



uc3m | Universidad
Carlos III
de Madrid

CUPCAKE

Pablo Yuste	100406674
Gonzalo Tomás	100406615
Javier Icaza	100406598
Ricardo Estaire	100406711

1. INTRODUCTION

Mate Buddy Trips is a mobile application designed to make easy to find mates to travel with. This idea came up because this is a problem nobody has ever thought about or gave an answer to.

In order to be able to use Mate Buddy Trips, you need to create an account using your mail direction and establishing a password and a username.

Once you have an account in the application you will be connected to all the other users that are looking for someone to go on an adventure with. The app navigation is going to work with tabs. We are going to have 3 different tabs.

PROFILE:

In this tab the user is going to be able to see how his profile looks for other users, he is going to be able to enter information about him, log out of the app and delete his account if he wants to. He will also be able to see his favourite users and the users that have them between their favourites. (Followers/Following)

SEARCH:

In this tab the user is going to be able to discover other users in the app using some filters in which our user will choose which parameters his perfect travel mate should follow. Through this section our user can access other users profiles and establish their favourite users.

POST:

This tab will work in a similar way as the search tab, but instead of users, post will be displayed. Each post is published by a user and each user can choose his favourite posts also. This post will be created, edited and deleted through a layout which the user will have access to through this tab.

In order for the users to interact between them, we will have a contact info part in each user's profile.

Our idea of the application is that it can be used by everybody no matter if they travel alone, with a partner or with the whole family as long as they wish to meet people and travel with them, they are welcome into Mate Buddy Trips.

For this part of the project, our main concerns have been to create the first layouts of the application, in which you can create an account and log in to the application. Connecting our app to a database was the hardest part of this project. But finally we were able to keep all users in a database only allowing those user in our database to access our app.

We will explain how we connected our app to an SQL database later on and what were the steps we followed in order to accomplish this part of our project.

2. LAYOUTS ELEMENTS

REGISTER AND LOG IN:

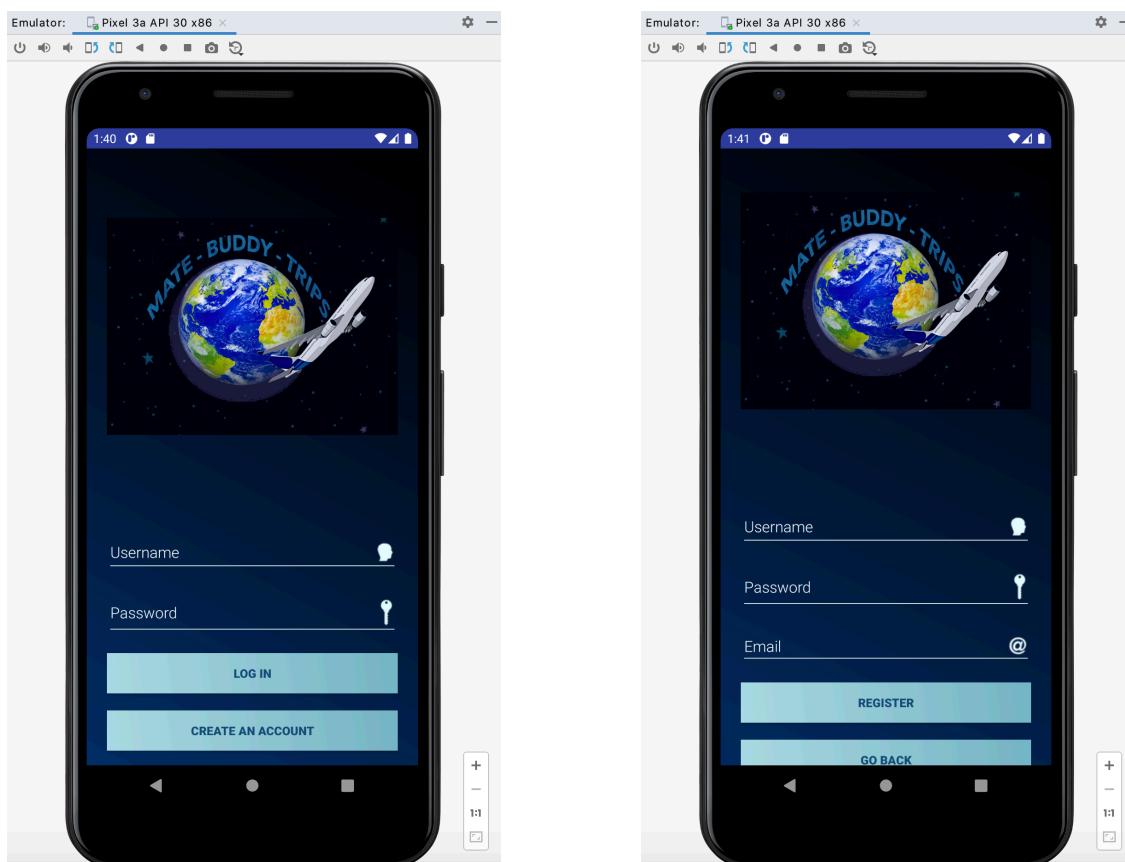
The Register and Log In layouts are created in the same .xml file, but depending on your actions some elements are visible and others are gone.

