Score Contest Project

Trodometre

A scooter meter

http://air.imag.fr/mediawiki/index.php/Trodom%C3%A8tre

Team members:

Jean-François BIANCO (bianco.jfrancois@gmail.com)

Brice THEOPHILE (b_theophile@hotmail.fr)

Tutor:

Jacques Lemordant / INRIA researcher

POLYTECH GRENOBLE

UNIVERSITE JOSEPH FOURIER SCIENCES, TECHNOLOGIE, SANTÉ

Contents

I) Executive Summary

The goal of the "Trodometre" project is to provide a system to make accurate measurements during a journey. This system is able to calculate both indoor and outdoor measurements. In fact, it doesn't use a localization system for its position in real time. Moreover, GPS is not available indoor and consumes a lot of energy which reduces the battery life of a system. Information collected by the program can be exported in XML format and used in OpenStreetMap. We can trace an itinerary everywhere precisely on the map. So, every circuit can be shared with everybody.
The system is separated into two parts. The first part is the hardware. It is made up of sensors, based on ANT + technology, and moving means (scooter). Then, there is a second part with software which is based on PhoneGap framework for an adaptability on every smartphone and mobile system.

This system will be used by another project, Navigation of Visually Impaired People (http://autonomie.minalogic.net/index.en.html). This project is intended to facilitate mobility and independence of people with visual impairments. It works both indoors and outdoors, and facilitates access to public transport. Our system enables new itineraries to be created quickly and speeds up completion of those available.

We are two French students studying engineering at Polytech Grenoble, the engineering school of Joseph Fourier University (Grenoble, France). We are studying IT engineering in our second year of RICM (Networks and Multimedia Communication). This project has been developed as part of the Software Engineering course, and the system project module provided by Didier Donsez (Joseph Fourier University, Grenoble). For our help, we have a tutor, Jacques Lemordant who is a researcher at INRIA (National Institute for Research in Computer Science and Control, Grenoble) and works on a project for disabled people.

Before beginning this project, we learnt about technologies such as C, Java and notion to JavaScript. We have studied assembly language, ARM code, depending on the operating system. However, we have no knowledge of PhoneGap development (html, java script), so this was a great chance for us to learn something new.

II) Requirements and problem statement

Today, itineraries are planned with GPS (outdoor) or done by hand with suitable map software (indoor). However, the first solution requires lot of energy, and the second is theoretical and takes a lot of time. Our project improves data creation in both cases.

For this, the system must provide solutions for the requirement stated below:

1. Provide a way to create an itinerary from a departure point.
2. The itinerary should be accurate (-50 cm / 20 inch)
3. Show the current orientation and modification
4. Record the user's information, such as points of interest. These records are audio and textual.
5. Backup the itineraries.
6. Export data in XML file compatible with OpenStreetMap for other uses, such as Navigation of Visually Impaired People.
7. Display itinerary on map.

III) Requirement specification

To begin with, we brainstormed different ideas. Then, we presented them and designed the main specifications with our tutor. The software had to be usable without specific skills in IT, and be used by anyone. The itinerary capture needed to be quick and accurate. The system has to be used in a public area (like stations) and pass unnoticed. It has to be useful and compact. So the system must be simple, accurate, easily transportable (public transport, plane) and discrete.

a. Technology research

In the beginning, we studied the different communication technologies available. They must respect certain features: low energy, short distance and reactivity. We have found three technologies which correspond to our needs:

- Bluetooth
- Zigbee
- ANT+

Our choice focused on ANT+ technology because it has a low consumption, a passive mode and it's a cheap technology. This technology works for 3 years with a simple coin-cell battery. It can send speed sensor information with a range of up to 30 meters. In comparison, the same battery with Bluetooth works for 7 days and has a range of up to 10-15 meters and 6 months for ZigBee with a range of up to 100 meters. Another benefit of ANT+ sensors are the possibility to connect 2^32 sensors in the same network.

b. Support

Support of the project must be mobile, discrete and lightweight. We are able to travel long and short distances easily. In addition, it must provide for attachment points for the sensors, and cheerful smartphone software. The smartphone should work with the different operating systems on the

market (Android, iOS, Blackberry, Windows Phone), and have an ANT + connectivity (native or with a dongle).

