Fall 2020

Student MeetUp: An AJS Project

Alejandro Zeballos
CCT College Dublin

Suelen De Campos Fiorelo
CCT College Dublin

James Harris
CCT College Dublin

AJS Project

STUDENT MEETUP

Alejandro Zeballos 2017351

Suelen De Campos Fiorelo 2017225

James Harris 2017396

School year 2019/2020

Abstract

This report focuses on our perceived lack of applications that can assist students in finding suitable environments in which they can study together. Our research has shown that while there are applications that are dedicated to helping students to study, none that we have found help students connect with one another, for the sole purpose of studying.

With our application, we intend to create a platform that will enable students to connect with each other, on both a website and a mobile application. We hope to give students the ability to create and join study sessions, as well as provide them with the opportunity to chat with people outside of classrooms in a distraction-free environment. This will help students to focus in addition to creating a larger network of people to interact with within their chosen course and when they enter the professional workspace. Our aims are supported by research that shows that organised group work facilitates and aids student learning

Acknowledgements

Thank you to all the staff at CCT, and especially John Snel and Graham Glanville for the time and commitment they have given us during the year. Thank you to all those involved in our research.

Contents

Introduction

During our studies at CCT, we have found that it is difficult to study when working on our own, be it due to loss of concentration or other factors such as work or family obligations that prevent us from focusing on studying. We realized that when we were in groups it was easier to focus on our work, this also allowed us to ask for help from our peers and also share any of our insights with them. We also have found that many students find it hard to form study groups for a lack of a study network.

Our idea to alleviate some of the pressure in forming these study groups is a mobile and web application which helps those students looking for such a group to find it. Our application will enable students to form study groups where they will be able to share work, study and experiences. Based on mutual interests users will be able to reach out and connect with a network of their peers. The benefits of group studying have been researched, 'McCabe, J. A., & Lummis, S. N. (2018)' outline that studying together increases the average grade of all those who actively participate, In this document, we hope to show the processes which we underwent in the completion of our idea such as the technology we used, the research we did, how we designed our application.

We are showcasing the knowledge gained through the college in web development, project analysis and design, programming, networking, databases, cloud technologies and graphic user interface design for the realization of this project.

Many computing areas are involved in this project, therefore we believe this is a great project and opportunity for us to show all the knowledge that we have learned through these years at CCT college. In our group, we are proud to have diverse skills and strengths thereby we are sure this challenge is manageable. We are a group of three and one is good in programming, other is good in web development and the third one is very good at databases so we complement each other for this project.

It will involve a couple of technologies in the project, for the planning and project management we are using Project Manager, which is an online service that allows us to manage an overview of the system development life cycle.

For the system design, we are going to use Visual Paradigm to draw diagrams (class diagrams, EER, use case diagrams, etc.)

For the website building, we are going to use HTML, CSS, Javascript, all these for the front end. For the back end, we are using PHP as a server-side scripting and Node.js for the chat-app. SQL for querying the database. Our database will be up in the cloud, we are going to use Google Cloud technologies for this purpose.

For the mobile application, we plan to focus only on android, therefore, we are going to use Visual Studio as an IDE (Integrated development environment) and the code will be made using Java, although Kotlin is according Ababei Adrian(2019) the "modern" language for android development, we opt to use Java because of the familiarity we have with it. We are going to use the same database as for the website. As we plan to implement a chat functionality we are going to use the socket.io library which will help us in this matter.

The structure of this document will be in two chapters, the first will focus on the initial design, including requirements, use cases, class diagrams and entity relationship diagrams. This chapter will outline our initial ideas on the application and how we plan to build our platform

Chapter two includes the documentation of the running application, this includes the GUI, databases design, programming, the website structure, the mobile application structure, and the testing that we did on the application.

Followed by the conclusion of our report and the appendices.

1. System Analysis

In the first chapter, we will show - in detail the description of the functionality of the system, which includes the requirements, use case scenarios and diagrams that we will need for building the application.

According to (Kendall and Kendall, 1998),- system analysis - is describing how the system will function by examining the input and processing of data and the output of information.

1.1 Requirements

The list below is a brief summary of the requirements for the Student meetup that we think will be necessary to build the application.

R1 - Registration is needed as a first step in using the application. During registration, the user has to put up his/her name, nickname, course, email, password and a short description of him/herself.

R2 - Login will require the email and password of the user.

R3 - Chatroom, each study session should have a chat room available for users within the group, and for new users to the study session.

R4- Profile information, the user profile should be available to other users to see and should include: name, nickname, course, short description, and a record of study sessions created or joined.

R5 - Profile editing, a user should be able to edit his/her profile changing the name, nickname, course and the description.

R6 - Creating study sessions, the system has to offer a platform for users to post a study session providing the session title, course, time, location, description and the user also can insert tags.

R7 - Joining study sessions, the system should provide a platform for the user to join a group based on the course that he/she is registered on.

R8 - Delete study session function, the user hosting a study session should be able to delete the post at any time.

R9 - Filter system search should be implemented so that the user is able to see only posts related to what they are looking for.

R10 - Report system, any user is able to report another user for whatever reason he/she thinks he/she has to be reported.

R11 - Location services, the user that is creating a study session has to be able to insert the location where the session will occur.

1.2 Actor Catalogue

Figure A-1 shows all the actors in the Student meet up use-case model. The brief descriptions of the actors are given in the subsections that follow the figure.

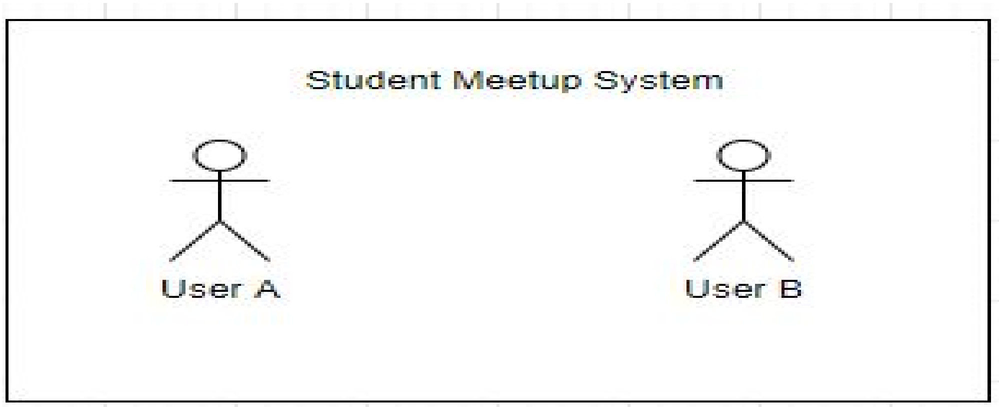


Figure A-1 The actors of the Student meetup system – Visual Paradigm.

1.3 Student meetup system use cases

This section will provide an overview of the use cases and a short description of each use case is about.

1.4 Primary use case

Figure A-2 Shows the primary use cases from Student Meetup use-case model. The brief descriptions of the use cases are given in the subsections that follow.

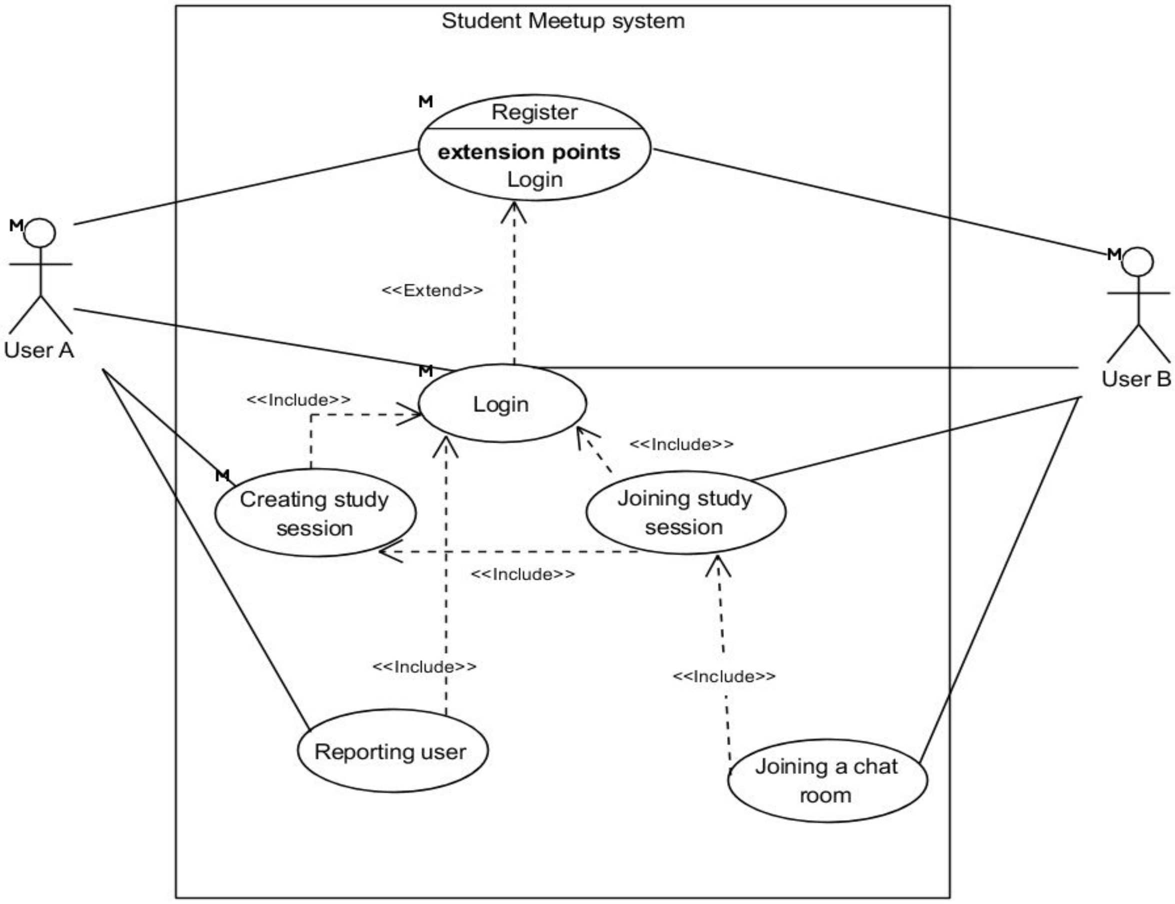


Figure A-2 The primary uses cases of the Student Meetup System – Visual Paradigm.

.

1.4.1 Register

This use case describes how a student has to register to the system for then log in if the user chooses to.

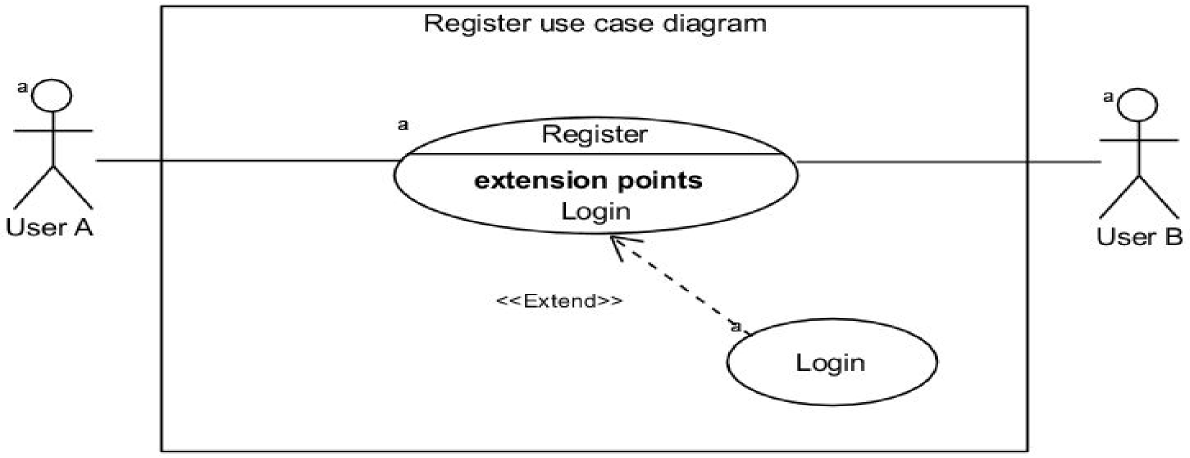


Figure A-3 Register use case diagram – Visual Paradigm.

Use case: Register

Actors: Users A or User B

Goal: Register as a new user.

Overview: When the student goes to the website/mobile application page, he/she will click on REGISTER, where they will have to enter the name, nickname, course, email, password and a short description of him/her self to register. After registration, they will be sent to the home page and will have the option to click on LOGIN.

Cross-Reference:

R1-R2

Typical course of events:

Actors actions:	System Response:
1 – The student goes to the application page. 2 – The student clicks on the REGISTER button. 4 – The student enters the name, course, email, password and a short description of him/her. 6- The user clicks on the LOGIN button.	3 – Displays the Register window. 5 - The system displays a message saying “ You are now registered” and The application will send him/her back to the home page. 7 - Initiate ‘Login’

Alternative course:

Actors actions:	System Response:
Step 4	The student is already registered in the application, He/She won’t be able to register again and he will be sent to the Login section instead of a register page.

1.4.2 Login

This use case describes the information the user has to input in order to log in.

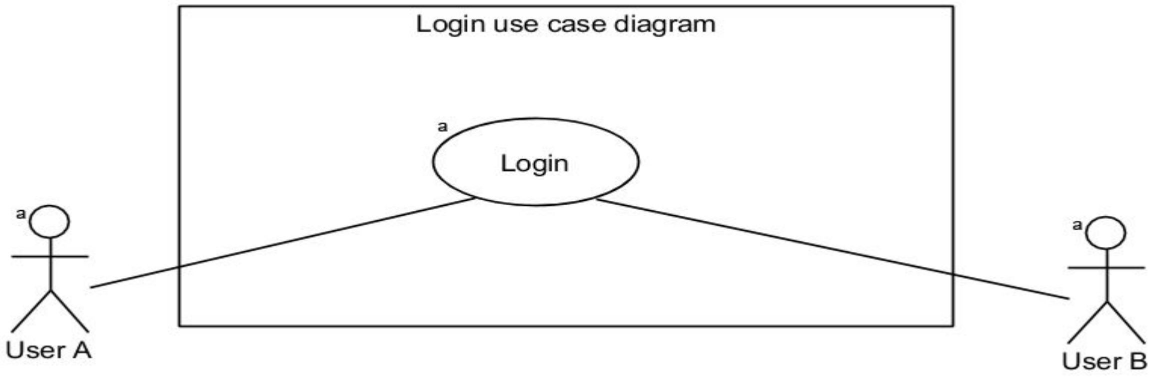


Figure A-4 Login use case diagram – Visual Paradigm.

Use case: Login

Actors: User A or User B

Goal: Login into the system.

Overview: When the user wants to login into the system they will have the button "Login" displayed in the home page and when clicked the user will be asked to insert the email and password.

Cross-Reference:
R1-R2
Typical course of events:

Actors actions:	System Response:
1 – The student goes to the application page. 2 – The student clicks the login button. 4 – The student enters the email and password and then clicks on login.	3 – Displays the Login window. 5 - The system sends the user to the user's welcome page.

Alternative course:

Actors actions:	System Response:
Step 4	The system does not recognize the user and asks to fill the fields again or click the REGISTER button if the user did not do so yet.

