INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Project Assignment for Ubiquitous and Mobile Computing

MEIC/METI 2016-2017 – 2nd Semester

LocMess

1 Introduction

The goal of this project is to develop a distributed mobile application named LocMess, which aims to provide a set of location-aware messaging functionalities to users in urban centers. As depicted in Figure 1, LocMess must allow users to post messages on certain *locations* (e.g., "Arco do Cego Park", "Deja Vu Coffee House", "Fitness Hut Lisboa"). Consider, for example, this usage scenario. Alice has just moved to another flat and she's looking for a roommate. Because she knows that students tend to hang out at the Arco do Cego Park, LocMess will allow her to post an advertisement to the visitors of this park. Visitors running LocMess on their mobile devices will receive a notification whenever they roam around this area. They can then receive and read the message sent by Alice. Location names are associated with geographical coordinates, which can be specified as GPS coordinates or WiFi identifiers. For message forwarding, LocMess must support message delivery over both an infrastructure-based network (i.e., 4G or WiFi) and an infrastructureless wireless network (i.e., WiFi Direct ad-hoc network).

So, the idea is to develop a mobile application that mimics the use of a traditional post-it note with some written text that can be glued on some wall. Others may pass close to the wall and, if interested on the subject, may read the post; or may make a copy of it and carry it with them to deliver it later to someone else. The following explains in more detail the functionality of the system.

The target platform for LocMess is Android version >= 4.0. WiFi Direct will be used for wireless communication.

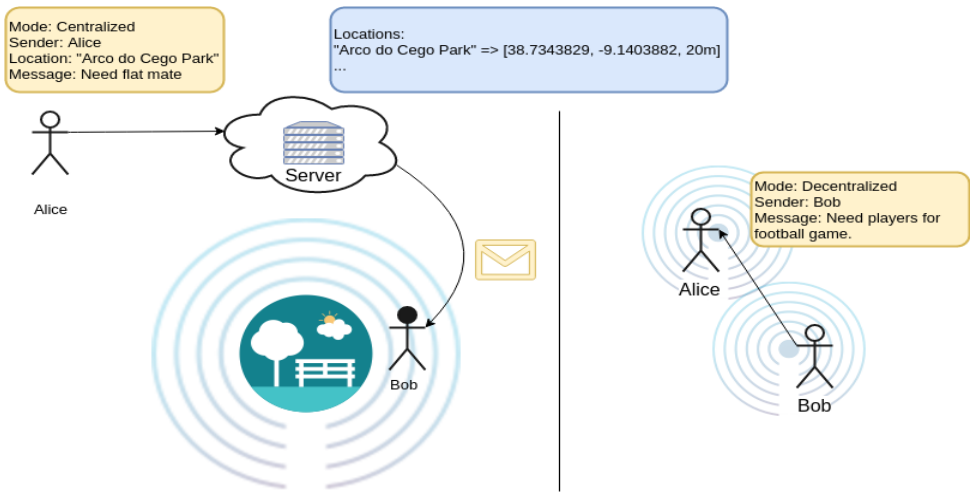

Figure 1: LocMess usage scenario.

This project will give students the opportunity to acquire several skills in the field of mobile and ubiquitous computing, namely the ability to: (a) design a mobile application based on a list of requirements, (b) implement an Android application, (c) manage mobile wireless networks based on WiFi Direct, (d) handle state replication and

consistency in mobile wireless networks, (e) develop adaptive techniques to improve resource utilization, data availability, and performance, and (f) develop security mechanisms for mobile applications.

2 Specification

2.1 Baseline Functionality

LocMess is a mobile application that allows users to publish messages based on users' locations. Users can: i) receive location-based messages from a centralized server, and ii) receive messages from nearby devices.

2.1.1 Architecture and Functions

The basic architecture of LocMess relies on a central server and a client mobile application. The central server is accessible through the Internet and is responsible for managing the state of the system. The client is a mobile Android application that users install and run on their devices and allow users to perform the following functions:

F1. Sign up

F2. Log in / out

F3. List / create / remove locations

F4. Post / unpost message

F5. Read message

F6. Edit user profile, list profile keys

The sign up operation allows users to create a new user account in the system; the user must enter a username and a password. The client then contacts the LocMess server, which must ensure that the new username is unique. If the operation is successful, the user can then log in and start a new session on the client.

To perform useful functions on LocMess, the user must log into the system with his account credentials (username and password), which must be validated by the server. If they are valid, the server must generate a new session id, keep an internal record associating that session id to the username and return the new session id to the client. However, if the credentials are invalid, the server must return an error. Only the users with valid session ids will be allowed to perform additional functions (F2-F6). The log out function ends the current session.

Once a session is active, users are able to manage locations. A location is a tuple that associates a human readable name with a set of a set of geographical. Coordinates can be expressed in two possible formats: GPS or WiFi IDs. In GPS, a location is defined by the latitude and longitude of a given spot and a radius in meters surrounding that spot, for example the location "Arco do Cego Park" => [38.7343829, -9.1403882, 20m] represents a circular area around a spot characterized by latitude and longitude (first and second fields of the tuple) and radius of 20 meters; any device positioned within this circle will be considered to be located at "Arco do Cego Park".