It is composed of the brand image, username and password requests and two buttons, the first to log in and the second to go to the register layout. If you click the 'Create an Account' button, the application shows you the 'Register' layout. This layout is formed also by the brand image, username, password and email requests and two buttons, the first one to register the account and the second one to go back to the log in activity.

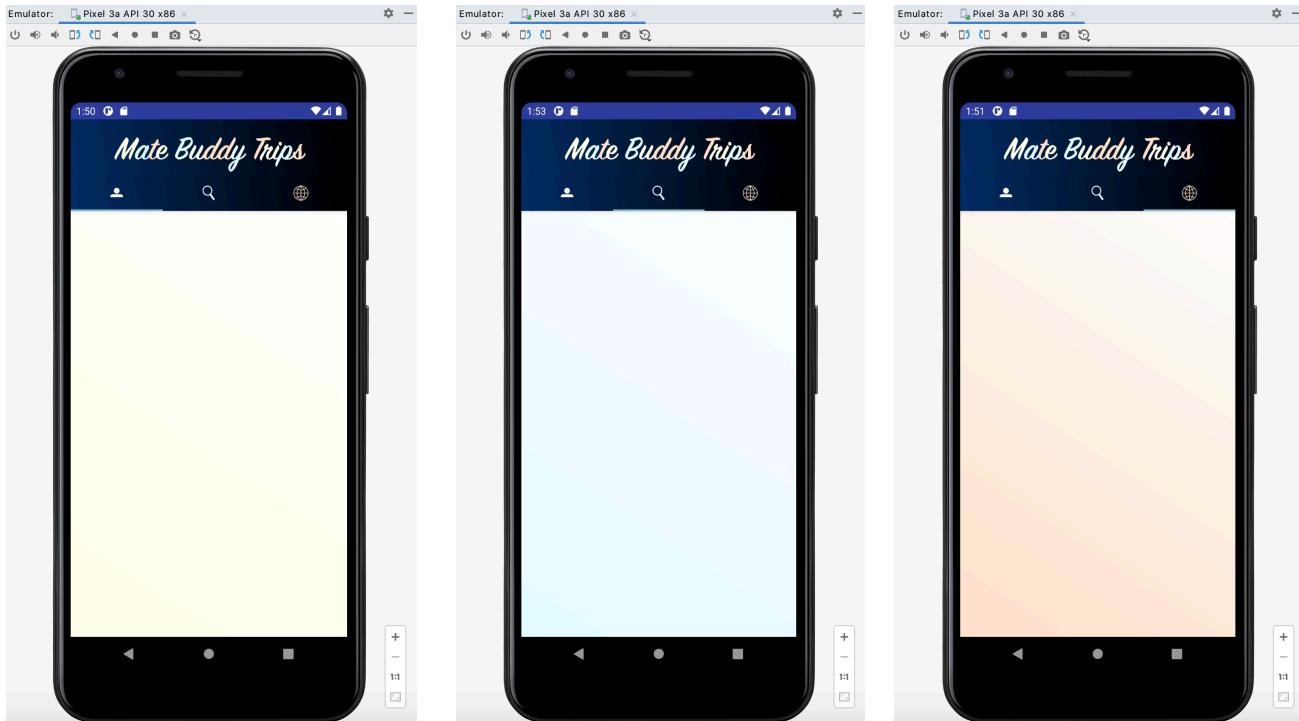
It is a layout with 3 EditTexts and 3 buttons, but depending on your actions some of them are visible and some of them are gone.

