Problem Statement Document

Purpose

The *Problem Statement* is the first handout given to the developers right after the project kickoff. It serves as a mutual understanding of the problems to be addressed. It is written by the client – often in collaboration or with the help of the instructor / project manager.

The problem statement describes the current situation, the requirements (functional and nonfunctional requirements) and the target environment in which the system will be deployed. It also defines the deliverables expected by the client, together with delivery dates and a set of acceptance criteria.

Audience

The audience for the *Problem Statement* includes the client, the project manager, and the developers who participate in the project.

Table of Contents

Document History

Rev.	Author	Date	Changes
0.1	Hazem		Base structure, 1st content and scenarios
0.2	Sacha	18.9.2015	Added info from previous lab

0.3	Olaf	20.9.2015	• Reworked problem description • Used the first scenario ideas for the formulated scenarios and added some. • Functional Requirements
0.4	Olaf	21.9.2015	• Non-Functional Requirements • ‘Refactoring’
0.5	Sacha	28.09.2015	• Minor updates (scenario 2, non-functional requirements), old content from iOS 2014 removed
0.6	Sacha	05.10.2015	• One scenario removed
0.7	Olaf	10.10.2015	• Refactoring based on feedback by Barbara • Removed ‘Driving Style’-Scenario, added address book scenario • Added usage of application name ‘in|Forming’

1. The Problem

Have you ever wondered what your mobile phone knows about you? What things you like or avoid? What personality you have?

Mobile phones are regarded differently. Some see them as a harmless and useful helper, others as a surveillance tool that exploits the information on users.
If a user is not sensible enough with personal information, it might be used to his disadvantage. However, a mobile phone is a useful device, so it makes sense to profit from its features.

The app **'in|Forming'**, which is going to be developed during the project, will address this problem. The name is derived from the three key points the app should address

- **Information**, because the app will collect information
- **Forming**, because the collected data will be formed and interpreted
- **Informing**, because the app will inform the user what information was won

in|Forming will collect data using a variety of sources available on a smart phone to create a detailed profile of the users behavior and personality. The sources of information may be GPS-coordinates, the address book, photos, the internet, whatever is within the limits of iOS. The data gets collected, enhanced and interpreted. The user will see the gained information and the derived assumptions.

in|Forming may be seen as a 'friendly Big-Brother'. It will try to get all available information about a user to present it to her. As a result, the user gains information about what other apps could know about him.

This will help the user to use the mobile phone responsible, to benefit from its advantages and avoid possible disadvantages.

2. Scenarios

Visionary Scenarios:

Scenario 1 (Detecting most important places):

Bob lives in a big city. He works 15 km away and usually takes public transportation to get there.
in|Forming recognizes that Bob spends a large amount of his time at those two places (home and office). From the time of day, it concludes which one is home, and which is the work place. An additional check of the addresses against known business entries identifies that he is employed by adesso.

Scenario 2 (Analyzing personal behavior):

Bob goes three times a week to the gym. The gym is 5 km from his home and he uses the bike to get there. The exercise usually takes 90 minutes.
When checking the address in|Forming might find out the place is a gym and make a conjunction between the duration and the location. The speed when riding the bike tells in|Forming, that Bob probably uses the bike.

Scenario 3 (Analyzing photos):

Bob meets with friends at the weekend and travels to the mountains. On the way he takes photos from the landscape and at the top of the mountain he takes a photo from his friends.
Photos contain GPS and time information. Even if in|Forming was not used during the trip, it can extract this information from the photos. Another way to gather information from the photos is face detection with checks against the contact list of the phone.

Scenario 4 (Analyzing address book):

Bob has a well-maintained address book of friends and relatives worldwide on his iPhone. It not only contains the names and telephone numbers, but also addresses and photos of his contacts.
By reading the addresses in|Forming gains the information that Bob has an international network with many connections.

3. Requirements

The goal of this project is to build the app 'in|Forming' for iPhone that integrates several sources to derive information on user behavior and personality and present it to the user in a tangible manner. The exact functionality of in|Forming will be determined during the agile project.
It is expected, that the participants are taking active part in creating the additional creative ideas that will result in further functional requirements during the project.
Those requirements will strongly depend on the restrictions given by the operating system to the apps.
Some basic functional requirements can be formulated.

- *Functional requirements*
    - The application must be able to learn from reoccurring situations. Result of this learning has to be an increased probability of assumptions done on the person using the application.
    - After a learning period, the application should be able to tell where the person lives and works with a high probability.
    - The application has to extract information from pictures with Geo-Coordinates to add this information to a personal movement profile.

  - The application should use information from all available sensors to gather additional information.
  - The application has to provide the gathered information and the derived assumptions to the user with references to the sources.
  - A backup functionality for the collected data and reports should be provided
- *Nonfunctional requirements*
  - Performance – the application has two parts. The first (referred to as 'Collector') gathers information and should be more or less invisible to the user. Maybe the user has to activate special functions. As a background task the 'Collector' must not consume a huge part of the devices resources.

    The second part shows the information the application won (referred to as 'Reporting'). This has to be provided without recognizable delays.
  - Memory leaks: to provide a stable application the implementation should avoid the occurrence of memory leaks
  - Data Integrity – it is not necessary, that every bit of collected data can be accessed directly. As the base of the data is unsure, data may be accumulated and refined. However, the data has always to be in a state to be displayed in the 'Reporting'.
  - User Interface: the application shall be fancy and easy-to use
    - Usability – 'Collector': In best case, it just has to be started and stopped.
    - 'Reporting': Display of the information and digging into details should be intuitive even for unexperienced users.
  - Scalability – The application should not infinitely build up data. By condensing data, the need on storage should be nearly constant after a certain size is reached. This will also help to keep access times down.

4. Target Environment

The application is a stand-alone application with no backend on a server.

- iOS 8 and later.
- iPhone devices only.
- Optional: iWatch (?)

5. Deliverables

- The applications
- Documentation
- Source code
- Testcases
- List of known issues
- Trailer (make sure not to use any copyrighted base material, like music or pictures)
- Project web page describing: Purpose, Team, Trailer, Screenshots

- (List of creative ideas for further enhancements)

6. Schedule

Thursday, 15.10.2015:	Kickoff Meeting
Thursday, 17.12.2015:	Design Review
Thursday, 21.01.2016:	Client Acceptance Presentation
January / February 2016:	Presentation at adesso

7. Client Acceptance Criteria

This is a broad problem statement, and as such, it is not expected that all functionality is delivered in one semester. Furthermore, the client should be willing to discuss changes of the requirements that might come up during the development. The delivery of the final the system should demonstrate the visionary scenarios described in the problem statement (or a subset to be selected by the client during the project reviews).
The team shall make sure not to use any copyrighted base material, like music or pictures.

8. Top-Level Design

This section contains an overview of the initial system design. This can be an informal description of the subsystem decomposition, if the system design is still in the early stages, or a UML drawing of the software architecture based on an architectural pattern, if the architecture is already decided upon. The subsystems of the top-level design are usually the seed for the team organization.

The application(s) can be designed on a greenfield. It's essential to use noncommercial frameworks or libraries except the apple development tools for iPad and iPhone.

9. Development Environment

This section describes the hardware and software environment to be used for the development of the system, in particular available tools, programming languages and software packages.
There are no restrictions concerning the development environment.