Geographical coordinates can also be expressed as a list of WiFi IDs, meaning that if a device is able to sense a WiFi signal from any of the WiFi IDs in the list, LocMess will acknowledge that device as being positioned on that location. This ID can be the SSID of a real WiFi router or (set of WiFi routers), for example "eduroam-rnl", or can be the SSID of a well-known BLE beacon (Bluetooth Low Energy) placed deliberately on some public place for advertisement purposes (e.g. "h3-monumental").

The locations are specified by the users and submitted to the server, which maintains a global list of all existing locations. Users can freely insert new locations into the system or delete existing ones. They can also list existing locations, even if these have been inserted by other users. Users can also delete locations that have been created by other users. The name of a location is just a string. In the basic version, assume that the locations are correctly specified and do not refer to misleading physical coordinates.

2.1.2 Profiles

Each user has a local profile, which consists of a set of key pairs that can be used to specify properties about the user, in particular interest topics (e.g., the preferred football club). These properties allow message senders to narrow down the scope of a given message within a target location, e.g., "send a message to the Real Madrid fans at Arco do Cego Park". LocMess must allow the user to change his profile by adding or removing key pairs (e.g., "club=Real Madrid"). Their profiles are maintained by the server, but can also be cached on the user's device.

The user is free to add arbitrary key pairs to his profile. The profile of each user is private, but the keys of each keypair created by any user are public; the server maintains a list of all users' keys and let them obtain this list. This is useful whenever a user wishes to send a message and needs to restrict the potential receivers of that based on a profile key.

2.1.3 Messages

Users can post messages on certain locations; the location name of the target message destination must be provided by the user and retrieved from the list of valid system locations. The user can further limit the set of users on the target location that can read the message by providing a simple policy, which consists of: a policy type, a list of profile keypairs, and a time window. Two policy types exist: whitelist and backlist. In the first case, no users on a given location will receive the message except those that match the key-value list, e.g., {"Arco Do Cego", whitelist, "job=Student"}, means that the message will be forwarded only to the users located at "Arco do Cego" whose job is "Student". In contrast, blacklist policy allows all users on that location to receive the message excepting those specified in the list. To receive all messages on a given location, the sender can use a whitelist policy without any specific restrictions, i.e., a null restriction list. The time window indicates the time frame where the message should be visible to users, e.g., 10:00 March 2nd – 23:00 March 10th. Messages are only visible to users that satisfy all the conditions of the posting policy. The message owner can also decide to remove the message at any time by issuing an unpost operation.

If a user visits a given location and his profile satisfy the conditions of posted messages for that location, then a notification is popped up on the visitor's device and the user can receive the post and read it. If the user receives the message, then the message will remain available to the user even if he leaves the message's target location or the message visibility window expires. Otherwise, if the user does not explicitly receive the message, then it ceases to be available to the user if one of such conditions occurs, i.e., the user abandons the target message location, or the time window expires.

If a user receives a given message, he will be shown the content of the message, the name of the publisher, and the time of publication. In the baseline version, LocMess will not explicitly allow the user to send messages back (i.e., reply) to the original publisher. For this reason, if the publisher wants to be contacted back, he must leave some contact information in the message itself, e.g., an email address, or a phone number.

2.1.4 Delivery Mode

When posting a message, the user can choose the preferred message delivery mode, between *centralized* and *decentralized*. In the first case, the messages are forwarded to the destination through the server. To post a message, a client connects to the server and uploads the message details and respective posting policy. The server buffers the message and notifies the clients that satisfy the conditions for receiving the message. To determine the current location of each client, the server expects each client to periodically connect to the server and advertise its current location (GPS coordinates and visible WiFi IDs). The server keeps track of each client's current position, and matches it against each posted message. If matches exist, the clients are notified that messages are available to them which they can receive. If they receive a message, the message is downloaded to the client's device and locally cached there. Under the centralized delivery mode, devices that have no connections to the server are neither able to post new messages nor to be notified of available messages for them.

On the other hand, in the decentralized delivery mode, the message is not sent through the server, but is sent to the destination location by one or more nodes. In the simplest case, the publisher itself delivers the message to the

devices in the target location; the message is stored on the publisher's device, which keeps tracking its own location (GPS or WiFi IDs). If this location matches the target location of the message, then it will scan for other nearby devices in order to detect those that satisfy the conditions of the message's posting policy. The scanned devices must reply indicating whether a match exists or not. If so, they inform the publisher node which then sends to it (the scanned device) a copy of the message. This message is now ready to be received by the local user.

Note that in this basic version, only the publisher can carry his own messages to other nodes in the target destination. Receiving devices can only show the message to the local user; they cannot forward it to other nodes. If the publisher never visits the target destination of the message, then it will never be delivered to other nodes.