We have also implemented some simple control over the inputs of the users before they are passed into the database like the inputs not being empty, the password must have 6 or more characters, you must enter a valid email. Our program also checks if the username or email already exist in our database, and if they do, a message is displayed.

Once all the inputs are valid the new account is registered and added to our database. Then the app takes you automatically to the Log In layout.

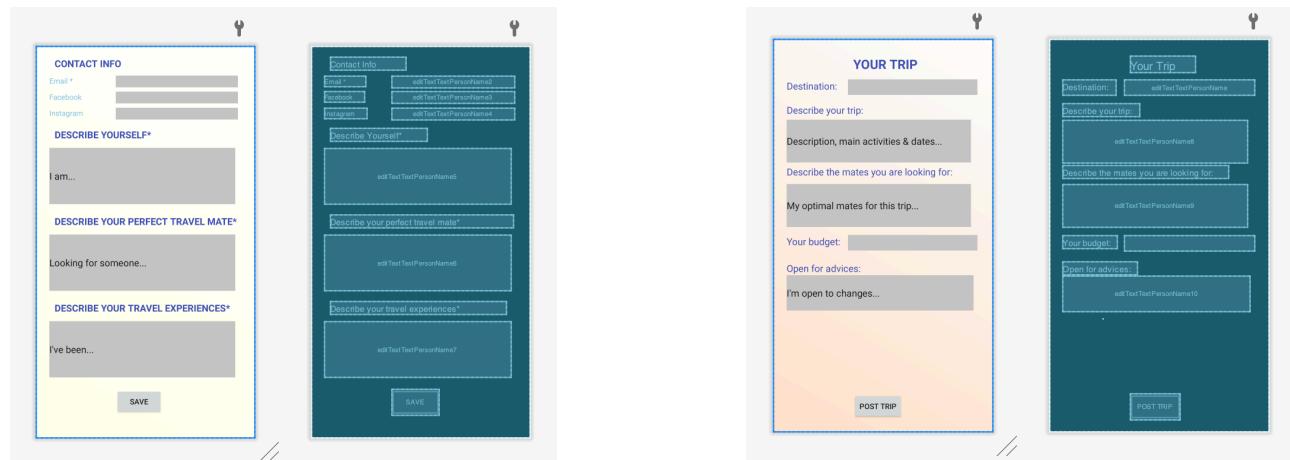


If you enter a valid username and password that corresponds to an account in our database, then, the app allows you to access the main activity of the app. Which as I said before, it works with 3 simple tabs.



Mate Buddy Trips application is characterised for being an extremely simple one but also remarkably functional. Basically, the application consists in just a menu in which users can select the action they want to do between just three. Once the action is selected, the application redirects the user to a layout designed to perform each activity.

We have tried many different layouts for this 3 main screens, but since all of them highly depend on being connected to the SQL database and we have had several problems to implement this, we have decided to not include them in our final code.



3. DIFFICULTIES

Starting to implement our application we faced with some obstacles that made our life as android studio programmers not so easy. The first and main difficulty we had was trying to connect our application to a database to store the specifics of the registered accounts so once an account is created it is possible to log in into the inside of the application. We solved this issue by establishing a connection between the database and the application using the program XAMPP.

CONNECTING THE DATABASE:

After downloading XAMPP, we open and run the application with the Apache and MySQL servers. Through the application we access phpMyAdmin where we are going to create our database ("firstDB"). The first table that we have created is the users table. The insert of new rows to this table is done only through the Register layout.