c. Use Case

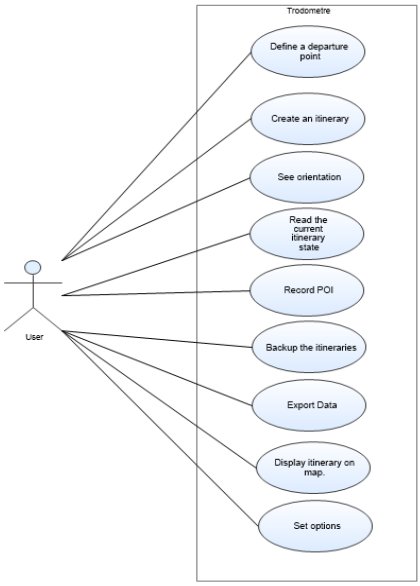


Figure 1 : Use case diagram

In *Figure 1* all the basic features we will make can be seen. We have an actor class, the user. The user represents the person who wants to interact with the system. This interaction is divided into different tasks that the user can carry out:

1. *Define a departure point*

This case corresponds to the definition of the starting point. Three different methods are possible: the first is to obtain the current position of the user, with a tracking system. That needs to be outdoors and have a tracking module on the smartphone. The second method is to ask the user. He selects a point on the map, and retrieves its value. This requires an access map (online or local). The third method asks the user for latitude and longitude. The selected point can be saved for repeated use.

2. *Create an itinerary*

This case corresponds to the recorded itinerary with the system. For this, the user needs a system and declares the point of departure (case 1). The user activates the record and navigates on the itinerary.

3. *See orientation*

The user can see the current system's orientation and its modification. The modifications appear when the user turns. The system must be in itinerary creation (case 2).

4. *Read the current itinerary state*

The user can see the different measures step. For example, the first distance after the first turn, the first POI emplacement… The system must be in itinerary creation (case 2).

5. *Record POI*

The user has the ability to save POI. To do this, after he has activated the function, it must name the POI. Then, optionally, the user can record a corresponding sound. This function allows you to add information about the route. For example, in a station, you can add POI to represent a lift, to warn the person following the route. The system must be in itinerary creation (case 2).

6. *Backup the itineraries :*

The user can save the current itinerary, and start a new measurement.

7. *Export Data*

The user can export data. This export can be done in two ways: the first is to save it on a storage device, such as an SD card. The second method is to send the data via email. For this method, the software requires the user to connect to the internet.

8. *Display an itinerary*

The user can display an itinerary in OpenStreetMap view. For this, the route should be saved, and selected by the user.

9. *Set option*

The user can set the system and improve the reliability. The user can modify the diameter of the wheel, and the number of sensors available on the wheel.

IV) No functional requirements

Our application should be compatible with a smartphone, doesn't use more battery and can be used by everybody. It can work without network and we should think about its improvement.

a. Interface

We have created some prototype of the interface. Our aim was to create an easy and complete interface. Firstly, we must start and stop a journey. Then, during the execution, the user must be able to pause his measurements. At any time, the user can add POI (Point of Interest) in his journey to indicate a change or information. We have added a button which requires the user to add a title with a voice recording. Moreover, we can see in real time the distance since the last action (turning or POI) or the global distance. At the beginning of a trip, a user may switch with another page where he can choose a starting point or wait for the GPS to find this point then cut off.

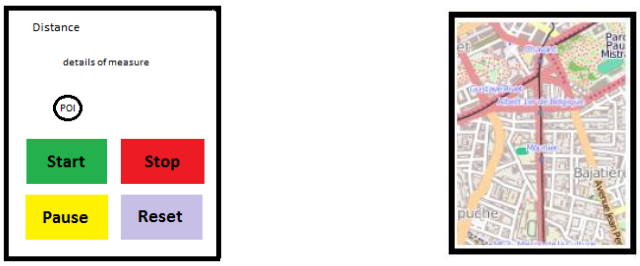


Figure 2 : draft interface

b. Reliability

Our system works even without connection to Internet. We can imagine it being used everywhere in the world. Thanks to this, it is possible to save measurements in a XML file in the smartphone. Another benefit is given by ANT+ sensor because these sensors record all actions even when offline and send them again when required. So, measurements are never forgotten.

c. Scalability

Firstly, all tests are carried out with one sensor and a single diameter. We leave the user the choice at of changing these values thanks to a menu and two options. We also hope to add compatibility with more sensors to improve measurements or obtain other information.

