Project Report

Mobile and Ubiquitous Computing - 2021/22

Course: METI

Campus: Tagus

Group: 4

Name: João Pedro Lopes	Number: 93588	E-mail: joaopedrolopes00@tencico.ulisboa.pt
Name: Francisco Bento	Number: 93581	E-mail: francisco.bento@tencico.ulisboa.pt
Name: António Salgueiro	Number: 93575	E-mail: antonio.bastos.salgueiro@tecnico.ulisboa.pt

1. Features

Component	Feature	Fully / Partially / Not implemented?
Mandatory Features	Create Unique Username	Fully implemented
	Create/Search/Join/Leave Public Chatrooms	Fully with a bug mentioned in bug section.
	Create/Share/Join/Leave Private Chatrooms	Fully implemented
	Create/Search/Join/Leave Geo-fenced Chatrooms	Fully implemented
	Show/Hide Geo-fenced Chatrooms When Moving	Not implemented
	Messages Sync Across Devices Promptly	Fully implemented
	History Available to All Users in Chatroom	Fully implemented
	Posting Text Messages	Fully implemented
	Posting Photos from Camera	Fully implemented
	Posting Locations	Not implemented
	Messages Indicate Author and Timestamp	Fully implemented
	Chatrooms Indicate Unread Messages	Not implemented
	New Message Notifications	Not implemented
	Efficient Message Retrieval	Fully implemented
	Download Images on Request with Cellular Data / Automatically with WiFi	Not implemented
	Data Caching	Not implemented
	Cache Pre-loading	Not implemented
Securing Communications	Encrypt Data in Transit	Not implement
	Check Trust in Server	Not implemented
Meta Moderation	Message Flagging	Not implemented
	Filtering Flagged Messages	Not implemented
	User Blocking	Not implemented
User Accounts	Account Creation	Fully implemented
	Login / Logout	Fully implemented
	Account Data Synchronization	Fully implemented
	Guest Access	Fully implemented
	Private Chatroom ACL	Fully implemented
Additional Media: Files	Pick File and Upload	Not implemented
	Download File and Open	Not implemented
Additional Media: Polls	Create Poll	Not implemented
	Vote / Change Vote	Not implemented
	Show Current Tally with Bar Plot	Not implemented
Social Sharing To Other Apps	Sharing Items	Partially implemented, only done for shared links.
Social Sharing From Other Apps	Accepting Shared Items	Not implemented
	Posting Shared Item to Chatroom	Not implemented
Localization	Translate UI	Not implemented
	Translate Chats	Not implemented
UI Adaptability: Rotation	UI Works Well Vertically and Horizontally	Fully implemented
UI Adaptability: Light/Dark Theme	UI Works Well in Light and Dark Mode	Fully implemented
Recommendations	Compute Most Likely Chatroom Pairings	Not implemented
	List Sorted Suggestions	Not implemented

2. Grading Adjustments

All group elements contributed with the same amount of network, since most of the work was mainly done through pair-programming, therefore no adjustment should be required.

Student #	Adjustment
93581	0
93588	0
93575	0

3. Mobile Interface Design

The wireframe design of our application, which addresses every possible combination of user actions, will be included as a separate appendix file. It contains the activity logic of our program.

4. Server Architecture

The server was designed using the Spring Boot framework, to create a fully production-ready environment.

4.1 Data Structures Maintained by Server and Client

Server: The server has 3 types of primary data structures: Chatrooms, Users and Messages.

Client: The client has a total of 4 activities: MainActivity, which represents the main screen with all listed chats, Login Activity which deals with logging in users and guests, CreateNewUserActivity, which deals with signing up new users as users with password or guests and at last, ChatMessageActivity, which represents the state of being inside a chatroom.

4.2 Description of Client-Server Protocols

To establish connction between the client and the server, two protocols are used:

- HTTP
- WebSockets

Web Sockets are only used to load information into ChatMessageActivity. However, we have created one socket for each Chatroom created, which comes to a higher cost of energy when compared with one for all Chatrooms. Our lack of familiarity with the subject led to this absence of thought.

HTTP is used to establish any other connections, such as: Login (as guest or not), Create New Account, Load the Chat Message List, etc.

4.3 Other Relevant Design Features

Although it wasn't requested, we chose to include light and dark mode buttons to provide users the option of temporarily overwrite the OS's theme (for that execution runtime). The theme will be synchronized with the OS theme after the app has been closed and opened once again.

5. Implementation

Regarding implementation choices, the server was implemented using SpringBoot and JPA (all in JAVA programming language). For the client-side, the program is mainly built over activities. All main components of the app are activities, such as login and create new account screen, the screen where all chats are listed and the screen of each chat (containing the messages associated with that single chat) are also activities.

All other features inside an activity are fragments. These features can be found on the action bar, at the top, and they can only be found in MainActivity and ChatMessageActivity.

The global state of the app is stored at the files HttpCalls.java and all of it is shared between all activities and fragments via its property of being an Parcelable class. If it is not shared via the Parcelable property, it will be directly invoked from the HttpCalls file, calling the constructor of any other class.

All communications are performed either using HTTP or Web Sockets, the state of this request it is not maintained at the mobile device except for the login. Login credentials are store at the shared preferences of the mobile device. Regarding the server side all state is maintained in a database as long as the server is not turned off. Once the server is turned off, all database is erased.

6. Bugs

Search chat fragment filter option only works when the fragment is opened for the second time. To test if the search input is filtering the list of new public chatrooms, it is recommended to do two quick clicks at the search button on the action bar of the MainActivity (mainscreen) after typing key letters.

7. Conclusions

The project itself is well-organized and well-documented, leaving little room for uncertainty.

From a technological standpoint, the project theme is very interesting because we can program an application similar to those used on a daily basis and tackle different topics separately.

On our end, we admit that we had a late and sluggish start. We believe it was due to a lack of adaptability in the learning curve between what we could achieve in lab sessions and what was required (a small adjustment between the last "theoretical programming class" and the project scope), but also to the extensive amount of work that came from other courses that the members were taking.

We believe that the trimester scheme is to blame for the two latter issues mentioned above, as it forces teachers to cut down their practical class goals in an ineffective manner and makes the MEPP experience worse than it should be for students to learn properly.