1.4.3 Creating study session

This use case describes what a student has to do for hosting a study session and how the system will perform this task

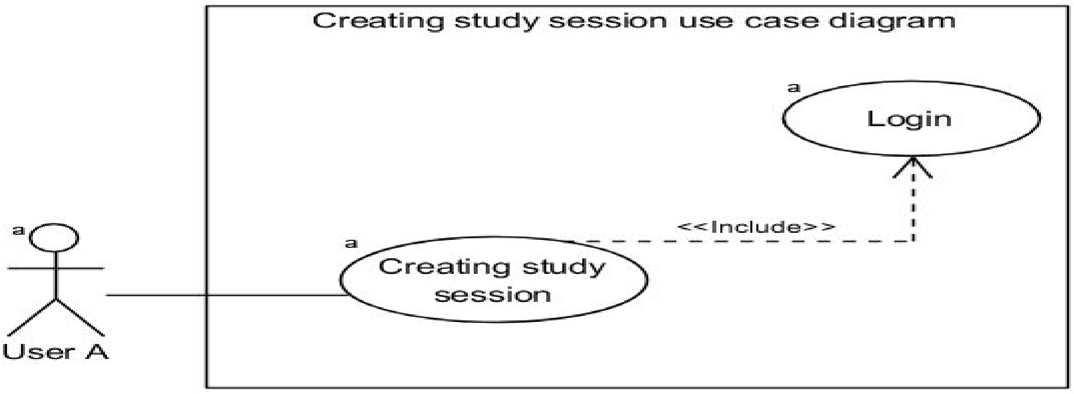


Figure A-5 Creating study session use case diagram – Visual Paradigm.

Use case: Creating study session

Actors: User

Goal: Create a Study session

Overview: Once the user has logged in to the application, he/she click on CREATE STUDY SESSION button where he will be asked to insert the Title, time, course, location, tags and description then the study session will be available for users to see and a notification will be sent to users email within the same career.

Cross-Reference:
R6-R2
Typical course of events:

Actors actions:	System Response:
1 - The user logs into the application. 3 - The user clicks on CREATE STUDY SESSION. 5- The user enters the title, name, course, location, tags and description then clicks on the submit button.	2 - The system sends the user to the welcome page. 4 - Displays the create study session windows. 6- The System creates the group

Alternative course:

Actors actions:	System Response:
5- The student did not fill all the fields at the submitting stage.	6- Asks the user to fill empty fields.

1.4.4 Joining study session

This use case describes the steps a student has to follow for participating in a study session that has been already created either by login into their account or clicking in an invitation link.

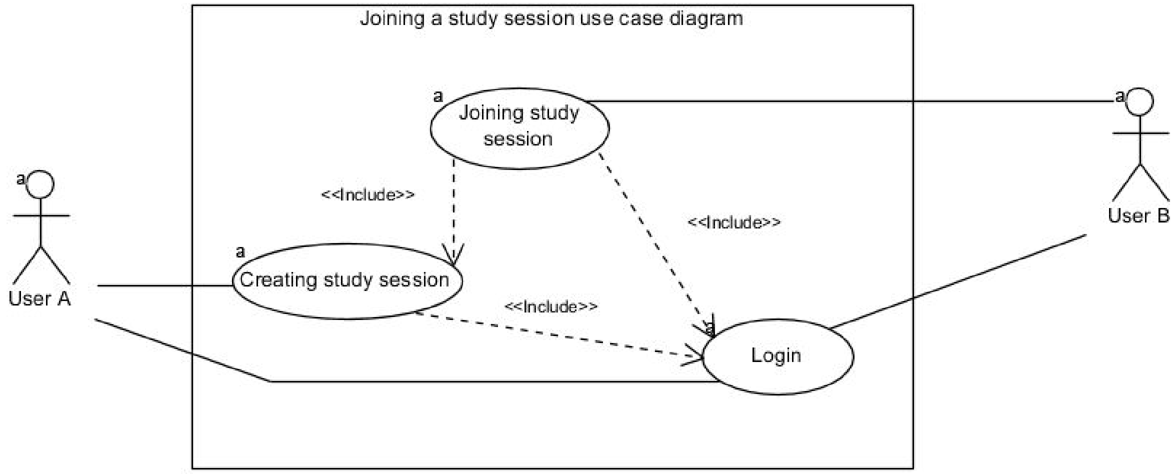


Figure A-6 Joining a study session use case diagram – Visual Paradigm.

Use case: Joining a study session

Actors: User

Goal: To join a study session that another user has posted.

Overview:

When a user wants to join a study session he/she will search for the subject he/she wants to study using the filter system. The system will then provide posts matching the requirements of their search. The user will then select which study session they want to join. He/she will then have access to the information of that study session as well as the chat room of that study session. They also have the ability to leave any session they have joined.

Cross-Reference:

R2-R7-R9

Typical course of events:

Actors actions:	System Response:
1.User Logs in to the system 3. User enters the search function. 5. User refines the search parameters. 7. User selects the session he/she wants to join. 9. User is able to interact with the chat function. 10 User can leave the session.	2. System logs in user and sends the user to the homepage 4. System shows available study sessions. 6. System updates to show study sessions matching the parameters. 8. System adds the user to the session and displays the hidden information to the user. 11. System closes session after set time and removes all users.

Alternative course:

Actors actions:	System Response:
Step 3. User inputs search parameters Step 5. User is already in a session Step 1,3,5,7 User clicks in the study session link invitation provided to the email.	Step 4. System cannot find any posts matching the refinements. System shows an error message. Step 6. System does not add the user to session and do not give access to session information.

1.4.5 Joining a chat room

This use case describes the iteration of the user with the system interface since he/she enters into the chat room.

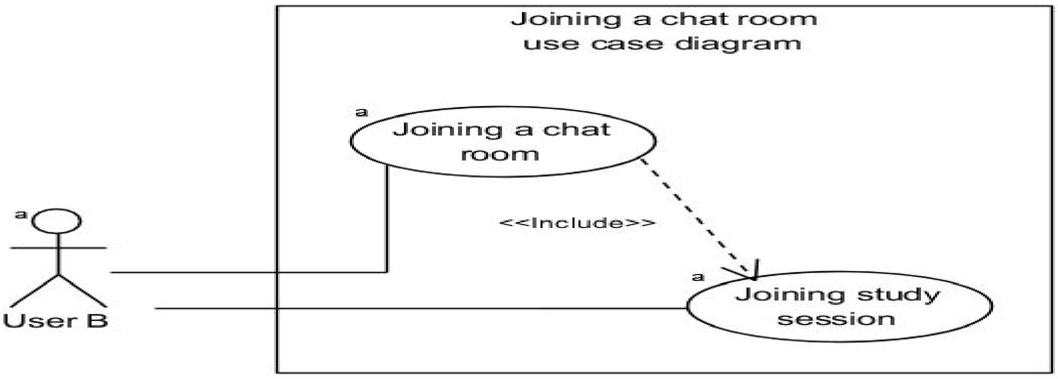


Figure A-7 Joining a chat room use case diagram – Visual Paradigm.

Use case: Joining a chat room

Actors: User A, User B

Goal: To Chat between users

Overview:

A user enters into the chat room, he is then able to post messages that other users in the chat room are able to see.

Cross-Reference:

R2-R3-R9

Typical course of events:

Actors actions:	System Response:

1. User A logs into the system 3. User A joins the chat room. 5. User A posts a message to the chat room. 7. User B posts a message to the chat room. 9. User B leaves the chat room 11..User delete session	2. System logs in user and sends user to the homepage 4. System displays the message on the chat board that User A has joined the chat room. 6. System displays User A’s message to other users in the chat room. 8. System displays User B’s message to other users in the chat room. 10.System displays a message in the chat room that User A has left the chat. 12. System destroys the chat room.

Alternative course:

Actors actions:	System Response:

1.4.6 Reporting user

This use case shows what steps a user has to complete in order to report to other users for bad behavior within the system community.

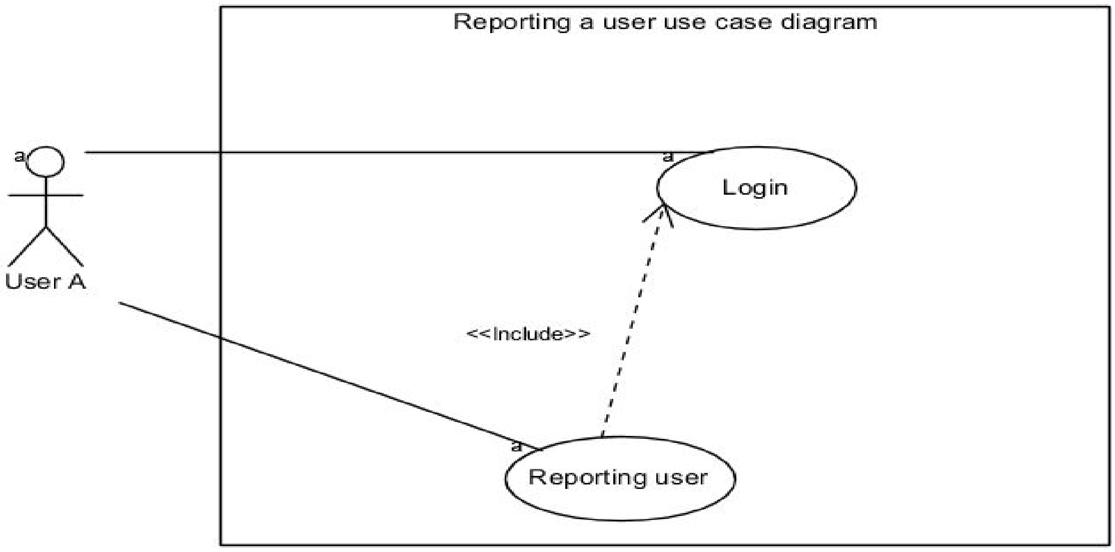


Figure A-8 Reporting a user use case diagram – Visual Paradigm.

Use case: Reporting a user

Actors:User

Goal: Report a user

Overview: A user is able to report another user when he/she has performed an inappropriate conduct within the application, chat room, or in the study session.

Cross-Reference:
R11-R2
Typical course of events:

Actors actions:	System Response:
1 – The user gets into the profile of the user he/she wants to report to. 3 – The user clicks on the REPORT button, 5 – The user fills the report form with a description of why he/she is reporting the user in question and clicks SUBMIT REPORT button.	2 – Displays the profile of the user to report. 4 - Displays a report form. 6 - Displays a message: “Report submitted, thanks for helping us to better this application”

Alternative course:

Actors actions:	System Response:
5- The student did not fill all the fields at the submitting stage.	6- System asks the user to fill in the form before submitting.

1.5 Student meetup system class diagram

The StudentMeet-up class diagram (Figure A-9) Defines the overall structure of the system, and the structure of every object in the system and the interaction between them in a high level view.

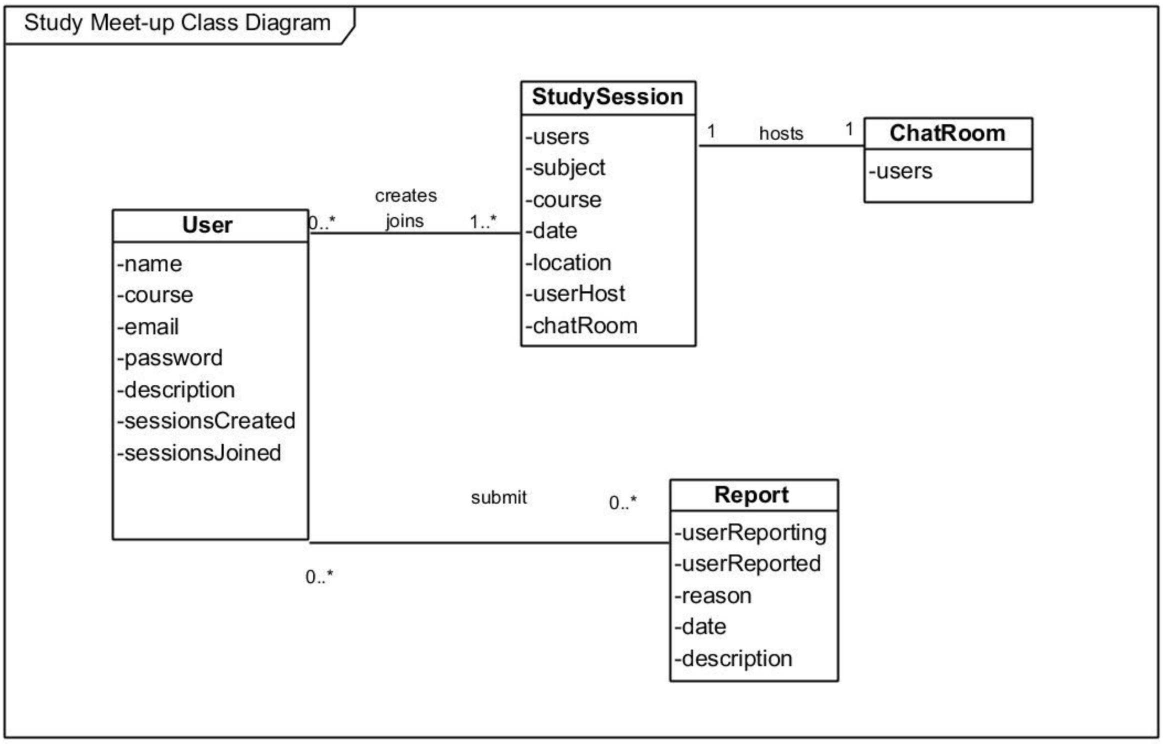


Figure A-9 Classdiagram – Visual Paradigm.

1.6 Entity-relationship diagram

In the figure A-10 shows the entity-relationship diagram of the database where the data of the web and mobile application will be stored.

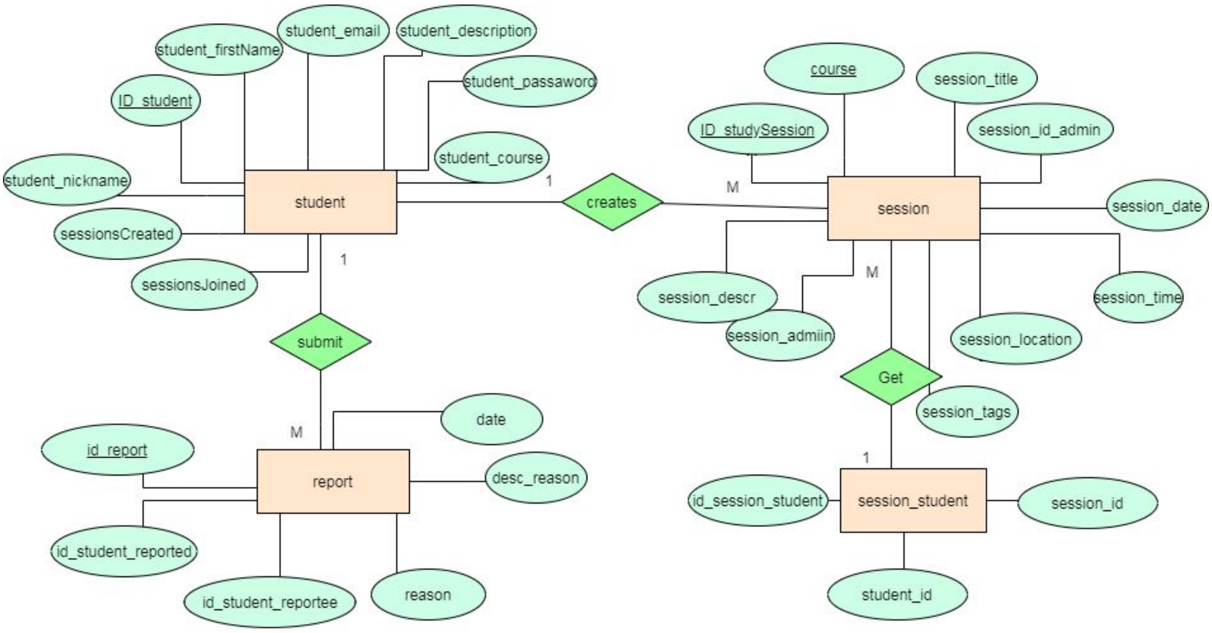


Figure A-10 Enhanced entity-relationship diagram –draw.io

2. System Design

The second chapter will describe the system design which we will talk about how the application will be implemented(User interface design, Function design and data design).

2.1 Technologies

A list of Technologies used in the creation of our application.

