Faculteit Ingenieurswetenschappen

KU LEUVEN

PROBLEEMOPLOSSEN EN ONTWERPEN, DEEL 3

Team CWA3

Lies Bollens
Ruben De Clercq
Jakob Festraets
Rugen Heidbuchel
Floris Kint
Peter Lacko

Quantified bike

PROGRESS REPORT

Supervisor
Prof. dr. ir. Erik Duval

Assistants
Sven Charleer
Jose Luis Santos
Robin de Croon
Joris Klerkx

ACADEMIC YEAR 2014-2015

Contents

List of Figures

List of Tables

1 Group members

The group consists of the following members.

1. Lies BOLLENS, Bachelor of Science in de ingenieurswetenschappen, 2nd year
2. Ruben DE CLERCQ, Bachelor of Science in de ingenieurswetenschappen, 2nd year
3. Jakob FESTRAETS, Bachelor of Science in de ingenieurswetenschappen, 2nd year
4. Rugen HEIDBUCHEL, Bachelor of Science in de ingenieurswetenschappen, 2nd year
5. Floris KINT, Bachelor of Science in de ingenieurswetenschappen, 2nd year
6. Peter LACKO, Bachelor of Science in de ingenieurswetenschappen, 2nd year

2 Brainstorm

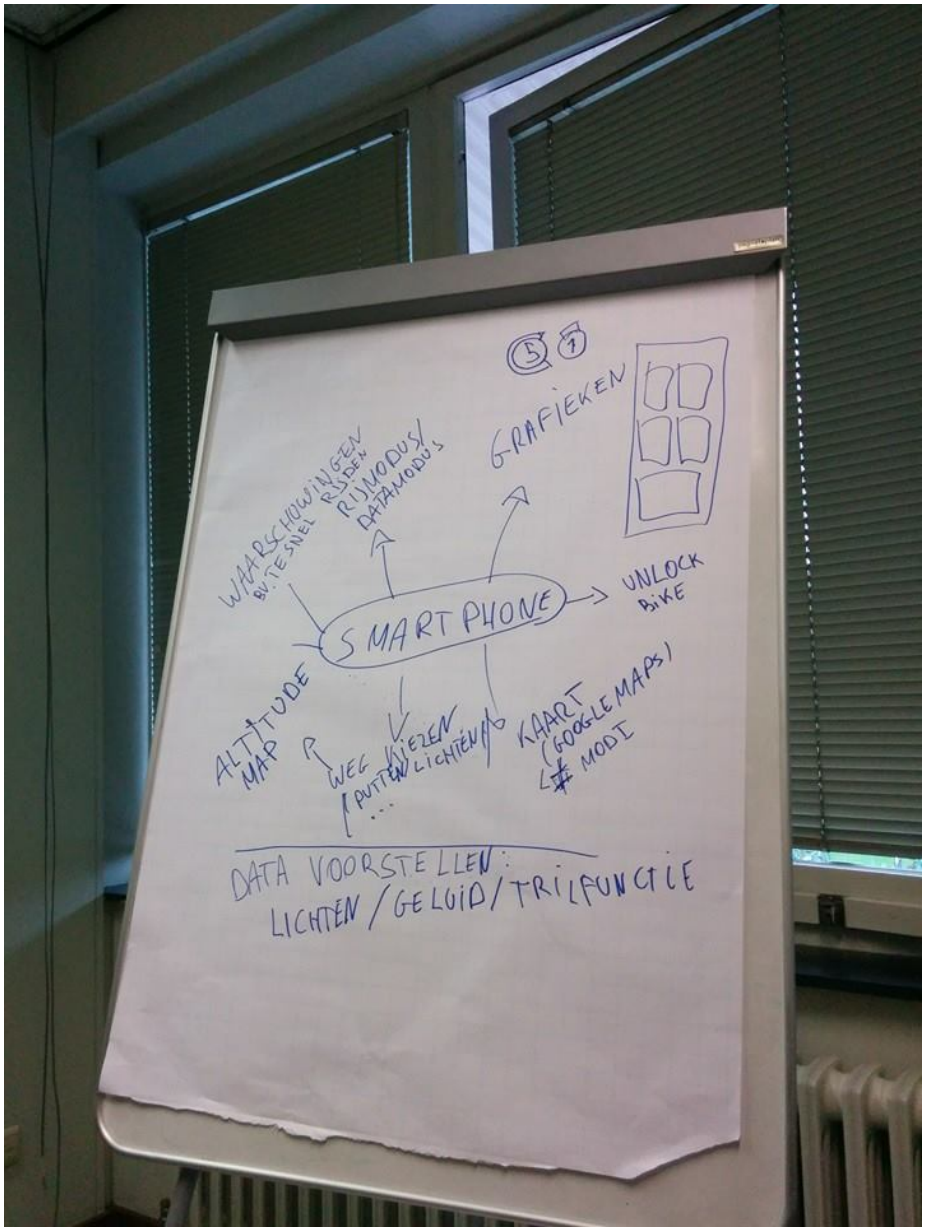

Figure 1: First brainstorm

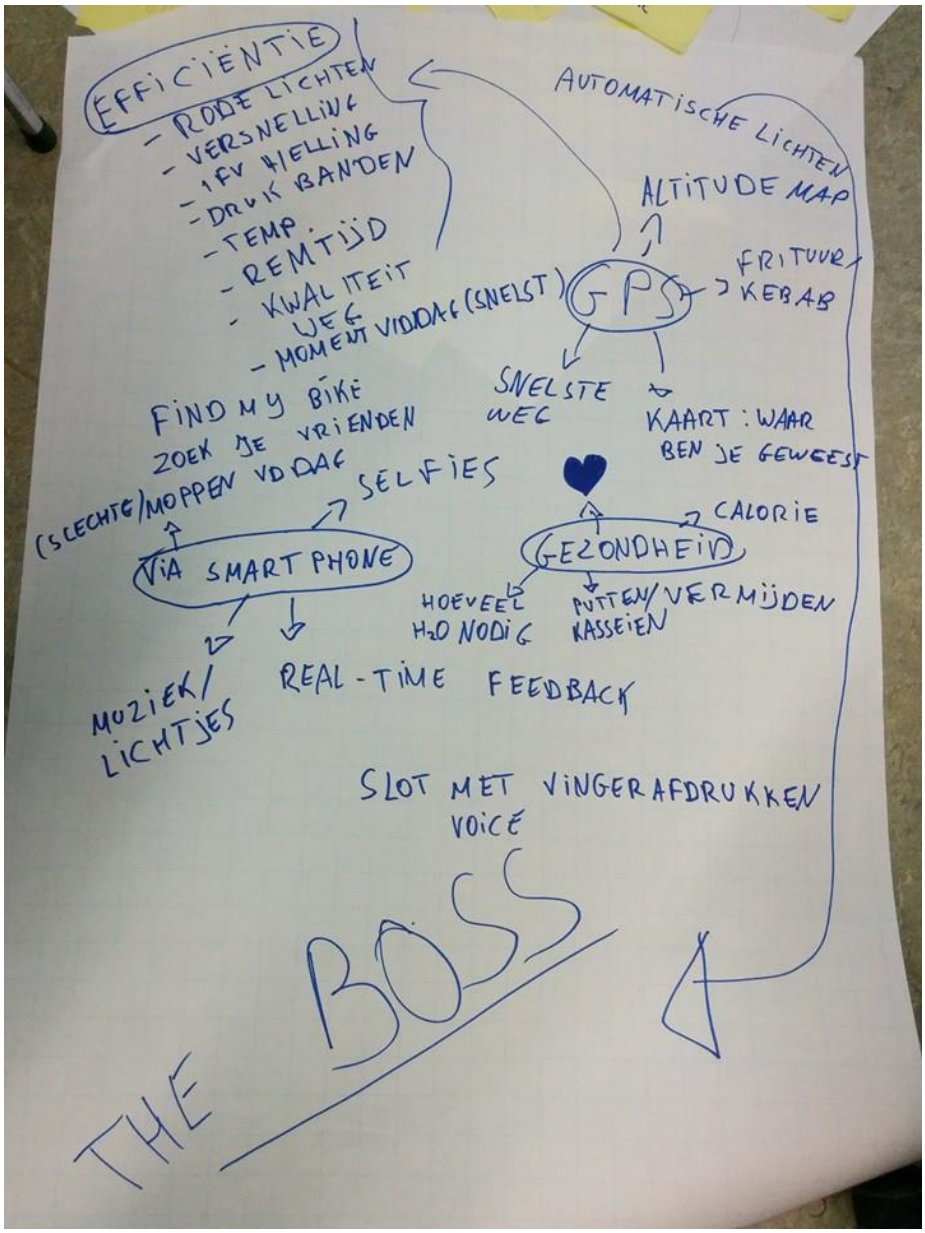

Figure 2: Second brainstorm

A list of all the ideas created during the brainstorm.

1. Using a smartphone
   (a) get real-time feedback
   (b) play music/ put on a light
   (c) take pictures of yourself/ your surroundings
   (d) easily find where your bike is parked

(e) view in an instance where your friends are

2. using a GPS
   (a) get the fastest way from one point to another
