Project Report

Ubiquitous and Mobile Computing - 2017/18

Course: MEIC

Campus: Tagus

Group: 13

Name: Afonso Caetano   Number: 82539   E-mail: afonso.caetano@tecnico.ulisboa.pt

Name: Bruno Santos   Number: 82053   E-mail: bruno.o.santos@tecnico.ulisboa.pt

Name: Pedro Lopes   Number: 81988   E-mail: pedro.c.lopes@tecnico.ulisboa.pt

1. Achievements

Version	**Feature**	**Fully / Partially / Not implemented?**
Baseline	Sign up	Fully
	Log in / out	Fully
	List tour locations	Fully
	Download Quiz Questions	Fully
	Answer Quiz Questions	Fully
	Post Quiz Answers	Partially
	Read Quiz Results	Fully
	Progress Sharing	Fully
Advanced	Timed Quizzes	Fully
	Security	Fully

2. Mobile Interface Design

The program architecture is composed by two major components, the client mobile application (HopOnCMU) and the server. The server provides the user with tourism quizzes that can be answered on the mobile app by the client, then the client can check the overall ranking of all the users of the application that have participated by answering at least one quiz. As the tourist answer the question, they can share the progress (general balance between right and wrong answers) while travelling between monuments. The process repeats itself until no more monuments remain to visit. The client can always revisit an answered quiz

An overall wireframe diagram of the program can be consulted in the "Attachements" section.

3. Baseline Architecture

3.1 Data Structures Maintained by Server and Client

Server:

- ArrayList<User> users;
- ArrayList<Quiz> quizzes;
- ArrayList<String> tickets;
- HashMap<String, SessionID> sessionIDs.

Client:

- Database that saves the answers of each user to each quiz preformed. SQLite was used for this implementation.

3.2 Description of Client-Server Protocols

The client server protocol implemented was based on the sample Server example gave by the teachers, where both client and server have a Command and Response interfaces, to send and receive requests respectively.

3.3 Description of P2P Protocol for Decentralized Message Delivery

Each time a client responds to a question, he sends his answer to all other clients currently answering a quiz that are in the same network group, such that the second ones receive a message with the answer of the associated client.

3.4 Other Relevant Design Features

The log in architecture in our program only request a ticket code, and for this field in it already has been associated with a user (code used) the application automatically performs the log in and takes the client to the main screen. If the code is being used for the first time, the client is prompt to choose a username that must be unique, and the user account is created. After this only the ticket code is necessary to log in for future sessions.

4. Advanced Features

4.1 Time Quizzes

When a client requests a quiz, the server responds to him with the mentioned quiz and saves the current time of the response associated to the specific client. At the time the client submits his quiz, the server compares the current time with the one of the quiz request and increments the client score accordingly the inverse of the time, meaning that if you get two answers correct in three minutes and other person gets only one but in 30 seconds, the second will have a higher score.

4.2 Security

Both symmetric and asymmetric cryptography is used to guarantee all the security requirements requested. It is assumed that both the client and the server exchange their public keys securely, represented by the first message between the two (HELLO message). The client then sends the ticket code ciphered using a random AES key, and sends this key ciphered with the server public key to guarantee that only this one can read the content of the message. After the server validation, a session id is generated along with a session key (128 bit AES key), used to cipher all the following messages. When the client logs out of the application the session key becomes invalid, forcing a new session key to be generated for a future session, guaranteeing perfect forward secrecy between sessions.

The algorithm chosen for the asymmetric ciphers is RSA with keys of 2048 bits with “PKCS1” padding because of the wide compatibility of this algorithm between Android versions. For the symmetric cipher AES is used with keys of 128 bits, due to the better performance characteristic of this type of cipher. Also all the messages have a digest field to guarantee integrity being that SHA-256 algorithm was the chosen one for this.

5. Implementation

- *The server was implemented with JAVA using “IntelliJ IDEA” as;*
- *SQLite;*
- *Android components:*
    - *Termite service connection, only active when a client has his WIFI Direct On, and it is only present on Main and Quiz activities.*
    - *Log in activity*
    - *Sign up activity*
    - *Main activity*

    - *Quiz activity*
    - *History activity*
    - *Ranking fragment*
    - *Monuments fragment*
    - *The communication between activities is illustrated in the "Attachments" section.*

- *The communication in the client application is threaded, so that every time the client needs to send a message a thread is allocated to send the message and waits for a response from the server.*
- *The client sends a message and waits for a response from the server, then process the message accordingly. The server waits for incoming messages, when one is received it is processed and a response is returned to the client with the result of the operation.*
- *Persistent state maintained on the mobile device*
- *Some activities use "ConstraintLayout" for better performance and tooling support.*

6. Limitations

Currently, when you finish a quiz the check box does not stays checked immediately, it is needed to press the monuments tab to force the view to refresh and turn the check on to that specific quiz. The same happens to the ranking tab, as it may be necessary to press it more than once to update it and display the latest version.

There is no differentiation between foreigners and locals, meaning that the communication with the server when submitting a quiz, it is performed as they all have a mobile connection to the internet.