- Microsoft Visual Studio Code
- XAMPP
- Android Studio
- Google Cloud Platform
- Google Drive
- BootStrap Studio
- InVision Studio
- Heroku
- Draw.io
- Postman

- Github
- Java
- PHP
- MySQL
- HTML & CSS
- JavaScript
- Node.js
- Socket.IO
- Bitnami

2.2 Android Design

The application will use Java as the development language, XML will be used for creating the layouts, socket.IO will be used for the client side chat application. Node.js will be used for the server side part of the chat application Android studio will be used as the primary IDE for the development of the mobile application.

We have chosen the android platform because It can be developed using Java which is a language we feel comfortable with.

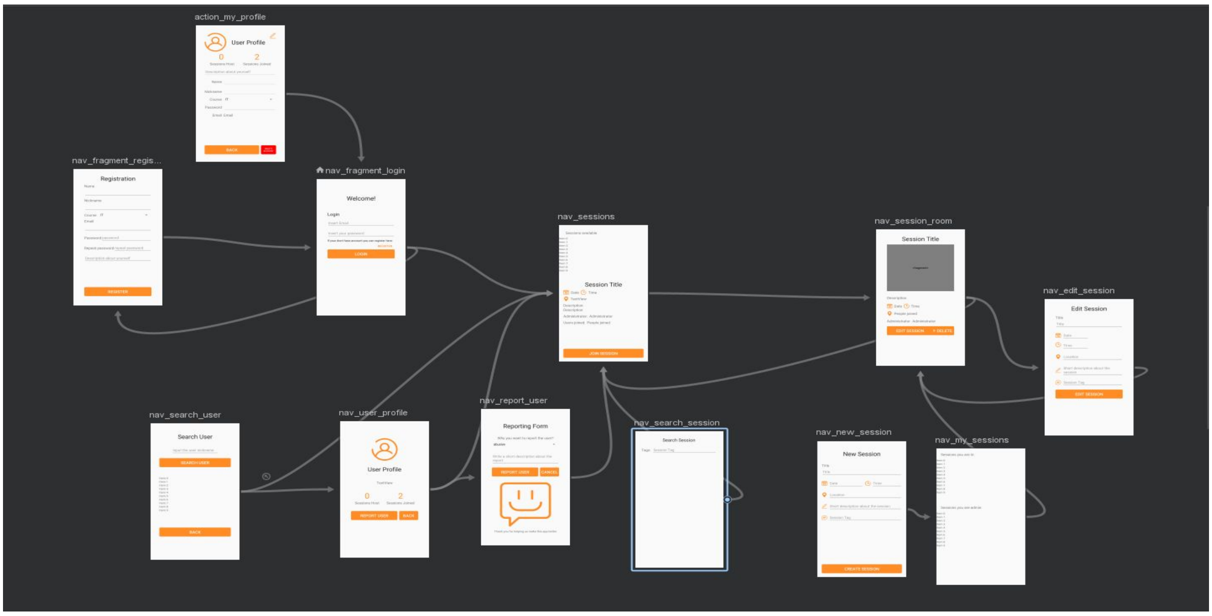


Figure B-1 Navigation view of android application – Android Studio

2.3 Website Design

The website will use HTML, CSS, JavaScript and PHP. The HTML & CSS will be used to create and style the website. PHP connects the website to the database and enables the website to query for data used in the operations. Javascript will be used to run the chat application within the website, Socket.IO will be used in the chat function of our web application, It will enable real time, bi-directional communication between web client and server. Node.js will allow for Server side communication between clients. Visual studio code will be used as the primary IDE for the website.

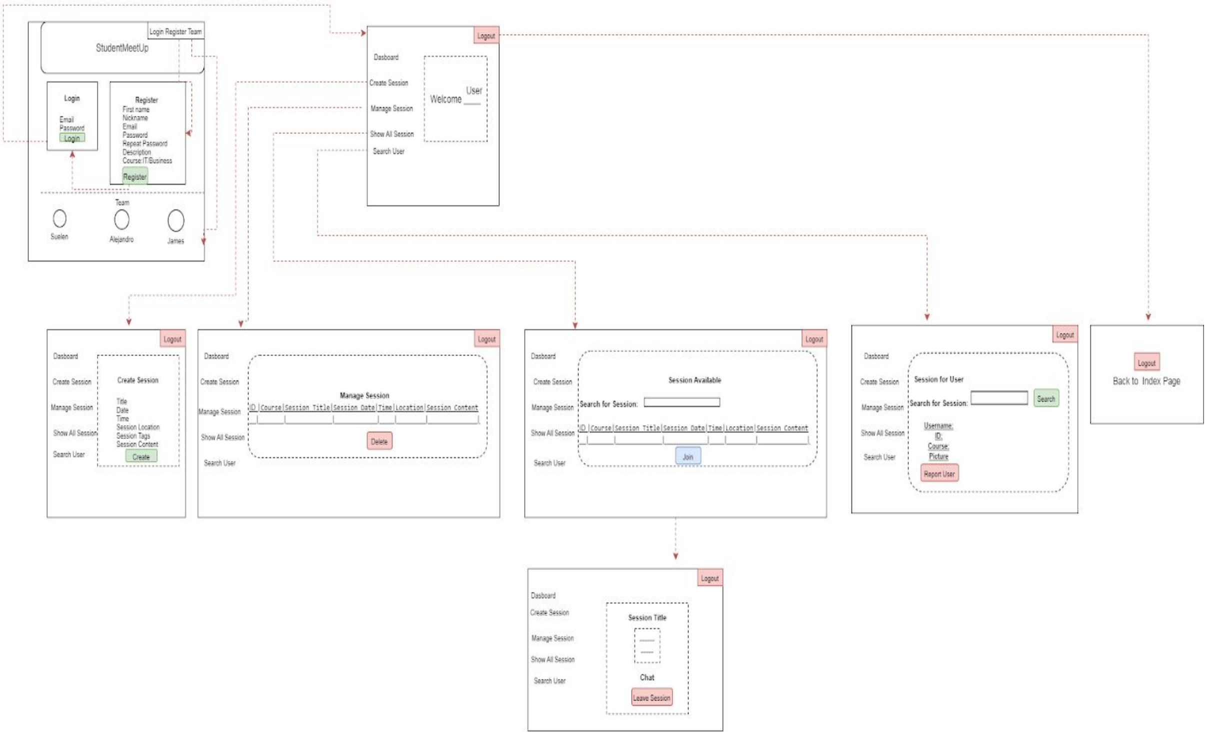


Figure B-2 Class Diagram - draw.io

2.4 Database design

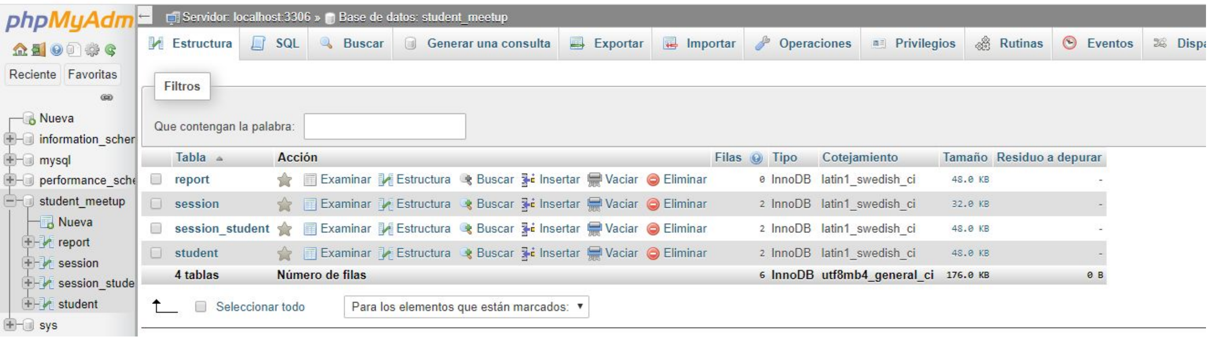
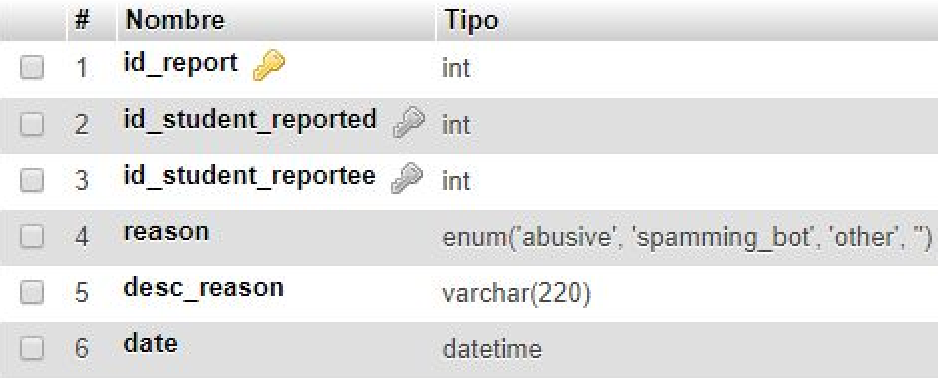


Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
report	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48.0 KB	-
session	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	latin1_swedish_ci	32.0 KB	-
session_student	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	latin1_swedish_ci	48.0 KB	-
student	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	latin1_swedish_ci	48.0 KB	-
4 tablas	Número de filas	6	InnoDB	utf8mb4_general_ci	176.0 KB	0 B

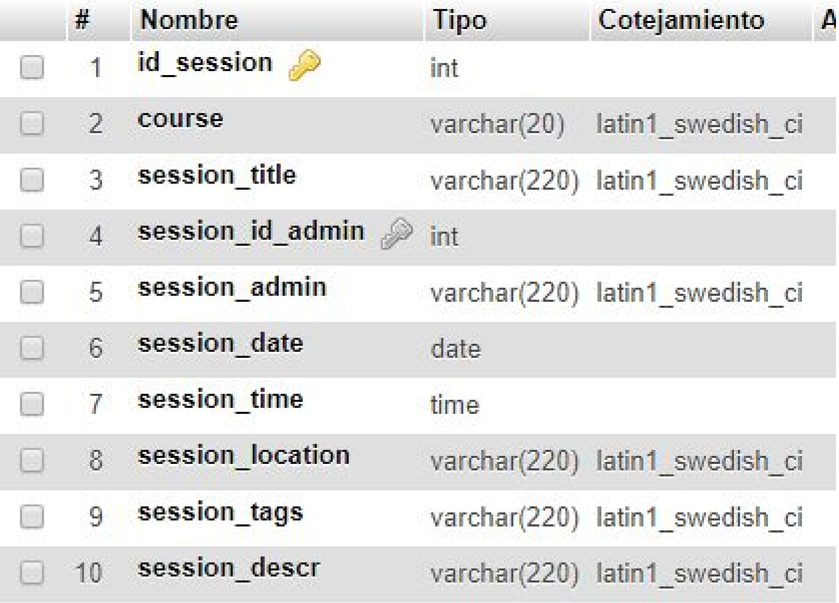
Seleccionar todo   Para los elementos que están marcados:

Figure B-3 Student Meetup database structure – phpMyAdmin



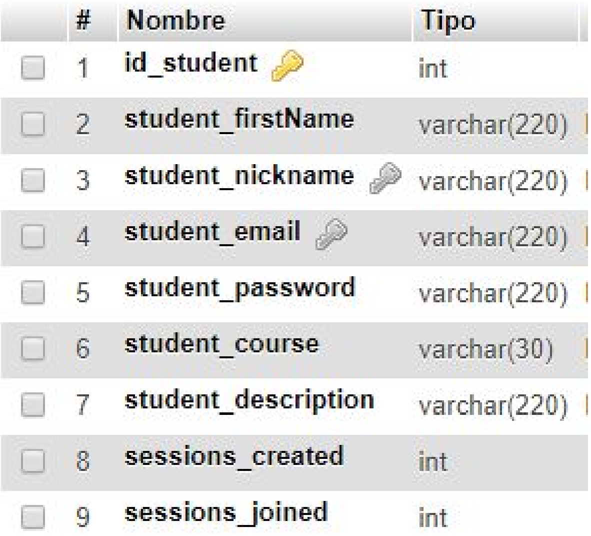
	#	Nombre	Tipo
☐	1	id_report 🔑	int
☐	2	id_student_reported 🔑	int
☐	3	id_student_reportee 🔑	int
☐	4	reason	enum('abusive', 'spamming_bot', 'other', '')
☐	5	desc_reason	varchar(220)
☐	6	date	datetime

Figure B-4 Student Meetup Report Table structure – phpMyAdmin



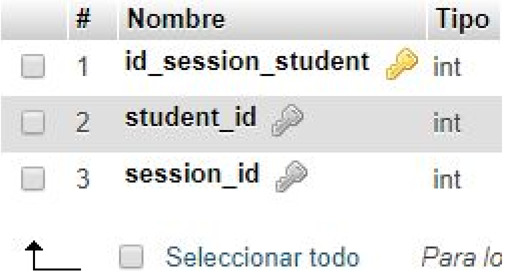
	#	Nombre	Tipo	Cotejamiento	A
☐	1	id_session 🔑	int		
☐	2	course	varchar(20)	latin1_swedish_ci	
☐	3	session_title	varchar(220)	latin1_swedish_ci	
☐	4	session_id_admin 🔑	int		
☐	5	session_admin	varchar(220)	latin1_swedish_ci	
☐	6	session_date	date		
☐	7	session_time	time		
☐	8	session_location	varchar(220)	latin1_swedish_ci	
☐	9	session_tags	varchar(220)	latin1_swedish_ci	
☐	10	session_descr	varchar(220)	latin1_swedish_ci	

Figure B-5 Student Meetup Session Table structure – phpMyAdmin



	#	Nombre	Tipo
□	1	id_student 🔑	int
□	2	student_firstName	varchar(220)
□	3	student_nickname 🔑	varchar(220)
□	4	student_email 🔑	varchar(220)
□	5	student_password	varchar(220)
□	6	student_course	varchar(30)
□	7	student_description	varchar(220)
□	8	sessions_created	int
□	9	sessions_joined	int

Figure B-6 Student Meetup Student Table structure – phpMyAdmin



	#	Nombre	Tipo
□	1	id_session_student 🔑	int
□	2	student_id 🔑	int
□	3	session_id 🔑	int

Figure B-7 Student Meetup Session_Student many to many Table structure – phpMyAdmin

The application will use SQL as the database query language. XAMPP will be used to test the database, XAMPP is free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. The application will then be deployed to a google cloud server where we have used a tutorial made by (apachefriends.org. 2020) to successfully implement our XAMPP server in the cloud. Google cloud 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, Gmail and YouTube.