   (b) display a card, showing all the places you have been to with your bike.
   (c) Make an altitude map
   (d) Easily track the distances you've covered.
   (e) display some points of interest (ex. All the pizza restaurants nearby)
3. try to make your trip as efficient as possible
   (a) avoid red lights, based on your previous bike trips
   (b) get a map, displaying the quality of the roads
   (c) find out during which moment of the day you cover a certain way the fastest
   (d) get the average time needed to brake
   (e) view your speed, based on the altitude difference, the temperature, the hour,...
   (f) avoid bad ways
4. monitor your health
   (a) how many calories have you lost
   (b) how much water do you need

The initial idea was to use a smartphone, since these devices contain almost all the sensors needed for the project. This way, creating an app for Android and iPhone devices would be enough for the concept to be functional. The assignment, however, was to incorporate an Arduino Nano and a Raspberry Pi. Since using a smartphone, an Arduino and a Raspberry Pi would get a bit too complicated, our team decided against using a smartphone. This meant that there would not be any screen attached to the Boss device anymore. Real-time communication with the user was therefore no longer possible and functions such as find-your-friends and find-your-bike not doable anymore.

A lot of the ideas created during the brainstorm, focused on getting an augmented bike. This is not what the Boss device was aiming for: the purpose of this project was to create an quantified bike device, without looking at the augmented side of a bike trip. Once this goal had been set, a lot of the brainstorm ideas could be thrown overboard: get the fastest way, view your calorie deficit, see how much water you need, locate your friends, get points of interest,...

The idea of trying to avoid red lights seemed very appealing at first, but turned out to be too difficult to implement. It would require an accurate route planner that constantly refreshed, while taking all the red lights in the area into account. A second reason against this idea was the fact that, as a lot of other ideas, it is much more on the augmented than on the quantified side of a bike trip.

3 User stories

3.1 Comparing daily trips

Sophia uses her bike and the BOSS on a daily basis. She bikes to class every day and clips the BOSS off her bike so it does not get stolen. On the BOSS website, she can easily review each trip she made in the calendar overview. She can get a general impression of how she did each day, and then she can compare her stats with a ranked list of the stats of her friends. This would include the most traveled distance, the longest bike trips (in time), the fastest speeds, etc.

