Net-computing
Weekly progress report #4

Peri Rahamim (s2683423),
Jits Schilperoort (s2788659),
Twan Schoonen (s2756978)

22.3.2018

Week 4

Overview

This week we worked on the implementation of the project.

- **Web services** Implemented user authentication and user storage.
- **Map** Graphical representation of a city with lists of costumers and cars that interact with incoming orders
- **App** Worked on the app on which users can create accounts, log in, send car requests, see the map, etc.
- **Message Queue** Implemented a message queue that takes in car requests and pushes them in the queue of the relevant center.

Tasks

Done

- Map of the area
- Message queue
- Web services

Future

- Finish the app such that a user can log in or create an account and send car requests
- Connect the map with the other parts so you can track the events live.

Work division

We worked on it together during the lab and had 2 more group meetings. In total we worked on the project around 10 hours each.

- Peri: Map of the cars
- Jits: Message queue, app
- Twan: Web services, app

Week 3

Overview

This week we started implementing the project. We chose to do it in Python.

- **Server-Client** We built basic server-client connection.
- **Sockets** Our project has basic socket communication, it is not used at the moment.
- **REST** REST application is implemented by using $json$ files that contain clients' data.
- **Database** The implementation of REST made us build a database that has clients' data that will be used in the application.

Tasks

Done

- Working server-client communication.
- Working socket implementation.
- Database consisting of clients' data.

Future

- Implementing communication using TCP protocol.
- Having center-car communication that send information using sockets (star communication).
- Having center-center communication (peer to peer).

Work division

We worked on it together during the lab and had 2 more group meetings. In total we worked on the project around 7 hours each.

- Peri: Writing database files, server-client.
- Jits: Server-client and REST implementation.
- Twan: Building server-client connection and implementing sockets.

Week 2

Overview

- **Architectural design** This week we mainly focused on the architectural design of the project, a document with this design is supposed to be handed in this week.
- **P2P** Furthermore we discussed the way in which we think we should implement the communication between subjects of the system. It was suggested to us to think about p2p communication which we first considered in car2car communication. We found out that this would not be very efficient in our case (in autonomous driving it would, but we will not implement that). So now we considered implementing p2p as communication between multiple centers. Each district would have its own center which can communicate with the centers in other districts (p2p) in the case of interdistrict transportation.

- **Car sharing** We also considered the possibility of car sharing. Whenever a car is occupied but it is the nearest car to a new costumer, it should be possible to have the option of putting multiple customers in the same car.
- **Car hopping** Another extension of the system includes the possibility of car hopping. Especially in long drives it could be more efficient to put people together. So when, for example, two people have to go from district A to district B but they are at different positions in district A, they can be brought together and put in one car.
- **Queue priority** When it has to be decided which customer has to be picked up by a certain car, some customers can have higher priority than others based on e.g. distance to the car and waiting time.

Tasks

Done

- Decided on the way the central, the cars and the clients should communicate with each other
- Architectural design document
- Created more use cases
- Consider Star and P2P
- Reconsidered our view on car2car communication
- Extended the idea of the system with car sharing and hopping

Future

- Process any feedback on the architectural design
- Practice with RabbitMQ
- Start looking for python libraries relevant to our system
- Start on the actual implementation of the system
- Think a little more about the queue prioritization

Work division

- We worked on it together during the lab and also made another appointment. In total we worked on the project around 5 hours.
- All: Discussed new ways of implementations.
- Peri: Mainly worked on architectural design
- Jits: Mainly worked on the weekly progress
- Twan: Mainly worked on architectural design

Week 1

Overview

This week we focused on the correct way of implementing the system. The complete project consists of a mobile application, Artificial Intelligence technologies, shortest distance calculations, wireless communication. We decided it is necessary to identify the parts of the system that are relevant for this course and, because of lack of time, implement these parts of the system alone.

We analyzed each element of the system to decide on relevance:

- **Mobile application:** Each costumer that is interested in getting a car to drive them uses a mobile app to make the order. A working app with an user friendly GUI is irrelevant for this course, so we decided to have some data structure that represents a user and has location and number of people who want to use the service instead.
- **AI:** The automated driving cars should know how to drive on roads without the help of humans. The AI of the car should follow the law and consider other cars (agents) or people crossing the road in its surrounding. This part is obviously too much to implement, and since it is also irrelevant to this course, we decided to assume the AI works.
- **Shortest distance algorithm:** We want the car to find the best route to reach its costumer, considering traffic and other parameters (such as construction that blocks the road). The algorithm helps deciding which car is most suitable to get to a costumer, in case there are few of them in different parts of the district. To make this work we need live data, so for this project, we only send messages with datasets we made ourselves, and leave out the complicated calculations.
- **Wireless communication:** The cars should communicate with the center and with each other, deciding which costumer to get to first. We will implement this part of the system, that covers socket and message queuing.

Tasks

At the moment, we have made decisions about what we should implement and how, this of course might be subject to changes, but for now we have a direction we can follow.

Done

- Have an overview of the complete project.
- Idea description document.
- Deciding on programming language for the project (python).

Future

This section will be more detailed as we move forward with the project.

- Architectural design document
- Car-Center communication.
- Car-Costumer communication.
- Car-Car communication.

Work division

So far, almost all of the work was done in meetings we had, where we discussed about how we should approach the problem. This week each member of the group invested between 3 to 4 hours.

- Peri: Group meetings+documentation.
- Jits: Group meetings+documentation.
- Twan: Group meeting+initial version of the idea for project.