2.5 Implementation of system

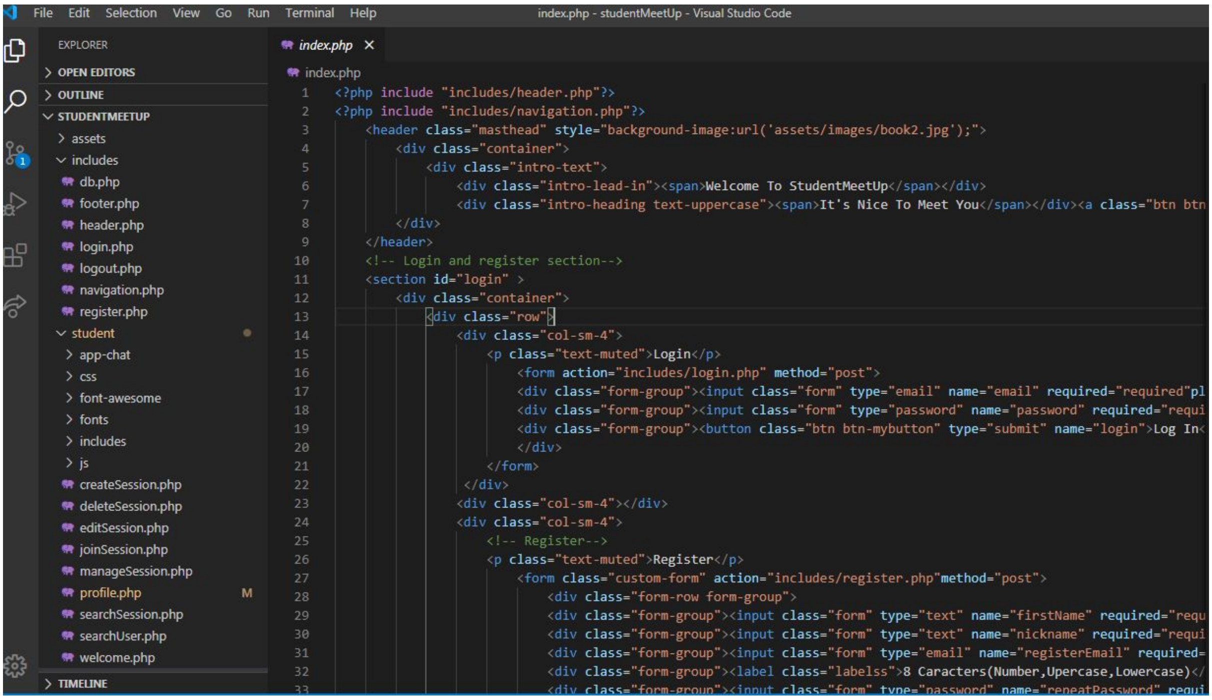
This section will go through the process of interacting with the application.

2.5.1 Website

The website uses PHP for the server-side,which means that it runs on the web server software,which is normally going to serve HTML to the visitor. We chose PHP because it was one of the languages we used in our second year of college.

Node.js was also chosen on the website for the server-side, using the socket.io which is a javascript library for realtime web applications, that permit communication between servers and clients.

The node.js was deployed on heroku.com , while the Php files and the database were hosted on google cloud.



```
STUDENTMEETUP
  assets
  includes
    db.php
    footer.php
    header.php
    login.php
    logout.php
    navigation.php
    register.php
  student
    app-chat
    css
    font-awesome
    fonts
    includes
    js
    createSession.php
    deleteSession.php
    editSession.php
    joinSession.php
    manageSession.php
    profile.php   M
    searchSession.php
    searchUser.php
    welcome.php
```

```
<?php include "includes/header.php"?>
<?php include "includes/navigation.php"?>
    <header class="masthead" style="background-image:url('assets/images/book2.jpg');">
        <div class="container">
            <div class="intro-text">
                <div class="intro-lead-in"><span>Welcome To StudentMeetUp</span></div>
                <div class="intro-heading text-uppercase"><span>It's Nice To Meet You</span></div><a class="btn btn
        </div>
    </header>
    <!-- Login and register section-->
    <section id="login" >
        <div class="container">
            <div class="row">
                <div class="col-sm-4">
                    <p class="text-muted">Login</p>
                        <form action="includes/login.php" method="post">
                        <div class="form-group"><input class="form" type="email" name="email" required="required"pl
                        <div class="form-group"><input class="form" type="password" name="password" required="requi
                        <div class="form-group"><button class="btn btn-mybutton" type="submit" name="login">Log In<
                        </div>
                    </form>
                 </div>
                <div class="col-sm-4"></div>
                <div class="col-sm-4">
                    <!-- Register-->
                    <p class="text-muted">Register</p>
                        <form class="custom-form" action="includes/register.php"method="post">
                            <div class="form-row form-group">
                            <div class="form-group"><input class="form" type="text" name="firstName" required="requ
                            <div class="form-group"><input class="form" type="text" name="nickname" required="requi
                            <div class="form-group"><input class="form" type="email" name="registerEmail" required=
                            <div class="form-group"><label class="labelss">8 Caracters(Number,Upercase,Lowercase)</
                            <div class="form-group"><input class="form" type="password" name="repeatPassword" requi
```

Figure B-8 Folder Structure –Visual Code

This is the folder that details the structure of the website as seen in the Figure B-8 above.

A simple BootStrap template was used for the website's initial design. The home page displayed a simple background image, and the user has the option of login in, registering and viewing team photos.

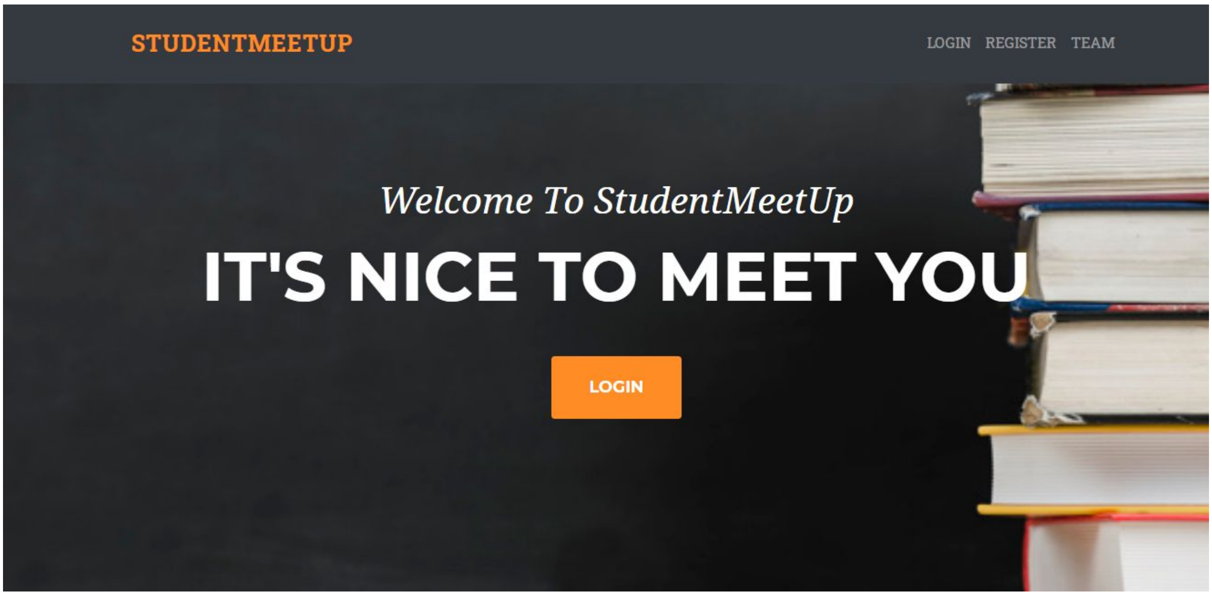


Figure B-9 Welcome page website – Student Meetup

After clicking the register option, the user is required to insert a name, nickname, email, password(the user has to repeat the password), a short description about themselves and the course they are enrolled in.This data is then saved to the database in the appropriate fields through PHP. Once the user wishes to login they must enter the email and password they entered at registration.

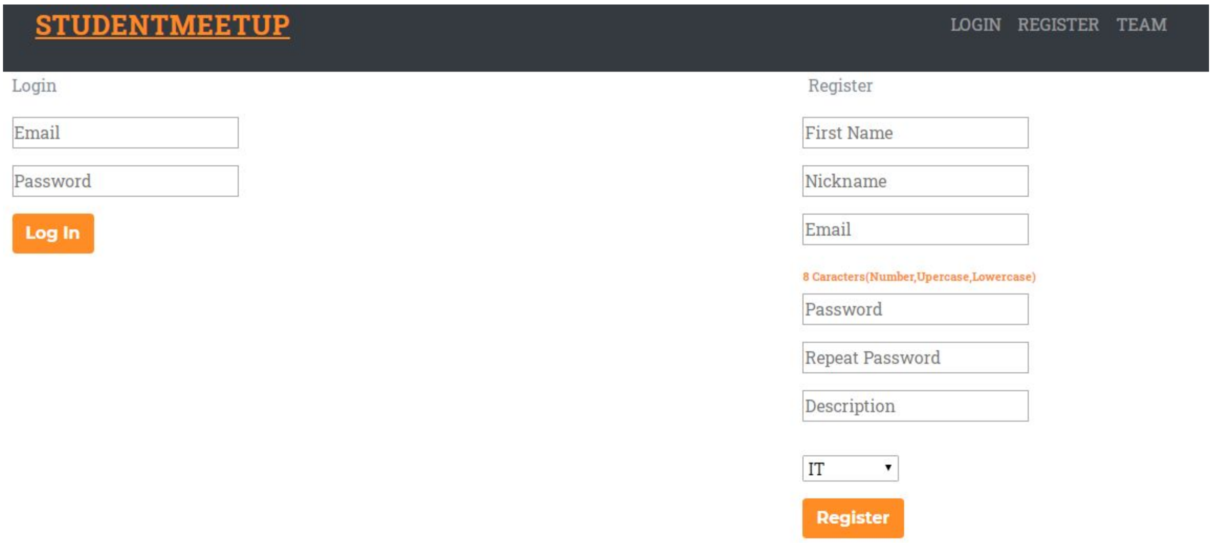
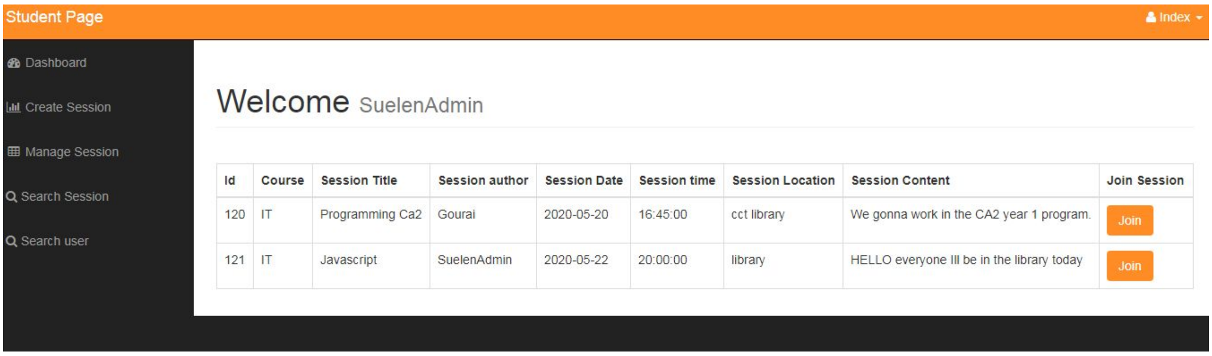


Figure B-10 Login and Register page website – Student Meetup

When a user has successfully logged in, they will be brought to their dashboard where they can see the available sessions. This includes the session title, session author, when the session will happen(session date), the location of the session and a short description of the session. The user will also have the option to join the session.

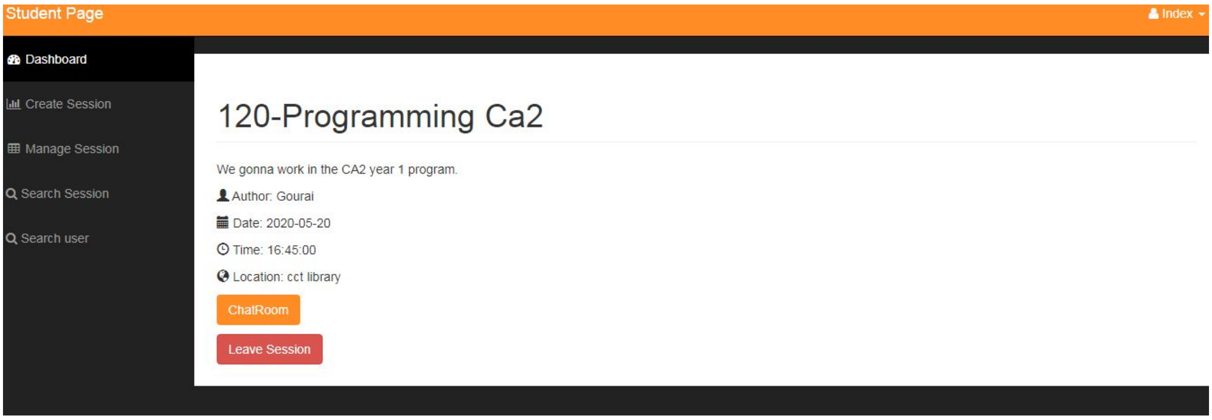


Welcome SuelenAdmin

Id	Course	Session Title	Session author	Session Date	Session time	Session Location	Session Content	Join Session
120	IT	Programming Ca2	Gourai	2020-05-20	16:45:00	cct library	We gonna work in the CA2 year 1 program.	Join
121	IT	Javascript	SuelenAdmin	2020-05-22	20:00:00	library	HELLO everyone Ill be in the library today	Join

Figure B-11 Dashboard page website – Student Meetup

When a user clicks on the join session button they will be brought to this page, where they will be able to join the chat room or leave the session.



120-Programming Ca2

We gonna work in the CA2 year 1 program.

Author: Gourai

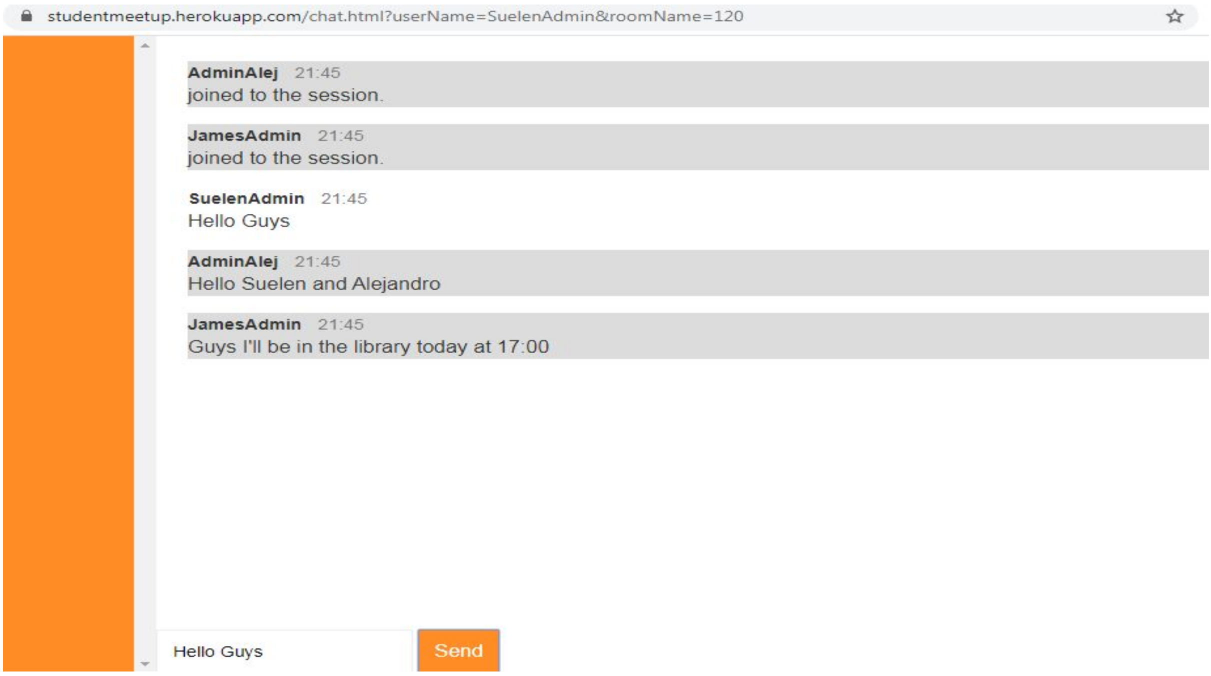
Date: 2020-05-20

Time: 16:45:00

Location: cct library

Figure B-12 Session page website – Student Meetup

Once a user enters the chat room they will be able to interact with the other users who have entered the chat room. When a user enters or leaves a chat room it is announced in the chat so that other users are informed.



studentmeetup.herokuapp.com/chat.html?userName=SuelenAdmin&roomName=120

AdminAlej 21:45
joined to the session.

