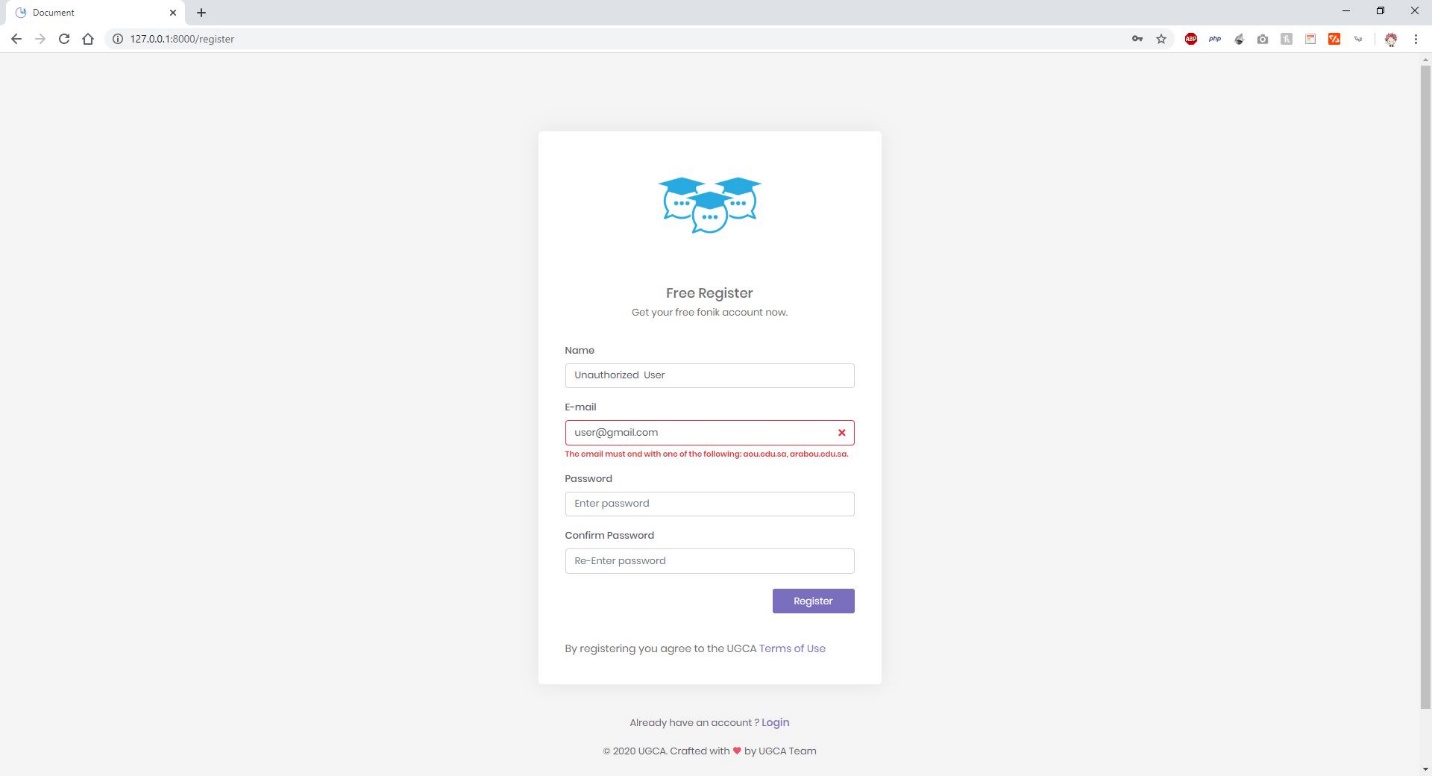
****

Figure unauthorized attempt test

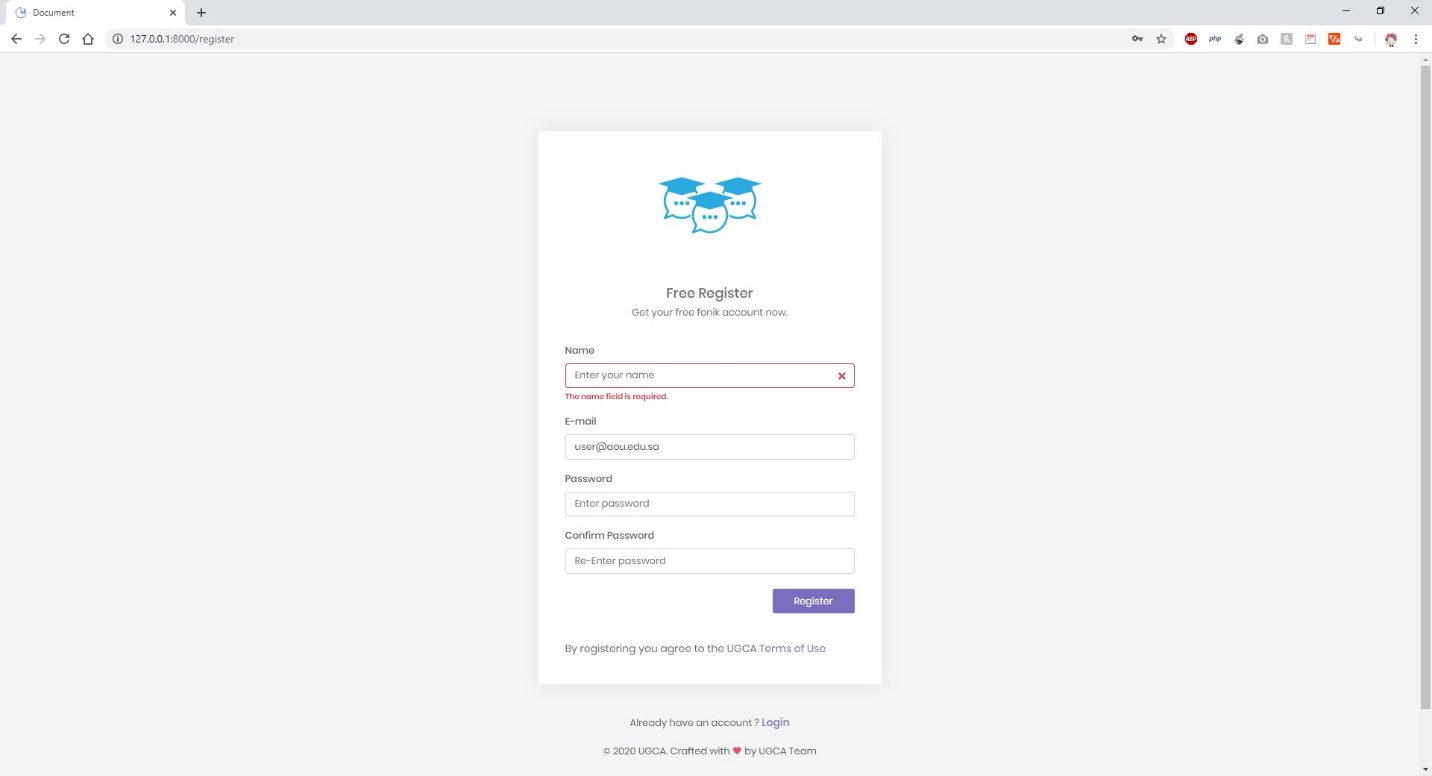
****

Figure empty field test

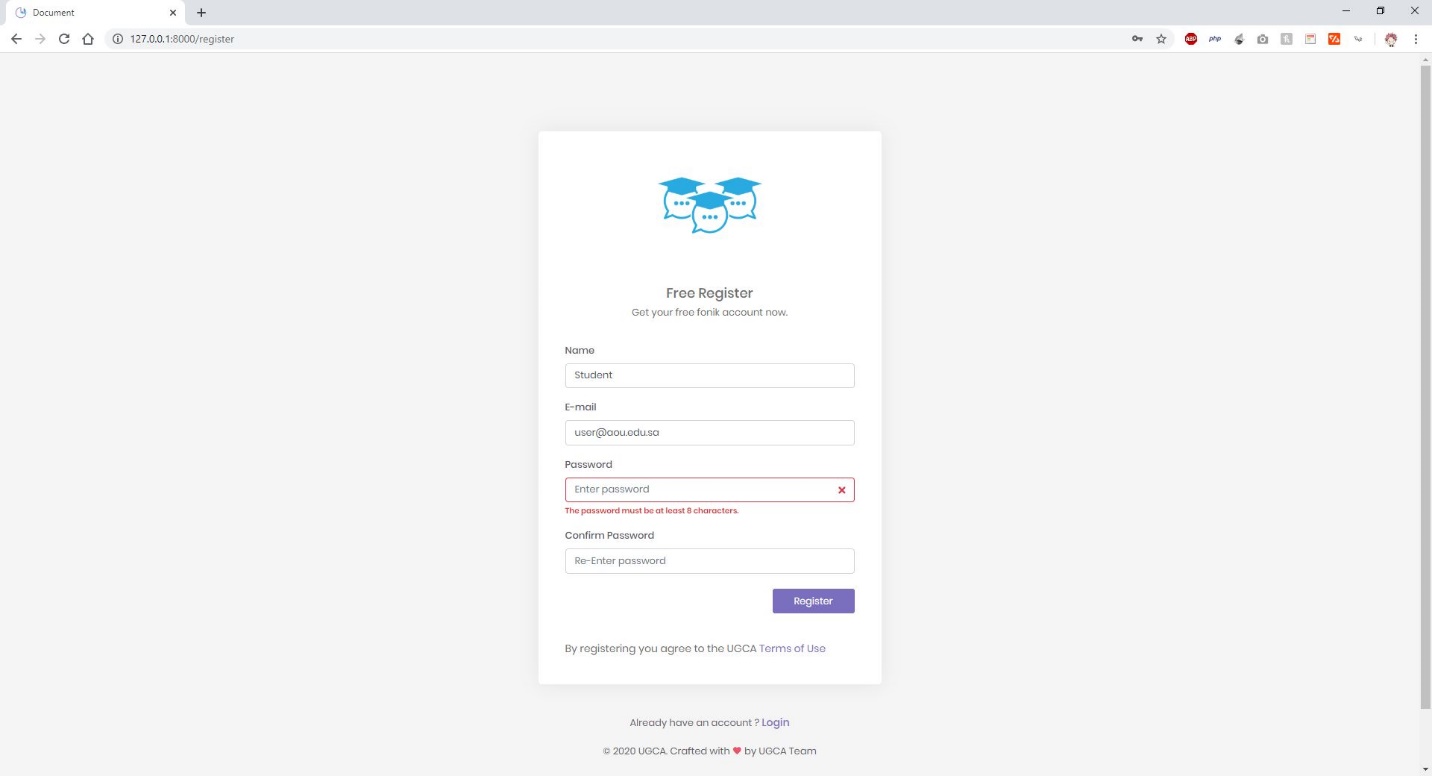
****

Figure password less than 8 charachters attempt

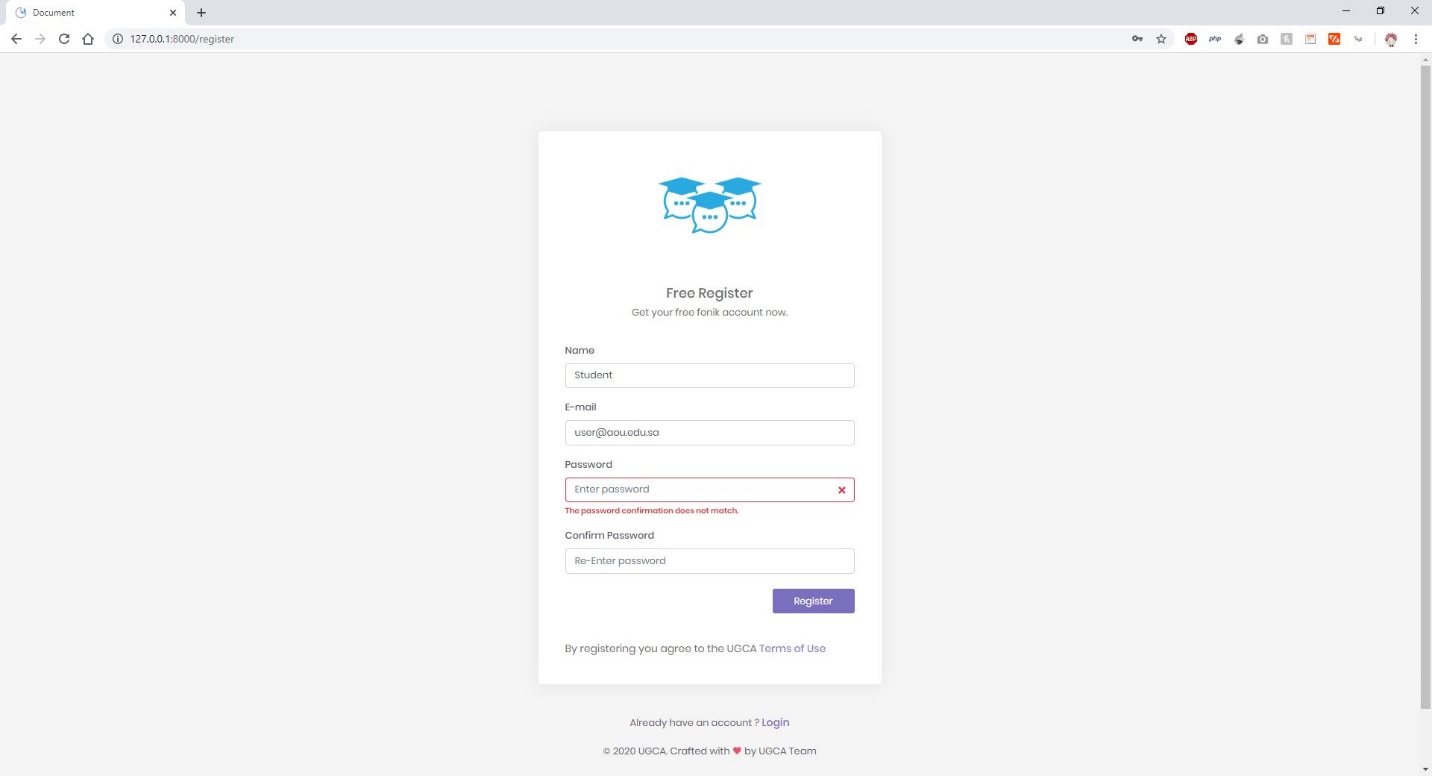
****

Figure password not match attempt

## 4.4.2 Testing Roles

In this test we test the roles of users, only users with admin roles can access the web application. if a new user registers he/she will be redirected to a thank you page with instruction on how to login. The admins can login to the web app and visit all pages. Else will be redirected to the home page.

## 4.4.3 Testing API

Using Postman software, we tested all API routes and validated it to assure it return JSON strings for the required data.

## 4.4.4 Testing group chat

Using the Mobile app we tested the chat functionality to validate that users get it simultaneously and it is stored in the database.

Chapter five:

Results and discussion

It was easy to think of an idea and this idea can be great, but it is really hard to transfer it into a real product or at least an MVP(Minimum viable product), no matter how much you think you know, you find out that there are still much to learn.

We aimed to help others and to contribute to the society by giving a better way for students to communicate with each other. During this process of trying to reach the goal, we discovered that learning can be at its best when there is applying to it.

Teamwork was the engine of this project, without it none of this would have happened. This also reminds us of why project management approaches always mention choosing a good team for the development.

The important thing to mention here is time management and organizing; at first, you find yourself lost between the 9 to 5 job and doing surveys, papers, documents on the computer, some working codes and some not working, many software to work with; but if you take a moment to organize everything things get easier and smoother.