2.2 Advanced Features

In addition to the baseline features, students must also implement two advanced features (to get the total classification, i.e., 20 points) as described next:

Relay routing. This feature aims to speed up the delivery of messages in decentralized delivery mode by relying on additional nodes to help carry messages to the target destination – such nodes are called *mules*. Assume for simplicity that a message can have at most one-hop mules, i.e., the publisher can pass the message to a neighboring node, but this node – itself a mule – can only deliver the message to final nodes, i.e., nodes in the target destination; the mule cannot forward the message to additional intermediary mules. Thus, the solution must be able to support several mules, but each mule is allowed only to carry messages to the final destination.

Note that the mules can be elected to forward messages on behalf of others even if the profile of the local user does not match the posting policy of the message. Mules have a limited space for transporting third-party messages. This limited space is a configuration parameter that must be specified by the local user. Students have the freedom to implement any algorithm that may help to maximize the probability of a message arriving at the destination. In particular, instead of forwarding the message to all nodes it bumps into, the publisher can be more intelligent at picking the mules. Solutions that reduce the number of messages exchanged or the power consumption will be favored.

Security. In the baseline setup the security is absent. Students are invited to enhance the security of the system in terms of attaining the following security goals: (1) ensure that the communication between clients and server is secure against an adversary with the ability to eavesdrop the network, modify the packets exchanged, suppress packets, or introduce new packets, and (2) ensure that a message receiver can authenticate the message as published by a specific user and that it has not been tampered with during the transmission.

3 Implementation

The target platform for LocMess is Android and the source code of the mobile application must be written in Java. Students can use android APIs freely. However, third-party libraries are disallowed unless explicitly approved by the faculty. The central server must be implemented as a standalone Java application.

Unfortunately, it is not possible to provide real Android devices and BLE beacon stickers to test the project. Therefore, the project must be tested on a software-based emulation testbed, which comprises the native Android emulator and the Termite Wifi Direct emulator. This testbed will allow you to emulate: 1) users' devices executing the mobile application, 2) communication between devices, 3) beacon signaling, 4) GPS tracking, and 5) communication with LocMess server. The testbed can be set up on a single computer. This computer can also be used to host the LocMess server.

The Android emulator comes natively with Android SDK and provides support for GPS and Internet connectivity. However, it does not emulate WiFi Direct and Bluetooth APIs.

Termite is a WiFi Direct network emulator. Termite can emulate a wireless network of virtual nodes that move over time and are able to interact opportunistically whenever they are located within close range. Virtual nodes consist of Android emulator instances running the test application. The developer is responsible for specifying the topology and dynamics of the virtual network.

In the context of LocMess, Termite must be used to emulate WiFi Direct communication between users' devices and signaling / sensing of BLE beacons. Since BLE beacons use Bluetooth to advertise their location, they cannot be simulated directly by Termite. Therefore, for the sake of simplicity, BLE beacons must be simulated as normal WiFi Direct nodes that are used uniquely to broadcast a network identifier to the devices of nearby cyclists. Termite will be made available on the course website.

The interface to be provided by the app LocMess should be simple and, at the same time, complete, i.e. show in a clear way all the functionalities supported.

4 Development Stages

We recommend you to develop the project in five stages:

1. **GUI design:** study the requirements of the project and design the graphical user interface of your application. Create an activity wireframe of the application.
2. **GUI implementation:** implement all graphical components of your application including the navigation between screens. At this point don't worry about networking. Use hard-coded data to simulate interaction with the server or beacons. Make sure to design your application in a modular fashion.
3. **Communication with server:** implement the central server and extend the mobile application in order to communicate with the server.
4. **WiFi Direct communication:** complete the baseline functionality of the project by implementing WiFi Direct communication. You only need to use Termite at this point.
5. **Advanced features:** implement advanced features about security and robustness.

5 Grading Process and Milestones

The projects will be evaluated based on several dimensions. The most important points are: baseline and advanced features implemented, modularity of the implementation, technical quality of algorithms and protocols, and resource efficiency decisions. We will also assess the responsiveness and intuitiveness of the application interface. However, the aesthetics of the GUI will **not** be considered! Therefore, as said above, keep the GUI as simple as possible while providing the needed information

There are three important project milestones:

- **May 13: Project Submission** – a fully functional prototype of LocMess and a final report. The prototype sources and the report must be submitted through the course's website. A template of the report will be published on the website. The report must describe what was and was not implemented, and it is limited to 5 pages (excluding cover). The cover must indicate the group number, and name and number of each of the group's elements.
- **May 15-19: Project Demonstration** – Each group will have 15 minutes to present the developed prototypes. Each group must prepare a set of tests for the demonstration. These tests must be carefully prepared in order to showcase the functionality of the application.
- **May 22-June 26: Project Discussion** – The discussion will take 15 minutes per group. The final grade awarded to each student will also depend on his performance in the oral discussion (in addition to the project itself) and may vary within each group.

  In addition, there will be a checkpoint approximately in the middle of the semester so that feedback can be given to students about the status of their project. More details will be given on the web page.

Good Luck!