Practical Exercises

Exercise 2: Project

To prepare for the project phase of this course that will start in the first week of lectures after the Christmas vacation, you will need to get familiar with the Transport4You (T4Y) system. T4Y is an e-Ticketing system, specifically developed to fit the needs of bus transportation companies in small-to-medium sized cities. The TY4 system is supposed to support the ticket purchasing, ticket verification, and accounting process.

The requirements for T4Y were elicited involving real-world stakeholders and have been gathered in the provided `Requirements_Specification.pdf` document. A substantial portion of the T4Y requirements already have been implemented by an external contractor. The respective codebase can be found in the `T4Y_Codebase.zip` archive in the lecture material.

For the project phase you should build small groups of 3-5 students that together work on the following tasks. Although you can distribute the work among the different group members, **each member must be able to report on all aspects of all tasks**!

Phase 1: Analysis and Maintenance

Task

Based on the provided documents and codebase, please answer the following questions:

- How does the architecture of the current prototype look like? What are the main system components and how do they interact? What are the benefits and shortcomings of this architecture?
- Is there a gap between the requirements that were specified by the customer and the actual functionality of the prototype? If yes, explain the main differences between what was specified in the requirements specification and what actually has been implemented.

Furthermore, please conduct the following tasks:

- Try to get the full system running by deploying the different components on a) your Android emulator, b) the GAE infrastructure, and c) either a RaspberryPi or your laptop.
- Try to run a full workflow on the system, including ticket creation on the web system, ticket synchronization with the mobile system, and ticket exchange with the bus system. If you encounter any bugs, fix them!
- In the end the full system needs to be able to run all of the aforementioned steps.

Material

- Requirements specification document: `Requirements_Specification.pdf`.
- Codebase for T4Y: `T4Y_Codebase.zip`.
- RaspberryPis and Android tablets can be used during the tutorial sessions or be borrowed for a few hours from Severin Kacianka, room: MI - 01.11.039 on first-come-first-served basis.

Deliverable

- Presentation on the answers to previous questions.
- Demonstration of running prototype.
- Deadline: Tutorial in the week of 11.01.2016.

Phase 2: Functional Extension

Task

The current prototype uses a custom bluetooth protocol to exchange tickets between a customer smartphone and the bus system. This ticket exchange is done to verify, if the presented ticket is unused and still valid. Unfortunately, the bluetooth connection is not very reliable and has issues with handling multiple request within a short time frame. The customer thus asked you to replace the bluetooth-based ticket validation by a QR-code-based approach. This is, instead of communicating via bluetooth, the mobile system should display a representation of the ticket as QR-tag that then can be scanned by the bus system's camera. Based on information it got from the web system and the scanned QR-code, the bus system should then be able to decide if the presented ticket is valid and unused, and inform the bus driver about the ticket status.

In detail, your tasks to realize this new requirement are as follows:

- Come up with a suitable concept to represent the e-tickets as QR-codes.
- Come up with an idea for a tamper-proof way to check the validity of e-tickets.
  (*Hint: you might want to use hash functions and signatures for this.*)
- Add a new function to the T4Y Android app to display QR-code representations of stored e-tickets.
- Add a new function to the bus system to scan QR-codes on demand.
- Add new functionality to check the integrity of the QR-code and check validity of the presented e-ticket.
- Add new functionality to signal to the bus driver, whether the presented e-ticket is valid or not.
  (*Hint: you can do this via the built-in display, the PiFace LCD display, or sound output.*)

Material

- Tutorial on how to create and encode data into QR-code tags in Android using the *ZXing* library:
  http://www.mysamplecode.com/2012/09/android-generate-qr-code-using-zxing.html
- Library and tutorial on how to grab a picture from a webcam in Java:
  http://webcam-capture.sarxos.pl/
- Tutorial on how to decode a QR-code tag using the *ZXing* library:
  http://javapapers.com/core-java/java-qr-code/
- If you want to do this in Python instead of Java, here is everything you need for this:
  http://github.com/malex984/dockapp/wiki/QR-Code-scanner

Deliverable

- Presentation of QR-code extension of the prototype (concept, workflow, implementation).
- Demonstration of running prototype.
- Deadline: Tutorial in the week of 01.02.2016.