V) Management Plan

a. Team coordination

Coordination in the team was simple because there were only two of us. Because of this we did not appoint a project manager. As a first step, we decided together. Then, for important decisions, we validated our tutor. When the choices were fixed, we selected one person to be responsible for this choice. That meant having one person responsible per module.

b. Programming Management

We chose to work with an agile method, like scrum type, but adapted to two people. We decided to keep the prototype iterations very short (two-three days). In fact, the prototype was carried out in three weeks. Iterations were divided into three steps:

1) Choice of specifications and implementations of features during the iteration.
2) Production
3) Operating test implementations.

Thus, at the iteration end, we were sure that the functionality implemented meets the specifications defined in the beginning of the iteration. We could then move on to the next iteration.
For critical features, we decided to work together. We used the pair programming method. This technique allowed us to save time and avoid bugs. We mainly used it to develop the communication module, because we are not familiar with ANT+ technology.

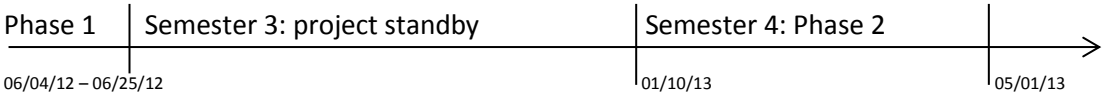
For the project, we decided to keep the same approach, but adapting the duration of iterations to the project, which is about 3 months. We decided to make iterations 2 to 3 weeks.

c. Managing project artifacts

We decided to develop this application with an IDE. Our choice is focused on Eclipse IDE because we knew this, and that plugin development for Google Android was available. Indeed, Eclipse manages SVN. In fact, the project files are located on an SVN server. We each have an access. The SVN service was Google code. Each commit was carried out once the functionality had been tested and functional. SVN enabled us to develop the same components at the same time. So, we had a local repository to develop the functionality, and carried out the test. In conflict cases (we had two conflicts), we carried out a merge in coordination. We have carried out 60 commits for the prototype.
For the project, we will continue to use SVN, and commit only once functionality has been tested and validated.

VI) Project plan

Our project is divided into two phases. The first is the prototype, and the second is the final project. These two phases, each corresponding to a project module of our first two years of engineering school. The distribution is as follows:



Phase 1: Prototype corresponds to first year project module (3 weeks).

Phase 2: Project realization to second year project module (4 months).

The third semester has no project module, so the project was on standby.

a. Phase 1

We can see on figure2 our start Gant diagram and the final one. Our tasks correspond to each of the iterations we decided on early in the prototype. The details of this were defined at the beginning of the iteration. Our Gant diagram allowed us to prioritize different tasks.

Tasks \ Iterations	1	2	3	4	5	6	7	8
Specification project								
Technologies research								
Android Formation								
Engine module								
IHM module								
Connection module								
Test and validation								
Bug resolve								
Documentation								

Tasks \ Iterations	1	2	3	4	5	6	7	8
Specification project								
Technology research								
Android Formation								
Engine module								
IHM module								
Connection Module								
Global Test and Validation								
Bug resolve								
Documentation								

Figure 3 : Initial Gant (right) and Final Gant (left). For the final Gant, the red cases correspond to the overshooting deadlines. An iteration corresponds to 2 days.

Tasks details:

- Specification project: Definition project features with our tutor.
- Technology research: Research on ANT+ and Android, and the hardware requirement for the prototype.
- Android training: We familiarized ourselves during this iteration with programming under Android. In particular we carried out program examples requiring access to the sensor manager, and the configuration of the rights of access.
- Engine module: Development of application engine.
- IHM module: interface development.
- Connection module : connection module development
- Test and validation: Tests in real conditions and module validation.
- Bug resolve : Correction of bugs which appeared during tests
- Documentation: writing documentation project.