In order to connect the database we have implemented the JSONParser java document and through the following IP address ("http://192.168.64.2/xampp_php/index.php") our app was able to access our database.

In order to create the connection and manipulate how the data is added and accessed in the database we are using different .php files that we modify using Visual Studio. All this code is inside the xampp_php folder, which must be added inside the htdocs folders of XAMPP in order for the connection to work. We will upload this folder with the rest of the Android Studio code.

This is how the users table looks right now:

Showing rows 0 - 23 (24 total, Query took 0.0006 seconds.)						
<input type="checkbox"/> SELECT * FROM `users` <input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]						
		Show all	Number of rows:	25	Filter rows:	Search this table
+ Options	← →				Sort by key:	None
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	22	Coladilla16	8de7179312d4b998eee4ec62666df28b polyuste@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	23	Gon007	3cbe21ffd9d0a6f90782ad7ac2900bc gon07@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	24	Peche	61b069d4ae632316b7cd3d95c513ee81 peche@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	25	icaza	ad2dd673ee6eab69c20be0ee957f3c5b icaza@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	26	willy	d055503a787726eab8e983ca566b0ca7 willy@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	27	alberto	177daec14b34103960ec27ba29bd686b alberto@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	28	username	14c4b06b824ec593239362517f538b29 username@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	29	username1	14c4b06b824ec593239362517f538b29 username1@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	30	username7	14c4b06b824ec593239362517f538b29 username7@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	31	username8	14c4b06b824ec593239362517f538b29 username8@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	32	username2	14c4b06b824ec593239362517f538b29 username2@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	33	username3	14c4b06b824ec593239362517f538b29 username3@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	34	username5	14c4b06b824ec593239362517f538b29 username5@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	35	username6	fc00e41a55f8ec467328bae44ebd6616 username6@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	36	username4	14c4b06b824ec593239362517f538b29 username4@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	37	user1	14c4b06b824ec593239362517f538b29 user1@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	38	user2	14c4b06b824ec593239362517f538b29 user2@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	39	user3	14c4b06b824ec593239362517f538b29 user3@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	40	user16	14c4b06b824ec593239362517f538b29 user16@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	41	user0	14c4b06b824ec593239362517f538b29 user0@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	42	user00	14c4b06b824ec593239362517f538b29 user00@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	43	user000	20bd845cd4ab3fb5826c88032f46ff user000@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	44	gonzza	974587b9115abe9bdb1e70fe7f61094d gonzza@gmail.com
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	45	Alicia	e94af563867e9cdf3fc999bdc045f5 alicia@gmail.com

The screenshot shows the MySQL Workbench interface for managing a database table named 'User'. At the top, there are tabs for 'Table structure' and 'Relation view'. Below is a table with columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
3	password	text	utf8mb4_general_ci		No	None			Change Drop More
4	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
5	created_at	text	utf8mb4_general_ci		No	None			Change Drop More

Below the table are several buttons: Check all, With selected: Browse, Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Add to central columns, Remove from central columns, Print, Propose table structure, Track table, Move columns, Normalize, Add, column(s), after created_at, and Go.

At the bottom, there is a section titled 'Indexes' with a table:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit	PRIMARY	BTREE	Yes	No	id	24	A	No	
Edit	username	BTREE	Yes	No	username	24	A	No	
Edit	email	BTREE	Yes	No	email	24	A	No	

Another problem we faced was that when we started to built the application we were not able to redirect correctly the user to the desired activity when pressing a button in the menu. To solve this issue we placed hyperlinks that leads the user to the activity of the button they have pressed.

Looking ahead to finish the app we think there are some tasks we are going to struggle with. Firstly, the application needs to store all the profile information each user provides in a database to latter show it to other users. Just as we need to store the profile information, we need to store all the posts the users upload to find travel mates. Another difficulty we are going to face is when implementing the search activity because we are going to have to specify how the search has to be made. Users could search for a specific place they want to go, a type of trip they would like to do or a particular activity, so this singular issue will not be easy to solve.