Almost all the project’s aims and goals have been covered and achieved as follows:

* Students joined chat rooms automatically.
* Students can use his student ID to register or login to use the mobile application.
* Tutors are in the same group as the students.
* Only the authorized members can join the app.
* Users can send messages simultaneously without the need to refresh.
* Students can see the tutor in the chat group.
* Tutors can send messages in the group.
* Admins can create new groups, assign a tutor, import students, etc.

Due to the time limitation, lack of resources and other reasons, we could not do some of the features which we will state as follow:

* Attaching media to the chat groups, socket.io documentation is not clear in this regard and it requires an expert knowledge in WebSockets.(comparing to all)
* Reducing media size without losing quality, which also requires expert knowledge in media compression on the mobile. (Comparing to WhatsApp)
* Keeping a log of all events, which requires extra time to be done.
* Private messages between users, require feedback from users if they need the feature or not which can be determined after the release.
* Ability to use the camera for taking photos or recording videos, due to that the official package from Flutter to control the camera is not the final release.

But even if the core features achieved successfully, at the moment the UGCA project will not replace current systems due to the following reasons:

* Users got used to the current systems (WhatsApp, Facebook, Telegram, etc) and it is difficult for them to change now.
* Current systems like WhatsApp, Telegram or Facebook groups, still offer more features than UGCA.
* The user might not find it easy to use, so we need to invest more time to collect data about how easy our application is.
* We claim that the UGCA app runs on both platform, Android and iOS due to using cross-platform technologies, but we did not test it on all versions of both Operating systems. This requires us to invest time and resources to ensure combability on at least a high percentage of users.
* Security and privacy are also major features in this matter. Users care a lot about who can monitor their data over the network and whether it is stored as encrypted data or not. We might need to do extra work to ensure that we can provide those features.

In the near future the project should be tested and we should collect feedback for its users. There are many new areas to investigate and implement new futures to not only the mobile application but all the systems. Which we will state as follow:

* We need to work on the features mentioned above that we did not accomplish yet due to the reasons we stated.
* We need to work on the UI/UX to ensure ease of use.
* We also need to consider accessibility for students with special needs.
* Another is multilanguage support for Arabic, English, French, etc.
* Video conference is an added value to the application, and we believe it is going to be a good feature to help tutors.
* Compatibility with current systems of the university to ease the integration.
* Documentation and resources for the application.

Chapter Six: Conclusions

Communication applications were created to ease its user’s life by breaking the distance barriers, nowadays we rely on those applications more than ever before.

During the work on this project, we had to do lots of things, meeting stakeholders, learning, applying, changing, re-planning, comparing technologies together to find out which best suits our needs and many more.

The project started with sketches and ideas on papers to a working product. We started by collecting information from end-users, we used a Scrum approach to manage the project, we did the backend using PHP with Laravel framework and MySQL for the database; the frontend was divided into two parts, the first part is for the admin which has been done using HTML, CSS, and JavaScript, the second part which is a mobile application for students and tutors has been done using Dart with Flutter framework. The mobile application is connected to the backend using API that is provided using Laravel. For the chat functionality to work at its best we used a Node.JS server with Socket.IO library.

For the project to be a good alternative to the current systems, we must continue the journey of learning, by applying what we have learned over the past years or will learn in the following years. and we must contribute back to the open-source community

# References

Alferd, S. (2019, 12 3). *Flutter vs Xamarin vs React Native — Let the Battle Begin!* Retrieved from Medium: https://medium.com/@stenalferd/flutter-vs-xamarin-vs-react-native-let-the-battle-begin-d3e783bb4bf1

AMC College. (n.d.). *Chat Apps For Mobile Platform.* Advanced Micro Systems Sdn Bhd.

Bootstrap. (2020, 04 16). *Bootstrap Docs*. Retrieved from Bootstrap: https://getbootstrap.com/docs/4.4/getting-started/introduction/

ERICKSON, C. (2012, 09 21). *A Brief History of Text Messaging*. Retrieved from Mashable: https://mashable.com/2012/09/21/text-messaging-history/

Facebook. (2020, 04 16). *About Facebook*. Retrieved from Facebook: https://about.fb.com/

Flutter. (2020, 04 16). *Flutter Documentation*. Retrieved from Flutter: https://flutter.dev/docs

GitHub. (2020, 04 16). *GitHub*. Retrieved from GitHub: https://github.com

Hosseini, P. (2018). *Flutter For Absolute Beginners.* Fatemeh Tajik.