Milestones:

We divided our project into milestones. Each one of these stages had one or two persons in charge. The person in charge was to check that we carried out the specifications of each module well, and was to check that we met the deadlines:

Id	Milestone Description	Responsible	Plan	Iteration +/-	Final
M001	Project Specification	JFB BT	1	0	2
M002	Technology research	JFB BT	2	0	2
M003	Android Training	JFB BT	3	1	4
M004	Engine module	JFB	4	1	6
M005	IHM module	BT	5	0	5
M006	Connexion Module	JFB	6	0	6
M007	Global Test and Validation	BT	7	0	7
M008	Bug resolve	JFB BT	8	0	8
M009	Documentation	JFB BT	4	0	8

Figure 4 : Milestones (with names in initials).

b. Phase 2

We have not yet completed this part of the project, we indicate only our initial Gant diagram:

Tasks \ Iterations	1	2	3	4	5	6	7	8
Specification project								
Technology research								
PhoneGap Formation								
Engine module : Implement management data								
Engine module : Implement export data								
Engine module : implement OSM interface								
IHM module : Global view								
IHM module : Option menu								
IHM module : OSM integration								
Global Tests								
Bug resolve								
Documentation								

Figure 5 : Preview Gant final project. An iteration corresponds to 2 weeks.

VII) Program design

a. Motivation and overview

We have divided our project in 4 parts. The code will be in Java Script to respect the Phone Gap requirement.

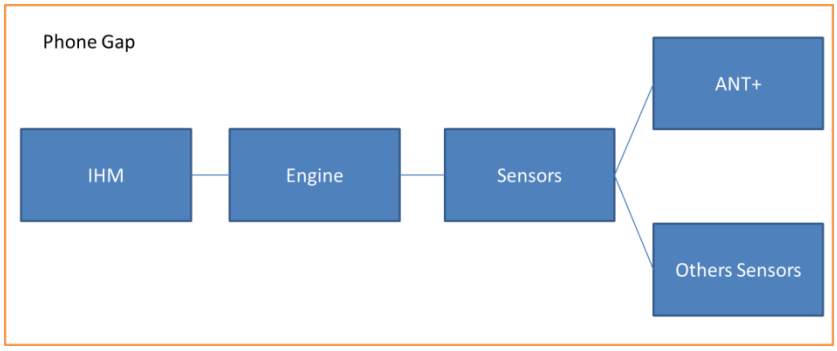


Figure 6 : general labor representation

- ✓ IHM class is composed of every element of interface user with interaction and option. User can control the application.
- ✓ Engine class is the core of program. It makes link between the user and hardware, mainly sensors. It performs remote calculations and processes POI.
- ✓ Sensor class associates ANT+ sensors with sensors included in the smartphone like gyroscope, accelerometer or compass. It obtains the data and transforms it to send it to the Engine class.

b. Class Diagram

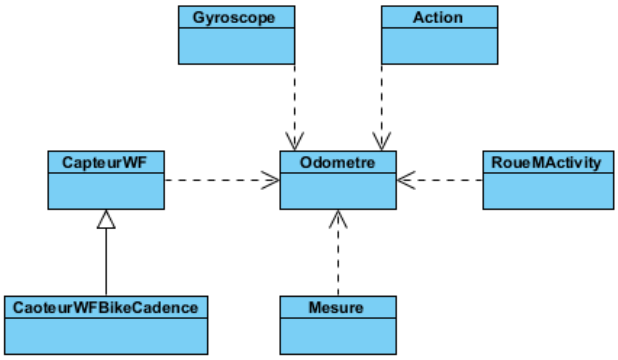


Figure 7 : Class Diagram

Detail of every class:

- Odometre: It's the main class. It controls interaction with sensors and interface. It sends a request for connection then obtains information and processes data before saving. It also gives the action to be done when the user presses a button.
- RoueMActivity: This is the class that gives the interface. It is the first class call at launch of application. It calls the Odometre class that initializes other parameters. It enables the creation of buttons and menus for user with the positions and dimensions.
- CapteurWF: This class controls ANT+ sensors. Its methods enable a sensor to be connected and disconnected and its parameters to be chosen.
- CapteurWFBikeCadence: This class inherits from CapteurWF. It retrieves information provided by the BikeCadence sensor.
- Gyroscope: This class initializes the gyroscope sensor. It processes information to give direction followed by user.
- Mesure: This class is made up of our data structure and contains a name of measurement, its distance and a list of different Actions performed.
- Action: This class defines an action, with its name and when it arrived, indicating the distance.

c. Sequence Diagram

Currently, our system enables us to perform some actions. We have set out these actions in this diagram:

