INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Project Assignment for Ubiquitous and Mobile Computing

MEIC/METI 2015-2016 – 2nd Semester

UbiBike

1 Introduction

The goal of this project is to develop a distributed mobile application named UbiBike, which aims to provide a set of functionalities to cyclists in urban centers. The target platform for UbiBike is Android version >= 4.0. WiFi Direct will be used for wireless communication.

This project will give you 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.

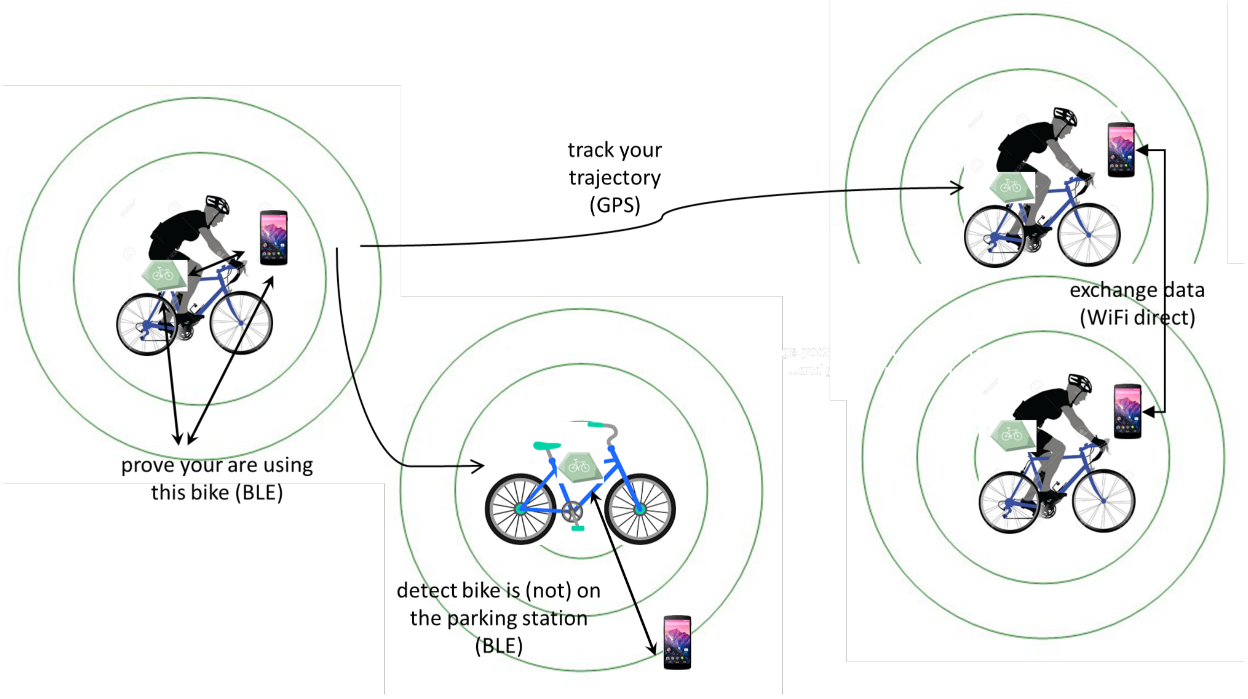

Figure 1: UbiBike usage scenario.

2 Specification

2.1 Baseline Functionality

UbiBike is a mobile application that allows cyclists to earn and share points as they cycle (see Figure 1). Cyclists earn points by cycling (more traveled distance translates into more earned points). Additionally, cyclists can i) send/receive points to/from other cyclists, and ii) send/receive text messages to/from other cyclists. Communication between cyclists (sharing points and exchanging text messages) is done through WiFi Direct.

The UbiBike application is responsible for tracking the trajectory (using GPS) while cycling. The tracking starts when the user approaches the bike. Using GPS, UbiBike tracks the distance (and trajectory) traveled using the bike and stops when the user moves away from the bike. Each bike has attached a BLE (Bluetooth Low Energy) beacon sticker, which allows UbiBike to detect when the user is using a (specific) bike or not.

Bikes are picked and dropped at parking stations. All users must register (using UbiBike) in a central server. The server knows about all users currently registered in the system, including their current score (total number of points) and trajectories. This data (score and trajectories) is opportunistically uploaded by the UbiBike app (running on users' device) to the central server. Additionally, by contacting the central server, the mobile application can be used to book an available bike from a particular parking station.

The server knows which bikes are available at each parking station because the UbiBike app informs the server whenever a bike arrives or leaves the station. To detect such events automatically, the app can use heuristics based on GPS readings and BLE beacons sensing. The app running on a cyclist's device can detect a bike pickup event by comparing the current GPS location of the device with the GPS coordinates of the parking station where the bike is stationed, and perform subsequent GPS readings in order to detect that the device is moving away from the parking station. To ensure that the cyclist is riding the bike (and not simply walking away), the application must be able to constantly sense the wireless signal of the bike's BLE beacon. To detect a bike drop-off event, the reasoning is similar except that instead of detecting that both device and beacon are departing from the parking station, the application must detect that they are approaching a parking station and eventually get immobilized, and the user moves away from the bike.

To sum up, for this project, you need to support the following interactions:

- Between mobile devices (using WiFi Direct)
    - Send and receive points
    - Send and receive text messages
- Between mobile devices and the central server
    - Register user
    - Send new trajectory
    - Show most recent and past trajectories on a map
    - Get user information (including current score and trajectories)
    - Get list of stations with available bikes to book
    - Book bike at specific station (while showing their location on a map)
- Between mobile devices, BLE beacons, and the central server
    - Notify bike pick up
    - Notify bike drop off
- Between the user and the bike being used (using BLE)

- Detect what bike is being used by which user

2.2 Advanced Features

Students can also implement two advanced features for extra points in the final classification: security and robustness. These features do not apply to text messages exchanged between users. Note that to implement these features, you can trust the application, i.e., assume that no one is able to attack the mobile device and compromise the UbiBike app. However, you cannot trust the network.

For handling security, when considering messages used to exchange points from one user to another, the system must be resilient to message replication and tempering. For example, if someone eavesdrops a message containing points, it must be impossible to: i) resend the message multiple times to the same user or to different users, or ii) tamper with the data in the message (for example, increase the number of points in the message).

Regarding robustness, the system must support the reordering of events. Since the central server is updated opportunistically, users can update the central server with information that depends on events that are still not in the central server. For example: imagine that user A cycles to gain points and then sends all the points to user B. Now, user B updates the central server before user A reports that it gained points by cycling. When such situations occur, the system must be able to: i) accept the points given to user B, and ii) update the score of user A. In other words, the server must always know how users earned their points.

3 Implementation

The target platform for UbiBike 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) cyclists' devices executing the mobile application, 2) communication between devices, 3) beacon signaling, 4) GPS tracking, and 5) communication with UbiBike server. The testbed can be set up on a single computer. This computer can also be used to host the UbiBike 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 UbiBike, Termite must be used to emulate WiFi Direct communication between cyclists' 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 UbiBike 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. GPS tracking can be implemented at this phase. 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: features implemented, robustness and 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 esthetics 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 UbiBike and a final report. The prototype sources and the report must be submitted on the 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 16-20: Project Demonstration** – Each group will have 20 minutes to present the developed prototypes of UbiBike. 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 23-27: Project Discussion** – The discussion will take 30 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 to said feedback can be given to students about the status of their project. More details will be given on the web page.

Good Luck!