JamesAdmin 21:45
joined to the session.

SuelenAdmin 21:45
Hello Guys

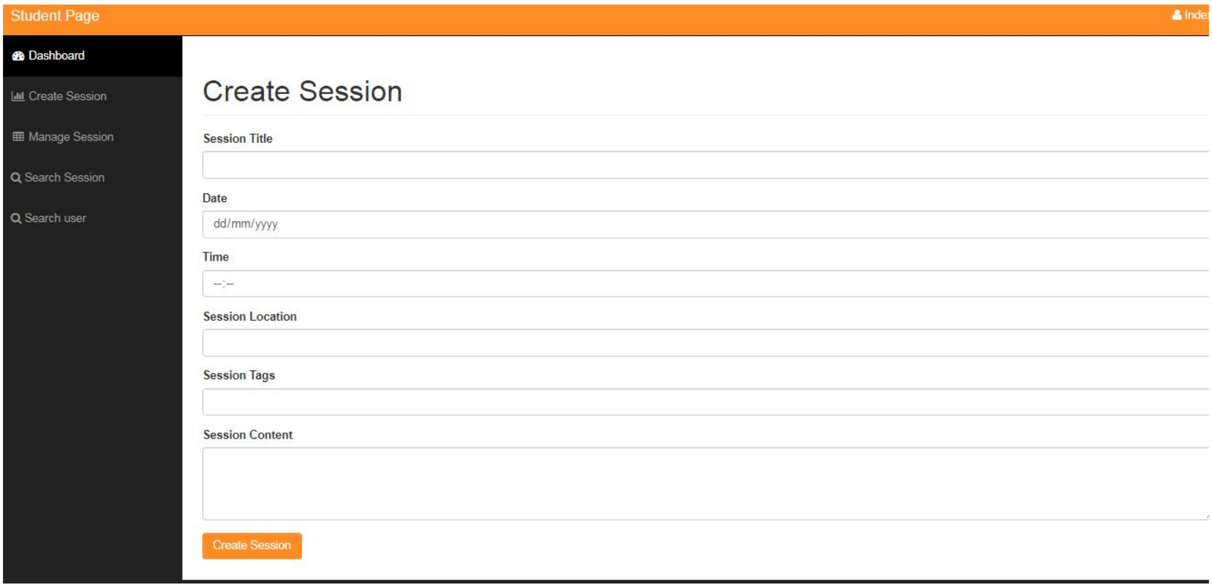
AdminAlej 21:45
Hello Suelen and Alejandro

JamesAdmin 21:45
Guys I'll be in the library today at 17:00

Hello Guys

Figure B-13 Chat Room page – Student Meetup

If a user wishes to create their own session they must click on the create session tab on the left of the web page. When clicked it will bring them to the create session page. This will require them to input the information shown in the image below. Once the user has filled out the form they click on the create session button. This then saves the data to the databases and makes it available to other users to join.



Create Session

Session Title

Date
dd/mm/yyyy

Time
--:--

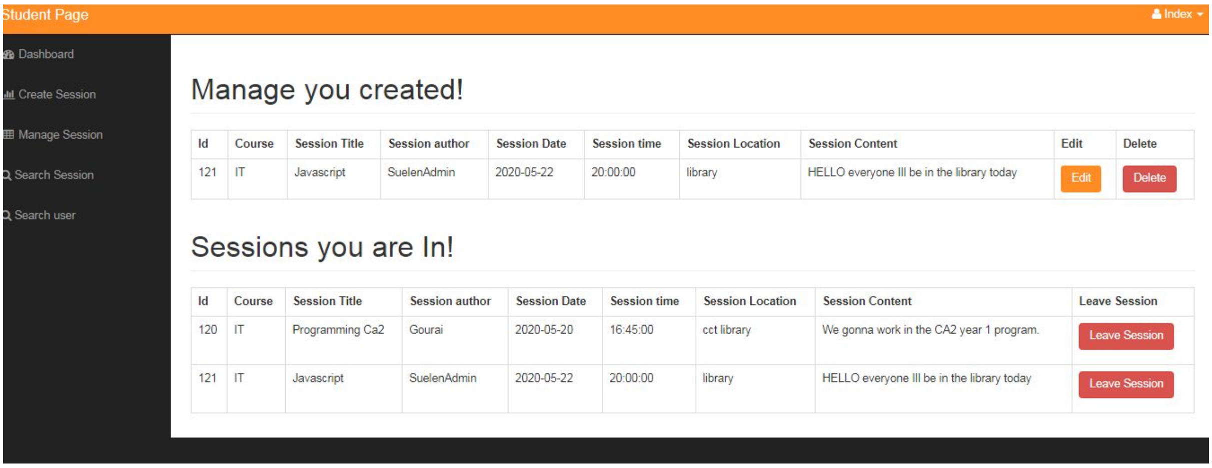
Session Location

Session Tags

Session Content

Figure B-14 Create Session page – Student Meetup

A user can manage their created sessions in the manage session tab. Here they will be able to edit created sessions or delete them. It also will display the sessions the user is in.



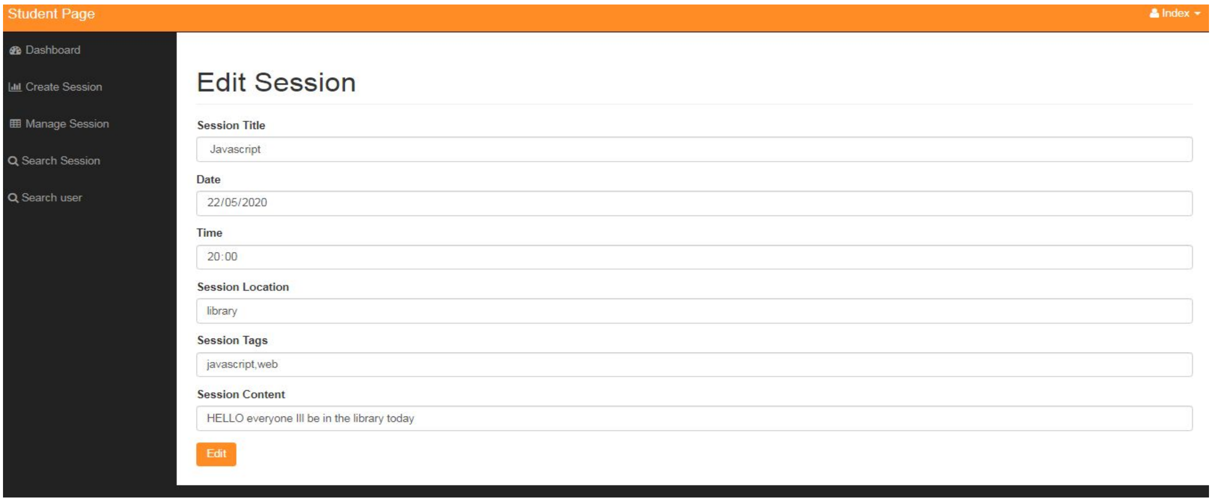
Manage you created!

Id	Course	Session Title	Session author	Session Date	Session time	Session Location	Session Content	Edit	Delete
121	IT	Javascript	SuelenAdmin	2020-05-22	20:00:00	library	HELLO everyone Ill be in the library today	Edit	Delete

Sessions you are In!

Id	Course	Session Title	Session author	Session Date	Session time	Session Location	Session Content	Leave Session
120	IT	Programming Ca2	Gourai	2020-05-20	16:45:00	cct library	We gonna work in the CA2 year 1 program.	Leave Session
121	IT	Javascript	SuelenAdmin	2020-05-22	20:00:00	library	HELLO everyone Ill be in the library today	Leave Session

Figure B-15 My Sessions page – Student Meetup



Edit Session

Session Title: Javascript

Date: 22/05/2020

Time: 20:00

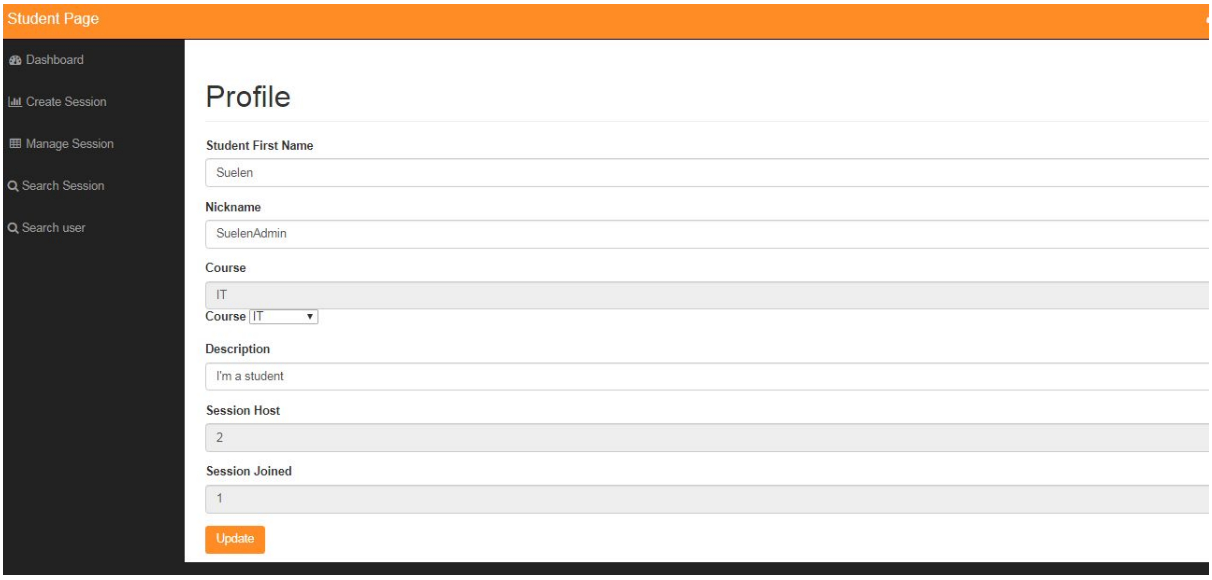
Session Location: library

Session Tags: javascript,web

Session Content: HELLO everyone Ill be in the library today

Figure B-16 Edit Session page – Student Meetup

Users are also able to edit their profile. They can change their name,nickname, course and description.



Profile

Student First Name: Suelen

Nickname: SuelenAdmin

Course: IT

Course: IT ▾

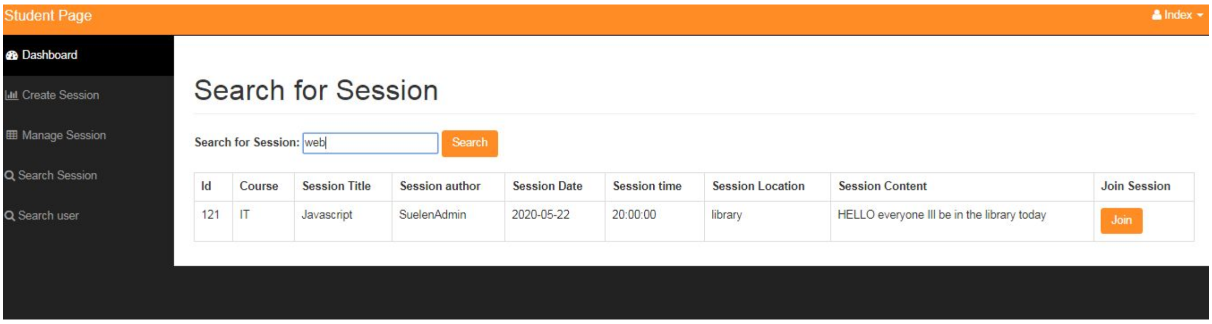
Description: I'm a student

Session Host: 2

Session Joined: 1

Figure B-17 Profile Session page – Student Meetup

In the search session, the user can search for a tag and receive a search result, giving the student the option to join the session.



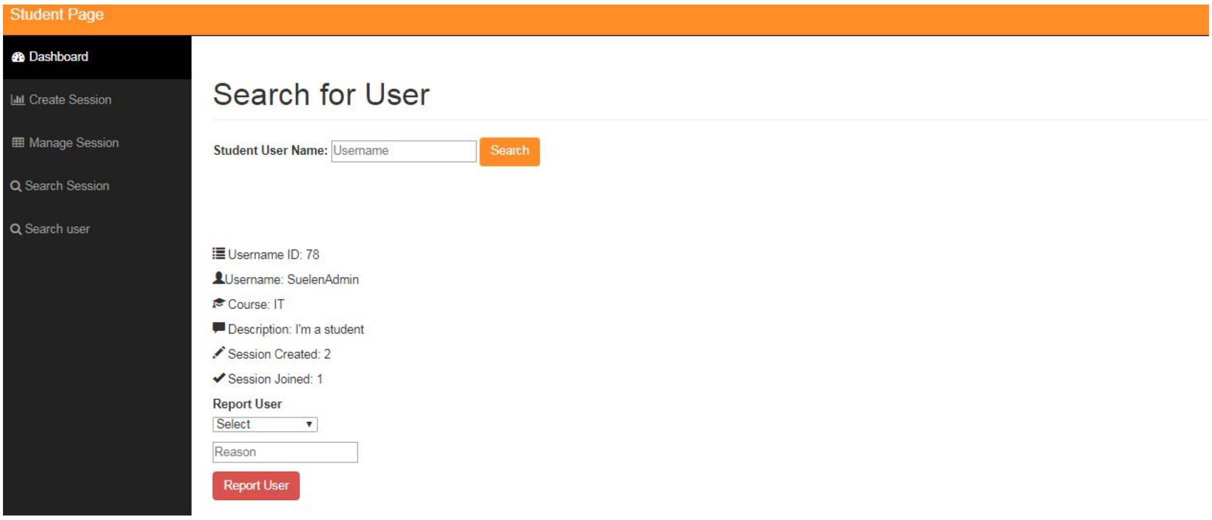
Search for Session

Search for Session: web

Id	Course	Session Title	Session author	Session Date	Session time	Session Location	Session Content	Join Session
121	IT	Javascript	SuelenAdmin	2020-05-22	20:00:00	library	HELLO everyone Ill be in the library today	Join

Figure B-18 Search Session page – Student Meetup

The last feature of the application is for the user to search for another user. When the search is finished it will display the student details. The user can also report another user for abusive behaviour, being a spamming_bot or for other reasons.



Search for User

Student User Name: Username [Search]

Username ID: 78
Username: SuelenAdmin
Course: IT
Description: I'm a student
Session Created: 2
Session Joined: 1

Report User
Select ▾
Reason
[Report User]

Figure B-19 Search User page – Student Meetup

2.5.2 Mobile Application

The mobile application uses Java as its coding language because it is the language we have used during our studies at CCT and we are more comfortable with.

Android studio was used as the IDE for the application, it is based on IntelliJ framework and it is a framework easy to use and intuitive.

Particularly for this project, it was used the LAMP stack for server side hosted in the Google Cloud Platform, but working together with a Node.js server running the library Socket.io in it that is the library we are using for allowing the users of the application to talk to each other. The Node.js server will be hosted in Heroku, which is a web service that allows applications to keep running in the cloud.