3.2 Sharing biking experience with friends

Luke likes to go on bike trips for fun and exercise. He sometimes doesn't know where he will go, but he can easily find out where he's been in the BOSS's detail page on the website. When he sees a beautiful view, or just something interesting, he can easily take a picture on the built-in camera with a click of the button. He can also press a second button to mark certain points of interest. The points of interest are displayed with the photos on a map of his route that he can share with his friends on various social media sites (Facebook, Twitter, Google Plus,...) On top of the map, he can share various statistics of his trip (how bumpy the road was, how fast he went, how hot and humid it was) along with personal comments and ratings. If he wants, he can also view the statistics and comments of fellow cycling friends.

3.3 Fitness stats

Johnny likes biking for sport, and loves using the BOSS to record his data. He always keeps the BOSS on his bike, and only has to flick a switch to turn it on or off. After each ride, he goes to the BOSS website to check specific details of how he did, and can compare his performance with previous rides, made possible by the detailed compare view of the BOSS website. Furthermore, he can also compare his data to his competitive friends, by use of simple numbers (for the mean speed, travel distance and travel time), or with graphs (speed during a trip, measured temperature throughout the trip, heart rate, etc.) Finally, he can challenge his friends to do beat his results via various social media sites.

4 Architecture

Our application mainly consists of 3 parts. A Raspberry Pi and an Arduino Nano are mounted on the bike to collect the data. The other one is the web interface.

4.1 Device

4.1.1 Arduino Nano

The Arduino Nano is connected to a GPS Sensor, a temperature sensor, a humidity sensor, a heart beat sensor and a few buttons and LEDs. The code on the arduino interprets the data coming from the sensors and sends it via a

Serial interface to the Raspberry, which is connected via a USB Cable. The interval between consecutive data transmissions depends on the kind of sensor. For instance, the state of a push button is sent more frequently than the GPS data. All data is sent in a specific format. One packet of sensor data consists of a single line. A semicolon is used as delimiter to separate values. The first value of a data packet identifies the sensor. The format of the next values depends on the kind of sensor.

4.1.2 Raspberry Pi

On the Raspberry Pi, a Python application reads data from the Arduino. When an Internet connection is available, the application can send the data to the server in realtime. Alternatively, all trip data can be stored in a local MySQL database. As soon as an Internet connection is available, that data can be sent to the server by a separate script. When that data transmission succeeds, the data is removed from the local database.

4.2 Web interface

5 Used technologies

5.1 Device

5.1.1 Python

Libraries

1. python-serial
2. SocketIO

Threads The application on the Raspberry Pi uses threads so the application can simultaneously read data from the Serial Interface to the Arduino and send data to the local database or the server.

5.1.2 MySQL

When the application can't send data to the remote server, all data is stored on a local MySQL database. MySQL is an open source relational database engine.

5.1.3 Web sockets

5.1.4 Raspberry Pi

5.1.5 Arduino Nano

Libraries

1. INSERT LIBRARIES

5.2 Web interface

6 Course Integration

7 Conclusion

8 Appendix: Workload

9 Appendix: Planning

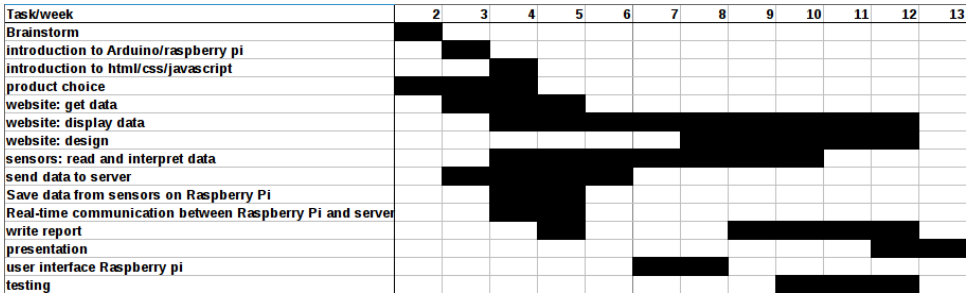

Figure 3: Gantt Chart

10 References