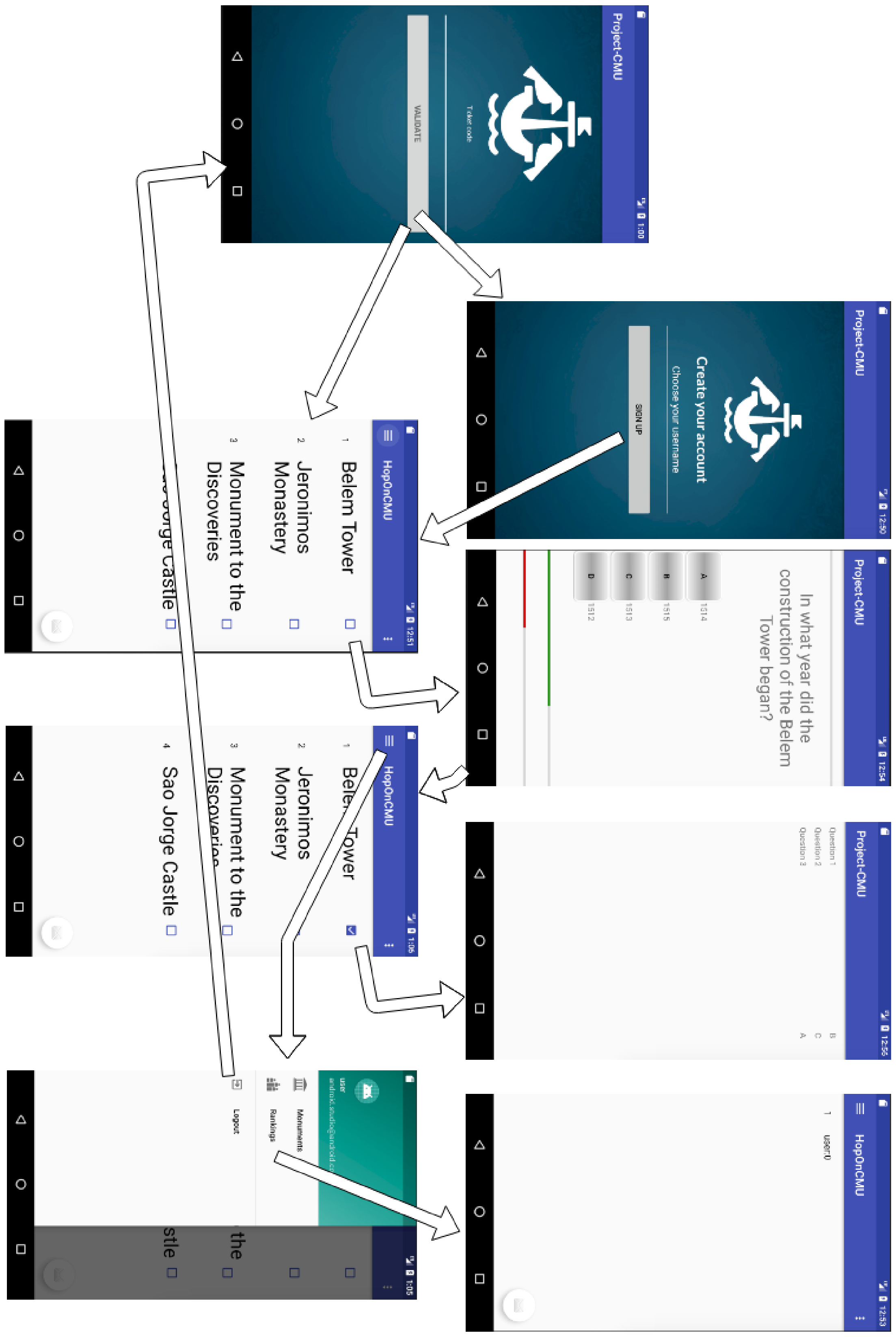
7. Conclusions

For future work, in terms of security the storage of the keys can be done using the Android Keystore that lets you store cryptographic keys to make it more difficult to extract them from the device. Moreover it offers facilities to restrict when and how the keys can be used or restrict keys to be used only in certain cryptographic modes.

A timer on the client side was not implemented due to the possibility the same changes the local time and with that changing the real time that he takes to answer the quiz. Another reason was that with the timer could be possible to the client to get in the quiz see all the questions, return to the previous activity and come back to the quiz and answer it quickly due to the fact he already knows the questions.

In terms of the practical component we believe that it could improve by giving more support to the students such as provide the solutions for each lab class.

8. Attachments


Project-CMU
Ticket code
VALIDATE
Create your account
Choose your username
SIGN UP
HopOnCMU
1 Belem Tower
2 Jeronimos Monastery
3 Monument to the Discoveries
4 Sao Jorge Castle
In what year did the construction of the Belem Tower began?
A 1514
B 1515
C 1513
D 1512
Question 1 B
Question 2 C
Question 3 A
1 user0
user
android.studio@android.com
Monuments
Rankings
Logout