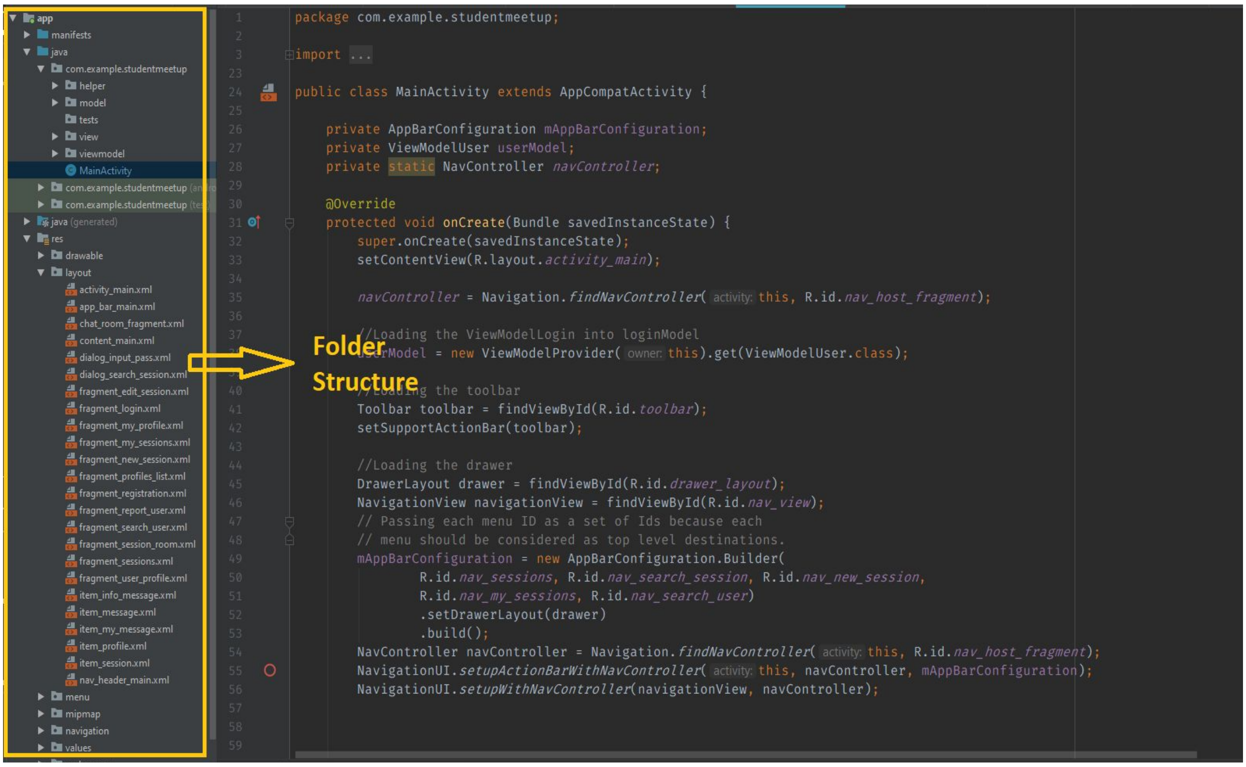
The MVVM design pattern (model, view, view model) was implemented in the application because according to “Android Developers. 2020. *Guide To App Architecture”* the MVVM architecture is nowadays a good practice to implement because it cleanly separates the user interface from the application logic and this improves application maintenance, by encapsulating functionalities.

In particular It has been used recommended technologies by the official Google Android Developer documentation like:

Retrofit, that is a REST client that will parse JSON objects, retrieved from the server, to Java objects, a format manageable by the app.

Shared preferences, this is used for saving user credentials in a private file, allows this particular application to save user login credentials in a secure and private way.

Live data, this new feature will allow create functions that will observe the live data and will be triggered when live data changes, this is very helpful when querying the server and running functions when the response comes back. It also allows the application to keep working as normal even if the expected response is delayed.



```
▼ app
  ▶ manifests
  ▼ java
    ▼ com.example.studentmeetup
      ▶ helper
      ▶ model
        tests
      ▶ view
      ▶ viewmodel
        MainActivity
    ▶ com.example.studentmeetup (an
    ▶ com.example.studentmeetup (te
  ▶ java (generated)
  ▼ res
    ▶ drawable
    ▼ layout
        activity_main.xml
        app_bar_main.xml
        chat_room_fragment.xml
        content_main.xml
        dialog_input_pass.xml
        dialog_search_session.xml
        fragment_edit_session.xml
        fragment_login.xml
        fragment_my_profile.xml
        fragment_my_sessions.xml
        fragment_new_session.xml
        fragment_profiles_list.xml
        fragment_registration.xml
        fragment_report_user.xml
        fragment_search_user.xml
        fragment_session_room.xml
        fragment_sessions.xml
        fragment_user_profile.xml
        item_info_message.xml
        item_message.xml
        item_my_message.xml
        item_profile.xml
        item_session.xml
        nav_header_main.xml
    ▶ menu
    ▶ mipmap
    ▶ navigation
    ▶ values
```

```
package com.example.studentmeetup;

import ...

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration mAppBarConfiguration;
    private ViewModelUser userModel;
    private static NavController navController;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        navController = Navigation.findNavController( activity: this, R.id.nav_host_fragment);

        //Loading the ViewModelLogin into loginModel
        userModel = new ViewModelProvider( owner: this).get(ViewModelUser.class);

        //Loading the toolbar
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        //Loading the drawer
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        NavigationView navigationView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        mAppBarConfiguration = new AppBarConfiguration.Builder(
                R.id.nav_sessions, R.id.nav_search_session, R.id.nav_new_session,
                R.id.nav_my_sessions, R.id.nav_search_user)
                .setDrawerLayout(drawer)
                .build();
        NavController navController = Navigation.findNavController( activity: this, R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController( activity: this, navController, mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);
```

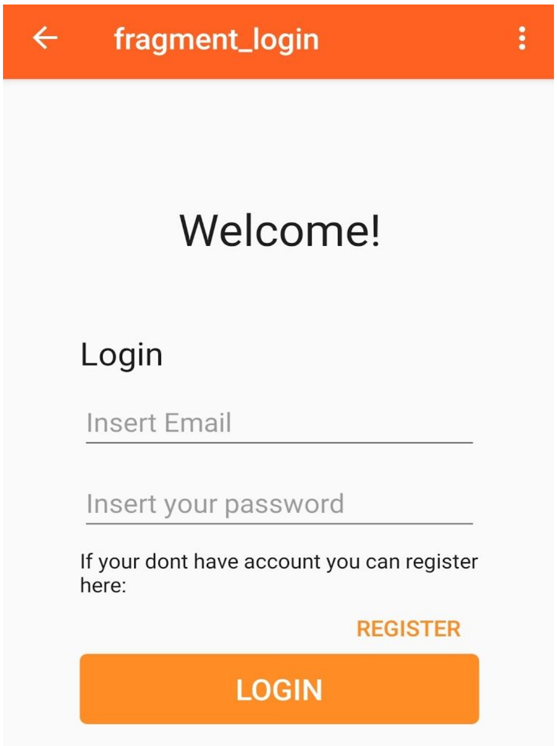
Figure C-1 Folder Structure – Android Studio

This is the folder that is used for the android application Figure C-1 which shows clearly the use of the MVVM model separating logic from the interface. It also used several XML fragment layout files that are the different layouts that the user interacts with. Each fragment layout has its own java class associated with it which permits layouts to have functionalities. Inside the menu folder resides the information related to the menu list options and the drawer list (show sessions, create sessions, etc.).

The mipmap folder stores all the icons this application uses as well as the logo.

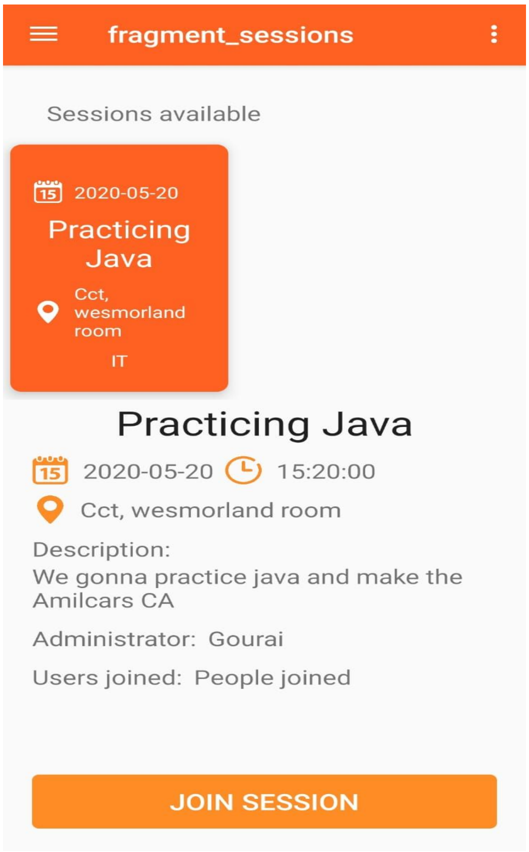
The navigation folder contains the new Android feature called Navigation view where it can be displayed how is the navigation of the different interface layers of the application.

The values folder contains XML information about the colors, strings, styles, dimensions and network security configuration of this application. This particular way of storing the values is very helpful and used, because it allows easy change and generalize values like colors, strings, dimensions, and even will be an important advantage if we decided to make our app available in different languages.



When the application is launched the user is prompted to log in or register. When a user registers they are required to fill in the same information that is required by the web application as the information is stored in the same database, if the email and password does not matches with what there is in the database, the server will send an error message and the app will tell so to the user.

Figure C-2 Login – Android Mobile app



Sessions available

2020-05-20
Practicing Java
Cct, wesmorland room
IT

Practicing Java
2020-05-20 15:20:00
Cct, wesmorland room
Description:
We gonna practice java and make the Amilcars CA
Administrator: Gourai
Users joined: People joined

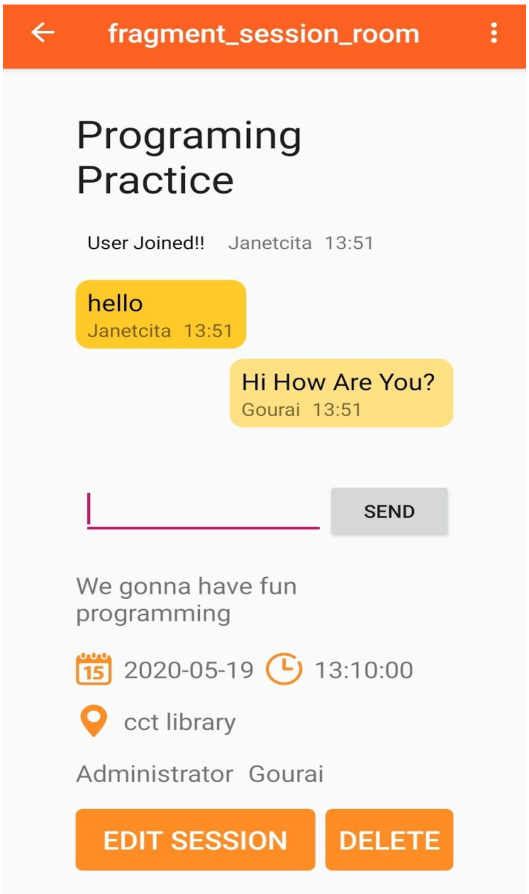
When the server recognizes the user, all its information will be send in form of a JSON object back to the application where Retrofit REST Client is used for retrieving this JSON object and parsing it into a POJO (Plain Old Java Object), next, the server is prompted to send back all the sessions available within the course, IT or Business, and will be stored in the Recycler View that will be in charge of displaying them and managing the memory used to store them.

These sessions will be shown all the currently available sessions. From here they will be able to join any of the visible sessions.

Figure C-3 Sessions – Android Mobile app

When a user selects a session to join they will be shown the session information and given the option to join the chat room.

When the user selects Join Session they will then enter the Session Room where they can better see the session description and have the chance to send and receive messages from members of the session, users from the session connected al chat will acknowledge the new user presence by a information message "User joined", when a user enter the chat room they are announced to the other users in the chat room.



Programing Practice

User Joined!! Janetcita 13:51

hello
Janetcita 13:51

Hi How Are You?
Gourai 13:51

We gonna have fun programming

2020-05-19 13:10:00

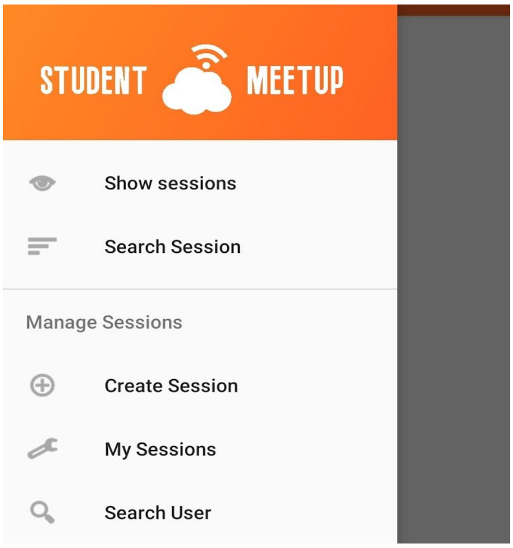
cct library

Administrator Gourai

When inside the session room, the user will be able to write in the chat and any user that is connected to the chat will be able to read and reply. This feature is possible thanks to the Socket.io server that will be running independently and will be only in charge of connecting the users of the same session through the chat. It's good to mention that our socket.io server will be running in another server hosted by Heroku.

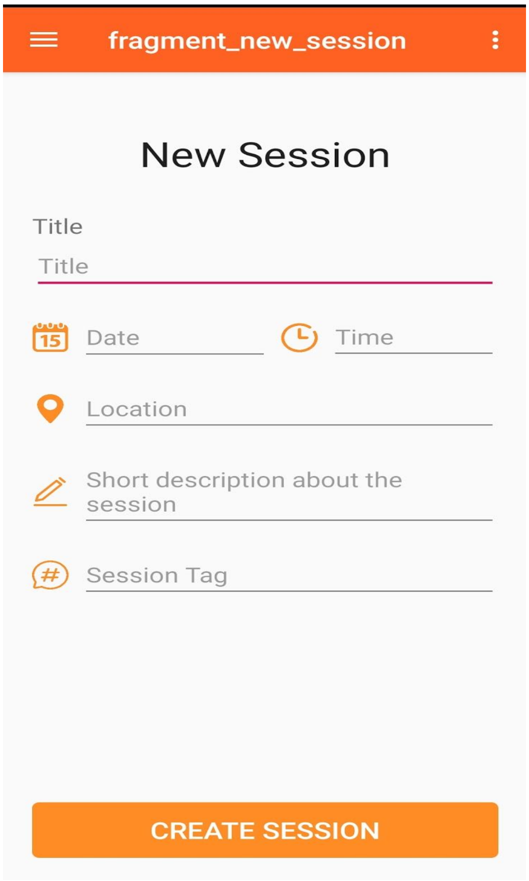
In this particular project the chat room is an independent fragment that will be included inside the parent fragment called Fragment_session_room, this allows reusability of the chat in other fragments.

Figure C-4 Session Room – Android Mobile app



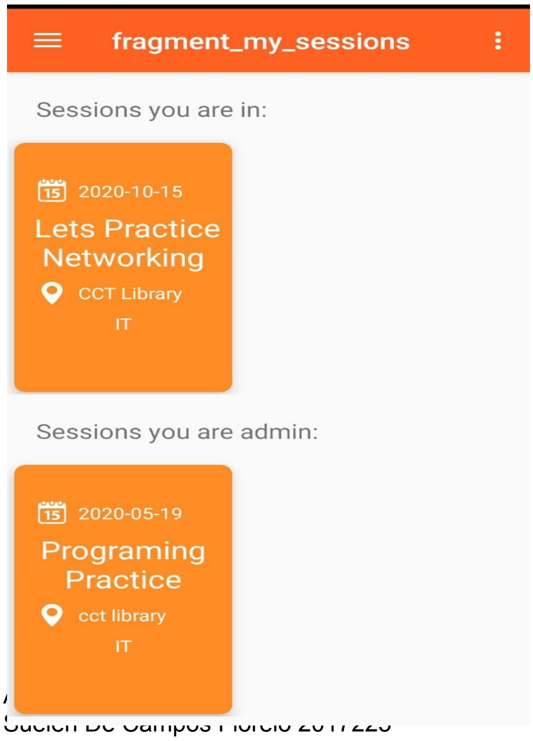
When the user swipes the screen from the right to left a menu is shown with all the available options to a user. These options are the same as the website.