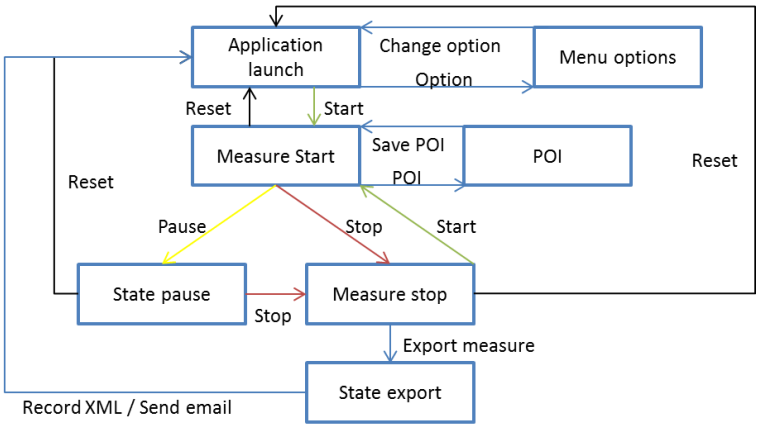


Figure 8 : Diagram states

States:

- Application launch: After launch application, we are in main interface and we can carry out our first measurement or change an option.
- Measurement: In this state, the application records all the information of a journey, sensors are initialized and give data.
- Pause state: This state enable new information to be locked and waits for the user to resume his journey.
- Stop state: When user stops his measurement, this state stores information in a structure and waits for export or a new measurement.
- Menu options: Thanks to this state, user can change the number of sensors and wheel diameter.
- Export state: In this state, the user has the choice between two types of export. He can send this data by email or save it in his smartphone.
- POI: A name is required of the user, and we propose his to record with his voice one additional information.

Actions:

- Start: When user wants to launch a new measurement
- Stop: At the end of measurement, when the user has finished, he presses the stop button and the measurement is considered to be finished.
- Pause: In a journey, a user can temporarily stop a measurement and take it up again. It's like Stop but the measurement isn't finished.
- Reset: at any time, the user can stop and restart the measurement. This action deletes all information at the last measurement.
- Export measurement: After a measurement, the user can decide to process the data. He can store it or send information.
- Press button option: Thanks to this button, he can change two types of options. According to its material, it can change
- Press button POI: When user wants to add information other than the distance at one point of the route.
- Change options: Modification of measurement data in the application.

- Save/Send an XML file: Action that the user wants carried out with data saved during the journey.

VIII) Implementation

We have made the choice to separation our implementation in two parts. In our first years, we have a project on three weeks and we have beginning to understand the project and realized a prototype. This year, we have a period longest for improve our system and make to compatible with every system thanks to Phone gap.

d. Prototype

In June, during 3 weeks we have beginning study a project and to implement an application. We have made this prototype for testing the feasibility of system and we have seen that it's possible. We have implement this application on Android because this system use Java code, a programming language that we know and Android have a good interface of development with Eclipse. Furthermore, we have Android material with ANT+ sensor indeed (Sony Xperia S).

We can see, we committed different files 60 time over our 3 weeks of work and we wrote 2000 lines of code separated into seven classes.

Finally, during this project we learned how the Android system works and how to access smartphone material, test the new ANT+ technology and study the correct distribution of tasks. Furthermore, we worked on a real, useful project and checked the good sense of this application. On the architecture side, the prototype will be the same for any new project ensuring compatibility with every mobile system.

e. Phone Gap

We decided to use PhoneGap because this framework is available on all smartphones on the market, and it’s an open source project. PhoneGap is based on web technologies such as HTML5, JavaScript, CSS. PhoneGap is made up of two layers:

- The presentation layer: HTML and CSS.
- The engine layer: a JavaScript library allowing access to the mobile hardware, GPS, camera, contacts, etc.

Our project is divided into 3 main parts. Each of them will use particular features of PhoneGap:

- IHM: this module uses the HTML and CSS feature.
- Engine: this module uses the JavaScript API. They include access to the various sensors, such as the gyroscope and the compass.
- Sensor: This module allows another extremely useful feature of PhoneGap to be used, the plugins. Indeed, to set up communication with the sensors, we will use an ANT +plugin, so we won't have to re - develop the management of communications between the phone and the sensor. In this way we will only have to develop management of the communication.

We will be also careful to separate the application engine and the interface. This is an important point for us, as it allows our interface to evolve, without having to recode engine if the project would be taken by another student group next year.