Kumar, S. (2020, 04 16). *MVC Design Pattern*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/mvc-design-pattern/

Laravel. (2020, 04 16). *Docs*. Retrieved from Laravel: https://laravel.com/docs/7.x

Lombardi, A. (2015). *WebSocket Lightweight Client-Server Communications.* O'Reilly Media.

Mozilla. (2020, 04 16). *The WebSocket API (WebSockets)*. Retrieved from MDN Web Docs: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\_API

OpenJS Foundation. (2020, 04 16). *Docs*. Retrieved from Node.JS: https://nodejs.org/en/docs/

PHP. (2020, 04 16). *Documentation*. Retrieved from PHP: https://www.php.net/docs.php

Quigley, E. (2018, 12 11). *The Benefits of Using an LMS Forum*. Retrieved from LearnUpon: https://www.learnupon.com/blog/lms-forum-benefits/

Schwaber, K. (2004). *Agile Project Management with Scrum.* Microsoft Press.

Scrum.org. (2020, 04 16). *WHAT IS SCRUM?* Retrieved from Scrum.org: https://www.scrum.org/resources/what-is-scrum

Smallcombe, M. (2017, 10 10). *The SQL vs NoSQL Difference: MySQL vs MongoDB*. Retrieved from Xplenty: https://www.xplenty.com/blog/the-sql-vs-nosql-difference/

Socket.IO . (2020, 04 16). *What Socket.IO is*. Retrieved from Socket.IO : https://socket.io/docs/

Telegram. (2020, 04 16). *Telegram FAQ*. Retrieved from Telegram: https://telegram.org/faq

W3Techs. (2020, 04 16). *Usage statistics of PHP for websites*. Retrieved from W3Techs: https://w3techs.com/technologies/details/pl-php

WhatsApp. (2020, 04 16). *About WhatsApp*. Retrieved from WhatsApp: https://www.whatsapp.com/about/

# Appendix:

The UGCA project would not have been accomplished without the use of the following open-source projects and tools:

* Adobe XD (<https://www.adobe.com/products/xd.html>)
* Android Studio (<https://developer.android.com/studio>)
* Bootstrap ([https://getbootstrap.com](https://getbootstrap.com/))
* Composer (<https://getcomposer.org>)
* Datatables (<https://datatables.net/>)
* Draw.IO (<https://www.draw.io/>)
* Flutter (<https://flutter.dev/>)
* Font Awesome (<https://fontawesome.com/>)
* Git (<https://git-scm.com/>)
* GitHub (<https://github.com/>)
* Jira (<https://www.atlassian.com/software/jira>)
* jQuery (<https://jquery.com/>)
* Laravel (<https://laravel.com/>)
* Material Design (<https://material.io/design/>)
* MySQL (<https://www.mysql.com/>)
* Node.JS (<https://nodejs.org/>)
* Postman (<https://www.postman.com/>)
* Select2 (<https://select2.org/>)
* Socket.IO (<https://socket.io/>)
* Trello (<https://trello.com/>)
* Unsplash (<https://unsplash.com/>)
* Visual Studio Code (<https://code.visualstudio.com/>)
* WAMP (<https://sourceforge.net/projects/wampserver/>)

During our search, we found a technique for time management called The Pomodoro Technique, that has been developed by Francesco Cirilo late in the 1980s. the technique involves the following steps:

1. Create a to-do list and give each item a priority
2. Work on the top priority item for 25 minutes.
3. Take a break for 5 minutes
4. Go back to the same item until it is completed, and move to the next one.
5. After completing 4 pomodoros (25 minutes for each period) take a longer break of 15 to 30 minutes.
6. Repeat until the end of the day.

This technique helped us a lot to accomplish many tasks from the small ones to the complicated tasks.