Figure C-5 Drawer menu – Android Mobile app



Users are able to start a new session. They have to select the create session option, this will bring them to the create session screen.the user will be required to input the information about the session. Once the user has filled out the form the user then selects the create session button. This then saves the data to the databases and makes it available to other users to join.

Figure C-6 New Session – Android Mobile app



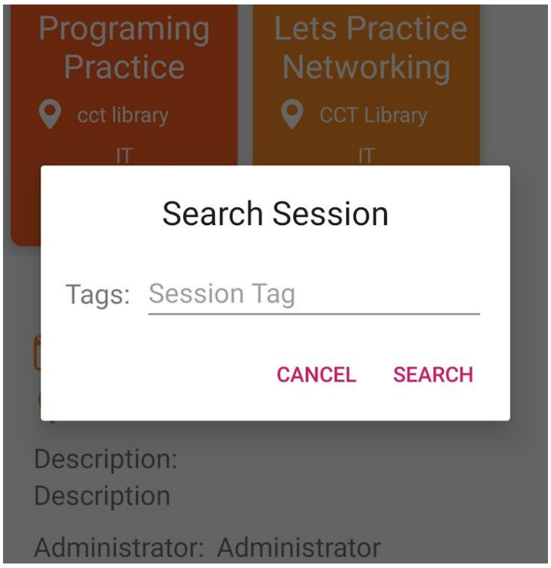
2020-10-15
Lets Practice Networking
CCT Library
IT

2020-05-19
Programing Practice
cct library
IT

Users are able to see the sessions they participate in the “My sessions” option. Here users can see what sessions they have joined and what sessions they have created. If a user wants to edit the information of a session they have created they select the session they want to edit.

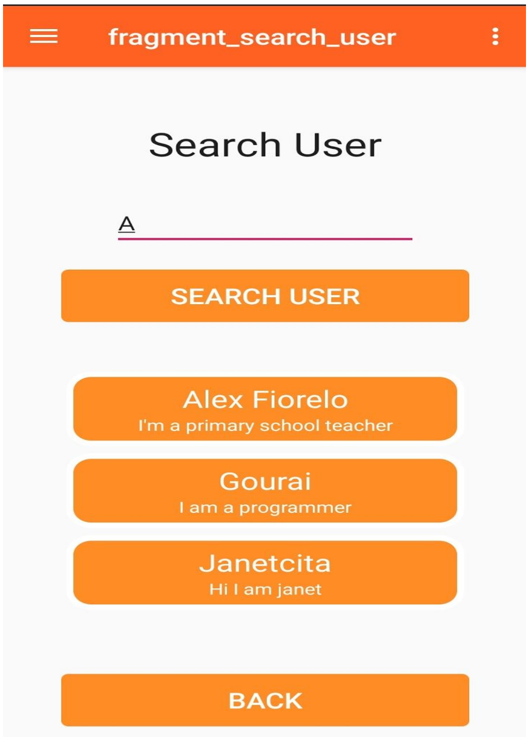
The user will then be able to change any of the information in that session.

Figure C-7 My Sessions – Android Mobile app



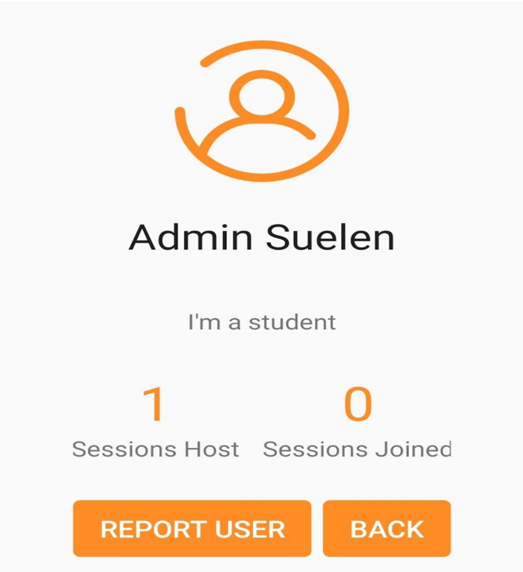
Users are able to search for sessions that they may be interested in by using the tags associated with the created sessions. The search returns any sessions that have tags that have contained the term searched for. What the server does is, grabs the sessions tags provided and searches any session that contains these tags and returns them in JSON format object.

Figure C-8 Search Session – Android Mobile app



If a user wants to find another user, they will be able to type in another user's nickname, this will display any other users with similar nicknames. They are then able to select a user, this is done by the server side that will use the query “LIKE” for retrieving all the users that contain the given String.

Figure C-9 Search User – Android Mobile app



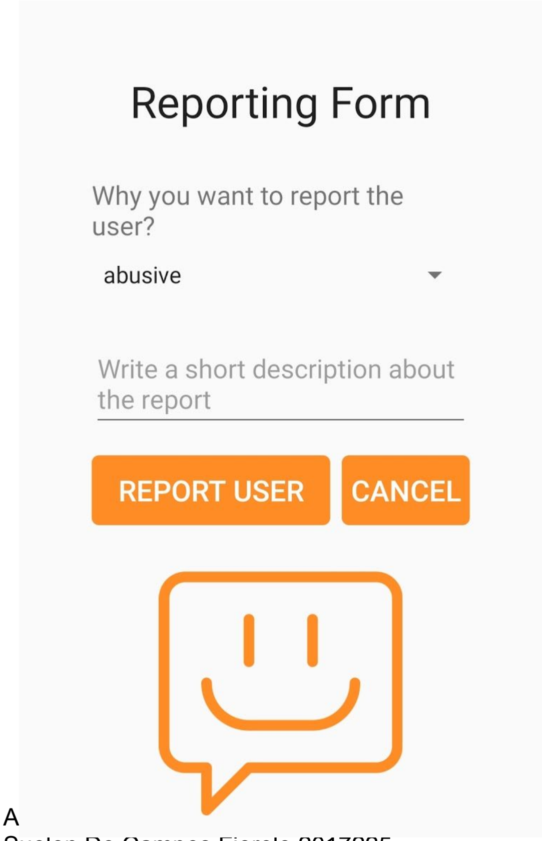
Admin Suelen

I'm a student

1	0
Sessions Host	Sessions Joined

This view is the user profile fragment that will display information about a particular user, It can be seen the nickname, the user description, sessions host and sessions joined, as well as the option to report the user which will open a report form that has to be filled in before reporting the user.

Figure C-10 User Profile – Android Mobile app



Reporting Form

Why you want to report the user?

abusive

Write a short description about the report

This is the reporting form that has to be filled in order to send the report, this report will take a predefined reason: “abusive”, “spamming_bot” or “other” and then a short description about the report also, then by pressing “Report user” the report form will be submitted and saved in the database.

Figure C-11 Reporting Form – Android Mobile app

2.6 Problems encountered

We have faced various challenges at developing the mobile application. First Alejandro, the person in charge of developing the mobile app had to take a 60 hour Udemy course in order to understand and work with Android Studio, which although uses java, he had to handle the many different aspects of a mobile development, like screen rotation, other apps being open when running current app, external notifications that if not correctly handled can cause the app to crash.

Beside the knowledge gathered from doing the course it was paramount to research a lot and overcome challenges like, best architecture to use for this particular app, how to connect to a REST API server, what kind of server to use for connecting both mobile and web application, how to implement the chat feature and enable it to be available and synchronized among website and mobile users, how to correctly deploy our LAMP server to Google cloud, how to make our server to run on HTTPS.

The most challenging part was to find and implement a chat library that could be used for any mobile or web user indistinctly.
After correctly deploying our LAMP server to Google Cloud with help of Bitnami we faced to few issues relating to HTTP traffic, for example, the mobile application was refusing to connect to a server using HTTP, it was enforcing HTTPS for security reasons, after more researching we found out that by default android platforms do not accept HTTP connections, and that this security measure could be fixed in 2 possible ways:

1. By coding an exception in the app that would allow it to connect to the server ip using HTTP.
2. By applying an SSL certificate to our server and making it HTTPS.

We decided to go for the second option, but it took longer than we expected and we were not able to implement HTTPS due to time constraints. Had we been able to implement HTTPS, our application would have been both fully functional and ready to deploy to the market.

2.7 Testing and evaluation

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more functionalities of the system, it helps to identify errors, gaps in code or to assert that the system meets the actual requirements. Guru99.com. 2020

For the mobile application we have used the Blackbox testing methodologies. The blackbox testing is the one where the testing is made simulating the end user, this “end user” will use and “test” the application and will then compare results obtained with results expected.

The whitebox testing is the one where the code and internal flow is tested and analyzed in order to produce the expected flow and results.

2.7.1 Blackbox testing

Systems tasks that the application should perform:

- User registration
- User login
- Create session
- Join session
- Report user
- Chat to session members

2.7.1.1 Set of Inputs and correct outputs

User registration

Input: User fills register form and press register button.

Output: User information was correctly saved into the database.

User login

Input: User writes email and password and press login button.

Output: User is correctly logged in and has been moved to the student dashboard or layout.

Create sesion

Input: User press create session tab, fills the create session form, and after pressed "Create session" is shown the newly created session description.

Output: New session saved correctly in the database and user is been displayed with a new session description.

Join Session

Input: User selects a session he/she wants to join and press "Join session".

Output: User correctly joins the session updating its "session joined" counter and connecting user to the chat app socket.

Report User

Input: User fills out a report form and sends it.

Output: System saves the report form correctly and tells the user so.

Chat to Session members

Input: User is inside the session room and chats to the session.

Output: All the members of the session get the particular message.

2.7.1.2 Test Cases for Blackbox Testing

Four tests were done during this testing, and they were done before and after the server was deployed to the google cloud platform, It is valid to mention that we have not contemplated the problem we could have with HTTP connection to the server. The following test cases were done after the server was deployed to the cloud.

Test case ID	TC01
Test case summary	To verify correct output given an input
Prerequisites	Application installed in an Android Mobile
Test Procedure	User fills the register form and presses the register button.
Expected Result	User information was correctly saved into the database.
Actual Result	Connection could not be made using HTTP, android default settings were blocking insecure connection
Status	Failed
Created by	Alejandro Zeballos
Date of execution	05/05/2020

Test case ID	TC02
Test case summary	To verify correct output given an input
Prerequisites	Application installed in an Android Mobile
Test Procedure	User press the create session tab, fills the create session form, and after pressing “Create session” is shown the newly created session description.

Expected Result	New sessions is saved correctly in the database and the user is displayed with a new session description.
Actual Result	Session was correctly saved into the database and the user was displayed with the result.
Status	Succeed
Created by	Alejandro Zeballos
Date of execution	05/05/2020

Test case ID	TC03
Test case summary	To verify correct output given an input
Prerequisites	Application installed in an Android Mobile
Test Procedure	User is inside the session room and chats to the session.
Expected Result	All the members of the session get the particular message.
Actual Result	Users received the messages but I could not see when a user left or joined the chat room.
Status	Failed
Created by	Alejandro Zeballos
Date of execution	09/05/2020

Test case ID	TC04
Test case summary	To verify correct output given an input
Prerequisites	Application installed in an Android Mobile

Test Procedure	User fills out a report form and sends it.
Expected Result	System saves the report form correctly and tells the user so.
Actual Result	Report was filled in but at the moment of saving, the system kept telling that the report could not be successfully submitted.
Status	Failed
Created by	Alejandro Zeballos
Date of execution	09/05/2020

2.7.1.3 Bug fixing of Black Box test failures

TC01 bug fix

An xml configuration exception was created in order to allow HTTP connection only to the app server.

TC03 bug fix

A bug was related to the type of object I was receiving from the socket.io server when a user joined or left the chat. I was not parsing the correct json object therefore I could not display the info. The bug was fixed when I properly parsed the data I was receiving from the server.

TC04 bug fix

The bug I encountered there was that the enums (type of report: abusive, spamming_bot, other) used in the database were slightly different to the enums I had in the mobile application. The bug was fixed when I changed the app enums to match the ones in the database.

2.8 Conclusions and Further Work

We have created this application because we believe that there is a need for this platform. There are other applications that share some of the functionality of our application, but they also contain functions that are unnecessary and distracting to users intending to study. That is why our group created this application, to take away the option of distracting alternative applications when looking for a group to study with, and to help those who have difficulty finding others to study with by making it easier to pair up with other students.

While we are very happy with the application we have produced so far there are always improvements to be made to any system including ours. During the development of our application a greater amount of testing would have helped us finding and dealing with bugs and usability problems. We should have also spent more time focusing on the documentation for the application, writing it out more frequently and in greater detail.

Some ideas that we were not able to implement into the application, which could be used to improve the functionality of the application include:

- A flagging system that will send notification to alert users of any sessions that match their particular interests.
- User profile pictures, this would allow users to be more comfortable with the groups they may join.
- Integrating google maps into the website, this would allow for users to more easily select the location where they would like to study, and better plan the sessions that they will create.
- Expanding the available courses within our application, making it more friendly to other students who might not study the courses available on the application.
- Updating the chatroom functionality so that users have more options to help them with studying.
- Implement a user storage so that they can save any study note or images that they can look back on after sessions have been completed.
- A buddy system in which users will be able to buddy up and see what people who they have enjoyed studying with can more easily reconnect.
- The current servers run on HTTP but with little more time we could have implemented HTTPS also that will endure the sensitive data traffic.

We have learned an enormous amount and have developed skills in project management, web design, programming, database management, application deployment and teamwork. We have refined our planning, research, development and implementation skills that I believe will assist us with our future careers. We have found the experience rewarding and believe that the students who use this application will benefit from the experience. Given more time to work on the application we would refine the application and implement some of the ideas that have been discussed in this section.

Appendices

Github : Website
https://github.com/DeCamposFiorelo/WebStudentMeetUp

Github: Mobile Application
https://github.com/Alejandro-Zeballos/StudentMeetUp

Contribution report

Student Name: Suelen de Campos Fiorelo

In the first semester in terms of my participation, I took charge of and completed the following areas of the project: Two Use Cases, Entity Relationship Diagram and Timeline of the Project. In addition to solely completing these parts, I also contributed to parts of other areas of the Project which I'll outline in further detail below.

Beginning with Use Cases, I first researched how to put together a use case, using the website usability.gov, in addition to Chapter 2 of the book "A Student Guide to Object Oriented Development" by Carol Britton. I then constructed multiple use cases using Microsoft Word. These involved outlining the steps that the user would need in order to register and also to login.

Regarding the Entity Relationship Diagram, I created the instructions that determined how the database would be built. I also used draw.io to design the diagram. Furthermore I defined the entities, their attributes, the primary keys and the foreign keys.

In terms of the Timeline, I first studied an example from Lecturer Ken Healy. Subsequently, I went to Gantt Charts and began to create our own timeline. This involved planning and deciding what was to be done, who was going to do it and when it would be due in accordance with the Project Guidelines. I then had to input this data into Gantt Charts.

Having worked on the above areas in the first semester , I also contributed to other areas as well. Firstly, I helped put together the introduction along with Alejandro and James. Similarly, I worked with Alejandro to determine the Data Requirements.

In the second semester, I took charge of the construction of the FrontEnd and BackEnd of the Website. I applied the technologies that I searched in the last semester as an example, the Bootstrap, PHP, Javascript, node.js and socket.io.

The Node.js and socket.io it was the most difficult part to implemented in the website, node.js was used in the chat-room,which allowed me to have a real time conversation with the other users. The challenge of the taks was the website and the mobile application to connect together.

I took charge to deploy the chatroom in heroku, which took me a few hours to fix the errors, but in the end the chat was deployed.

I also helped with other tasks, such as the implementation, the test plans, test results, host the database and php files on google cloud and write the final report which we all wroked together.

<u>Student Name: Alejandro Zeballos</u>

During the realization of this project I took part and completed the list of requirements and features of the mobile and web application, I drew the use case diagrams and I formulated three use case scenarios. I designed and drew the system class diagram using Visual Paradigm based on the requirements that we put together previously, for this task I have followed previous examples on “A Student Guide to Object Oriented Development” by Carol Britton.