IX) Our prototype

Our application has to be suitable for a scooter so we have linked a sensor and a smartphone attachment (Figure 9). A scooter responds to all desired the characteristics. It is foldable and lightweight so easy to carry. It is a means of transport that goes unnoticed in public places. In addition, with its small size it can go anywhere.

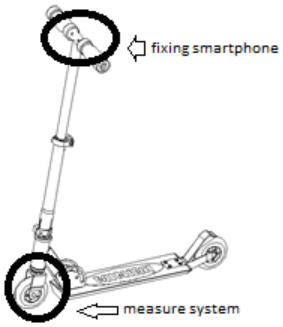


Figure 9 : Schema of Trodometre


Figure 10 : picture of measure system

On the wheel of scooter, there is a magnet which interacts with a Wahoo sensor which could calculate the number of turns (Figure 10). When the magnet passes the sensor, it gives us the number of passages incremented by one. The scooter's wheel is smaller than a bike wheel and provides good precision +/- 45 centimeters.

In the smartphone, the gyroscope allows us to know when the user changes direction. We can thus take into account the change in the route description.

X) Verification and validation

We carried out a large number of tests on the prototype, to validate the concept of our application, and be sure of its operation. The prototype tests were as follows:

Action	Expected result	Actual result	Verdict
Itinerary Creation	measures lunch	measures lunch	OK
See orientation	gyroscope launch and display current value	gyroscope launch and display current value	OK
Read the current itinerary state	Display itinerary details	Display itinerary details	OK
Record POI	Save POI name and sound	Save only POI name	FAILED, FIXED
XML creation	XML created and correct	TAG problem	FAILED, FIXED
Export Data SD	save the itinerary on SD Card	save the itinerary on SD Card	OK
Export Data Mail	save the itinerary and send by EMail	save the itinerary and send by EMail	OK
Set Option	edit correct	edit correct	OK

We tested our prototype in a Grenoble station, to verify in real conditions if the system responded correctly. As a result of this test, we used the latest iteration of development to fix any bugs. During the summer, our tutor used the prototype in Japan, for measurements in the Sugimotocho station:

Figure 11 : Students from the GISLab using the Android Kick-scooter at Sugimotocho, Osaka City University

These tests have allowed us to obtain an opinion on the system by people who had not participated in the development. These opinions were important, because they gave an objective opinion on the project concept, and the prototype. The results show that the concept is good. Users appreciated the Kick-scooter, and the method used to calculate the itinerary. The utilization is intuitive and easy. However, we got poor feedback about the XML format we chose. In fact, for real use, it isn't practical. For the final project, the users want a standard XML format. Our choices are focused on GPX, and OSM format.

For the final project, we plan to redo all prototype tests. We will add new features-related tests. These tests will be the following:

- Define a departure point, manually or with OpenStreeMap view.
- Display an itinerary on OpenStreetMap
- Many sensors on the well, to check the precision measures.
- Import the export file in Navigation of Visually Impaired People project

When we have done these tests, we will request new people who have not participated in the test application to give us returns, and improve the application in problems case. This step is important for us, because this project will then be used to complete routes for another person, and will be used in production.

XI) Summary

This project has enabled us to increase our software engineering knowledge. The fact that it was in two parts has enabled us to integrate it into our courses. We learned how to work as a team on an important project.

The prototype was very interesting, because it delivered a product in a very functional way. Communication was an important point. Indeed, each worked on a different project part, at the same time, and this was essential. Then we learned to use the development tool versions, like SVN. Now, we will adapt our program to better aggregate use. We hope to improve its quality and functionalities for a better experience. We have learned more at the end of the project and create an application that everybody way used.

We are currently carrying out the second part of this project, with final realization product. We hope that the use of PhoneGap will allow us to learn new knowledge about the new web technologies that are beginning to emerge as standard.

In human terms, this experience has allowed us to have contact with many types of people with different roles and responsibilities. We have seen how to design a software and customer interactions. The user point of view was also very beneficial because we realized that the user often had a different point of view from the developer.

XII) References

1. ANT+ - http://www.thisisant.com/
2. ANT+ - http://en.wikipedia.org/wiki/ANT%2B
3. Wahoo Fitness sensors - http://www.wahoofitness.com/Products/Wahoo-Fitness-Wahoo-Cycling-SpeedCadence-Sensor.asp
4. PhoneGap - http://phonegap.com/