I have researched what technologies are needed for the development of a mobile and web application and I redacted that part in the project proposal.

Having worked on the areas mentioned. I also contributed with Suelen and James in putting together the introduction.

Finally, for the next semester, I will be in charge of the construction of the system functional design and with the Mobile Application, as I do not have experience doing this, I am actively researching and watching tutorial videos about android development.

I will also be helping in other tasks, as testing, implementation, interface and database design.

After researching and solving problems I have successfully developed the mobile application, php API for the mobile application and I have successfully deployed the LAMP stack to google cloud.

I have participated in the research and adapting of the chat socket.io API.

Student Name: James Harris

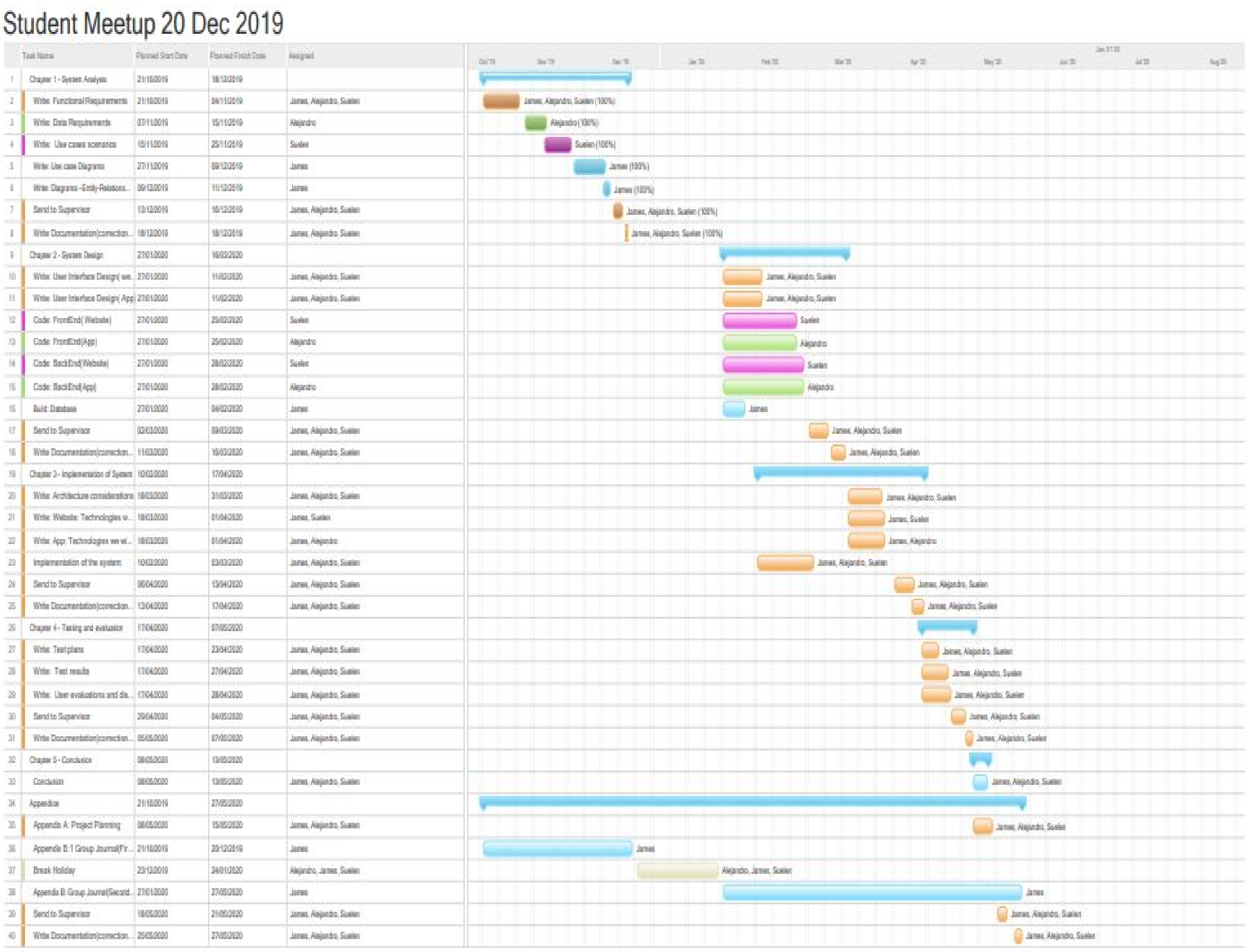
As part of this project I have contributed to the introduction and created two use case scenarios, I also organised a few of our meetings with our supervisor John Snell as well as done research into the effects of group study and if there are other applications similar to ours already available to the public.

For the research on the effectiveness of group study i used google scholar to find articles related to our area of interest, and I also used google search to find out if there are any similar applications to our own.

I used the same source that Suelen De Campos Fiorelo used to construct the use case scenarios. Along with Alejandro Zeballos and Suelen De Campos Fiorelo, I helped to write the introduction to this paper.

For next semester I will be helping to create and maintain the database for our project. I will be learning how to use Google cloud services as well as further research into how to create a robust and stable database that will be able to handle our application. I will also be learning how to use the visual studio IDE with my other team members. I helped with documentation.

Here is the proposed timetable of our projects.



Student Meetup 20 Dec 2019

	Task Name	Planned Start Date	Planned Finish Date	Assigned
1	Chapter 1 - System Analysis	21/10/2019	18/12/2019	
2	Write: Functional Requirements	21/10/2019	04/11/2019	James, Alejandro, Suelen
3	Write: Data Requirements	07/11/2019	15/11/2019	Alejandro
4	Write: Use cases scenarios	15/11/2019	25/11/2019	Suelen
5	Write: Use case Diagrams	27/11/2019	09/12/2019	James
6	Write: Diagrams -Entity-Relations...	09/12/2019	11/12/2019	James
7	Send to Supervisor	13/12/2019	16/12/2019	James, Alejandro, Suelen
8	Write Documentation(correction...	18/12/2019	18/12/2019	James, Alejandro, Suelen
9	Chapter 2 - System Design	27/01/2020	16/03/2020	
10	Write: User Interface Design(we...	27/01/2020	11/02/2020	James, Alejandro, Suelen
11	Write: User Interface Design(App	27/01/2020	11/02/2020	James, Alejandro, Suelen
12	Code: FrontEnd(Website)	27/01/2020	25/02/2020	Suelen
13	Code: FrontEnd(App)	27/01/2020	25/02/2020	Alejandro
14	Code: BackEnd(Website)	27/01/2020	28/02/2020	Suelen
15	Code: BackEnd(App)	27/01/2020	28/02/2020	Alejandro
16	Build: Database	27/01/2020	04/02/2020	James
17	Send to Supervisor	02/03/2020	09/03/2020	James, Alejandro, Suelen
18	Write Documentation(correction...	11/03/2020	16/03/2020	James, Alejandro, Suelen
19	Chapter 3 - Implementation of System	10/03/2020	17/04/2020	
20	Write: Architecture considerations	18/03/2020	31/03/2020	James, Alejandro, Suelen
21	Write: Website: Technologies w...	18/03/2020	01/04/2020	James, Suelen
22	Write: App: Technologies we wi...	18/03/2020	01/04/2020	James, Alejandro
23	Implementation of the system	10/03/2020	03/03/2020	James, Alejandro, Suelen
24	Send to Supervisor	06/04/2020	13/04/2020	James, Alejandro, Suelen
25	Write Documentation(correction...	13/04/2020	17/04/2020	James, Alejandro, Suelen
26	Chapter 4 - Testing and evaluation	17/04/2020	07/05/2020	
27	Write: Test plans	17/04/2020	23/04/2020	James, Alejandro, Suelen
28	Write: Test results	17/04/2020	27/04/2020	James, Alejandro, Suelen
29	Write: User evaluations and dis...	17/04/2020	28/04/2020	James, Alejandro, Suelen
30	Send to Supervisor	29/04/2020	04/05/2020	James, Alejandro, Suelen
31	Write Documentation(correction...	05/05/2020	07/05/2020	James, Alejandro, Suelen
32	Chapter 5 - Conclusion	08/05/2020	13/05/2020	
33	Conclusion	08/05/2020	13/05/2020	James, Alejandro, Suelen
34	Appendice	21/10/2019	27/05/2020	
35	Appendix A: Project Planning	08/05/2020	15/05/2020	James, Alejandro, Suelen
36	Appendix B:1 Group Journal(Fir...	21/10/2019	20/12/2019	James
37	Break Holiday	23/12/2019	24/01/2020	Alejandro, James, Suelen
38	Appendix B: Group Journal(Second...	27/01/2020	27/05/2020	James
39	Send to Supervisor	18/05/2020	21/05/2020	James, Alejandro, Suelen
40	Write Documentation(correction...	25/05/2020	27/05/2020	James, Alejandro, Suelen

James, Alejandro, Suelen (100%)
Alejandro (100%)
Suelen (100%)
James (100%)
James (100%)
James, Alejandro, Suelen (100%)
James, Alejandro, Suelen (100%)

Figure D-1 Timeline –Gantt Charts

List of References

Schroenherr, N. (2006). *Discovering why study groups are more effective | The Source | Washington University in St. Louis*. [online] The Source. Available at: https://source.wustl.edu/2006/07/discovering-why-study-groups-are-more-effective/ [Accessed 13 Oct. 2019].

Weimer, M. (2018). *The Benefits of Study Groups*. [online] The Teaching Professor. Available at: https://www.teachingprofessor.com/for-those-who-teach/what-students-can-learn-from-studying-together/ [Accessed 13 Oct. 2019].

Top Universities. (2013). *Joining a Study Group: The Benefits*. [online] Available at: https://www.topuniversities.com/blog/joining-study-group-benefits [Accessed 13 Oct. 2019].

Weimer, M. (2019). *The Elusive Benefits of Study Groups*. [online] Faculty Focus | Higher Ed Teaching & Learning. Available at: https://www.facultyfocus.com/articles/teaching-and-learning/the-elusive-benefits-of-study-groups/ [Accessed 13 Oct. 2019].

McCabe, J. A., & Lummis, S. N. (2018). Why and how do undergraduates study in groups? *Scholarship of Teaching and Learning in Psychology, 4*(1), 27–42. https://doi.org/10.1037/stl0000099 [Accessed 13 Oct. 2019].

Affairs, A. (2019). *Use Cases* | *Usability.gov*. [online] Usability.gov. Available at: https://www.usability.gov/how-to-and-tools/methods/use-cases.html [Accessed 25 Nov. 2019].

Britton, C (2005). *A Student's Guide to Object Oriented Development.* Elsevier Butterworth-Heinnemann,

Kendall, K. and Kendall, J. (1998). *Systems analysis and design*. 4th ed. Prentice hall, p.9.

Ababei, A. (2019). *8 Reasons to Use Kotlin Over Java for Android Development - DZone Java*. [online] dzone.com. Available at: https://dzone.com/articles/what-are-the-biggest-advantages-of-kotlin-over-jav [Accessed 20 Dec. 2019].

Android Developers. 2020. *Guide To App Architecture* | *Android Developers*. [online] Available at: <https://developer.android.com/jetpack/docs/guide?gclid=Cj0KCQjwncT1BRDhARIsAOQF9LnyDgddcgNPMdHBoEw4zxweWg8oLVhzn4TnffwZUH5TgNfPaoIG6pQaAmMrEALw_wcB&gclsrc=aw.ds> [Accessed 5 March 2020].

Android Developers. 2020. *Viewmodel Overview* | *Android Developers*. [online] Available at: <https://developer.android.com/topic/libraries/architecture/viewmodel?gclid=Cj0KCQjwm9D0BRCMARIsAIfvfIZUi4dA-tv21IY7_wbLmGFZjypxadHF70ww0Ji0D-MvYZz532ybK0QaAskXEALw_wcB&gclsrc=aw.ds> [Accessed 17 February 2020].

Square.github.io. 2020. *Retrofit*. [online] Available at: <https://square.github.io/retrofit/> [Accessed 24 February 2020].

Medium. 2020. *Android Architecture Patterns Part 3:Model-View-Viewmodel*. [online] Available at: <https://medium.com/upday-devs/android-architecture-patterns-part-3-model-view-viewmodel-e7eeee76b73b> [Accessed 17 February 2020].

Douevencode.com. 2020. *Android P - CLEARTEXT Communication Not Permitted By Network Security Policy*. [online] Available at: <http://www.douevencode.com/articles/2018-07/cleartext-communication-not-permitted/> [Accessed 5 April 2020].

Hosting, A., 2020. *How To Configure The Windows Firewall To Allow Only Specific IP Address To Connect Your Ports?*. [online] AccuWeb Hosting. Available at: <https://manage.accuwebhosting.com/knowledgebase/2984/How-to-configure-the-Windows-Firewall-to-allow-only-specific-IP-Address-to-connect-your-ports.html> [Accessed 15 April 2020].

[Jetpack], H., Denysenko, A. and Valdes, P., 2020. *How To Change Start Destination Of A Navigation Graph Programmatically [Jetpack]*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/51173002/how-to-change-start-destination-of-a-navigation-graph-programmatically-jetpack> [Accessed 20 March 2020].

Tutorialrepublic.com. 2020. *How Append A String In PHP*. [online] Available at: <https://www.tutorialrepublic.com/faq/how-to-append-a-string-in-php.php> [Accessed 4 January 2020].

[duplicate], C., Stander, A., Wheat, M. and J., E., 2020. *Combining "LIKE" And "IN" For SQL Server*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/1865353/combining-like-and-in-for-sql-server> [Accessed 19 February 2020].

Simpson, R., 2020. *Five Ways Colour Can Affect Children In A Learning Environment* |. [online] Lsneducation.org.uk. Available at: <http://www.lsneducation.org.uk/five-ways-colour-can-affect-children-learning-environment/> [Accessed 25 April 2020].

Medium. 2020. *Mobile App Design: 14 Trendy Color Schemes*. [online] Available at: <https://medium.com/@Adoriasoft/mobile-app-design-14-trendy-color-schemes-2669b5bb77d3> [Accessed 5 May 2020].

https://www.udemy.com. 2020. *PHP For Beginners*. [online] Available at: <https://www.udemy.com/course/php-for-complete-beginners-includes-msql-object-oriented/learn/lecture/2380310?start=0#overview> [Accessed 12 November 2019].

Udemy. 2020. *Curso Web Moderno Completo Com Javascript 2020 + Projetos*. [online] Available at: <https://www.udemy.com/course/curso-web/> [Accessed 15 November 2019].

Udemy. 2020. *The Complete Node.Js Developer Course (3Rd Edition)*. [online] Available at: <https://www.udemy.com/course/the-complete-nodejs-developer-course-2/?start=60> [Accessed 20 January 2020].

Mark Otto, a., 2020. *Bootstrap*. [online] Getbootstrap.com. Available at: <https://getbootstrap.com/> [Accessed 12 November 2019].

W3schools.com. 2020. *W3schools Online Web Tutorials*. [online] Available at: <https://www.w3schools.com/> [Accessed 7 November 2019].

En.wikipedia.org. 2020. *Google Cloud Platform*. [online] Available at: <https://en.wikipedia.org/wiki/Google_Cloud_Platform> [Accessed 5 January 2020].

En.wikipedia.org. 2020. *XAMPP*. [online] Available at: <https://en.wikipedia.org/wiki/XAMPP> [Accessed 10 February 2020].

apachefriends.org. 2020. *Host Your Application In The Google Cloud With XAMPP And Bitnami*. [online] Available at: <https://www.apachefriends.org/docs/hosting-xampp-on-google.html> [Accessed 17May 2020].

Guru99.com. 2020. *What Is Software Testing? Introduction, Definition, Basics & Types*. [online] Available at: <https://www.guru99.com/software-testing-introduction-importance.html> [Accessed 17 May 2020].

Hammar Chiriac, E., 2014. *Group Work As An Incentive For Learning Â€“ Studentsâ€™ Experiences Of Group Work*.