POLYTECHNIC UNIVERSITY OF MILAN

School of Industrial and Information Engineering

Computer Science and Engineering

Design and Implementation of Mobile Application Project: Safe Car Design Document

Course Professor: Prof. Luciano BARESI

Authors:

Mattia CRIPPA 1039725

Alberto PIROVANO 10396610

Academic Year 2017–2018

Contents

List of Figures

Chapter 1

Introduction

The *Design Document* is a document meant to provide documentation which will be used to help developers in implementing the entire system by providing a general description of the architecture and the design of the system to be built.

1.1 Purpose

The purpose of the Design Document is to provide a description of the system detailed enough to understand which are the components of the system, how they interact, which is their architecture and how they will be deployed. The level of the description is high enough for all the stakeholders to capture the information they need in order to decide whether the system meets their requirements or in order to begin the development work.

1.2 Scope

This document provides a detailed description of *Safe Car* software design and architectural choices. Every portion of the document is designed itself to be comprehensible, but a big picture of the system must be present to the reader in order to obtain the best knowledge on the matter when consulting this document.

1.3 Document Structure

This document is intended for individuals directly involved in the development of Safe Car application. This includes software developers, project consultants, and team managers. This document is not meant to be read sequentially; users are encouraged to jump to any section they find relevant. Below is a brief overview of each part of the document:

1. **Introduction:** This section gives general information about the Design Document of Safe Car application

2. **System Overview:** This section contains an overall view of the system, describing all the components that are part of the system and their interaction. This Section also contains constraints which are needed by the application in order to work correctly

3. **Architectural Design:** This section exposes in details the design chosen for the architecture of the system to be

4. **User Interface Design:** This section covers all of the details related to the behaviour, the graphical user interface (GUI) and the components of the application. Readers can view this section for a tentative glimpse of what the final product will look like.

Chapter 2

System Overview

Android is the first complete, open and free mobile platform. It is developed and supported by Google and this project uses a Google Android Mobile SDK (VERSIONE) for testing an application. The software development kit contains the emulator and advanced debugging tools to run and test the applications.
SafeCar is an application that has been designed to help the driver in improving its driving style. On one side it offers an *after trip* inspection of the data reports about past trips, and on the other side it provides a *during trip* hint generation engine to correct the driving style during the driving experience. The aim of our project is to develop an android application which can be used in the real world.
In order to use this application, the driver has to register as a user of the service. The *Registration and Login* procedures can be performed via the custom application functionality or via Google Plus APIs. At this point the application flow develops in two different ways: if the user's smartphone is not in the radio scan area of the Plug, he can only navigate data about his history trips or about his profile. If the user's smartphone, instead, is near the Plug, he can pair his device with that Plug and, after that, he can start a new trip; in the case that the user has already paired his device with his Plug and his smartphone is near it, the application automatically notifies the user he is being detected; this feature is called *Automatic Presence Detection*. Once the user has been detected, the application asks him if he is actually driving. This check is done in order to avoid the application to register the trip of a person

if he is not actually driving, for example is near the plug of a friend. If the user answers "Yes" the app understands that the trip is beginning and the *Driving Experience* actually begins. After this moment the driver will be provided with several hints about how to improve his driving style.

As previously told, the application is comprised of two main features:

1. **After Trip data presentation:** The user can inspect these data from a screen generated after the trip has finished. In this screen the user can see:
    - A GPS trip tracking of his movements during the trip, shown in a custom google maps widget
    - A general report about the just finished trip, along with a Driver Safety Index (DSI) that estimates the quality of the drive
2. **During Trip functionalities:** This functionality instead is mainly based on the concept building an engine to estimate a Driver Safety Index by using different data coming from a specific device. This device is a Plug smart object that has to be inserted in the custom car port and that, has the capability of sending cleaned and custom driving style data to a remote Cloud. This cloud is accessible via its custom APIs

Using all these data the engine builds a weighted sum of all the contributions given by all the data sources, called *Driver Safety Index* (DSI). This basic algorithm is easily extensible, because building a more complex one is only a matter of replacing the specific piece of code (a method) with another one that takes as input the same data and generates a same type index. In order to provide the user with an almost continuous feedback about his driving style, this engine recomputes the DSI index once every 30 seconds. Its computation is done in a dedicated separate thread that generates the current piece of advice to be sent to the user by simply switching on the value of the DSI. This hint is conveyed by filling a specific screen on the application.

System constraints:

These are the constraints which must be met in order to allow the application to work correctly:

- User of the mobile should have internet access
- User's device should support bluetooth
- User's device should support geo-localization

Chapter 3

Architectural Design

The System Architecture is a way to give the overall view of a system and to put it in relation to external systems. This allows the reader to have a more complete and general idea of the entire system and at the same time to have a deeper view of the principal components of the system itself.

Chapter 4

User Interface Design

Here are presented some mockups that represent an idea of the structure of the application pages.

4.1 Splash Screen and Registration & Login Pages

These mockups show an idea of the *Splash Screen* and *Registration & Login* pages of the application. The Registration and Login Page allow the user of the application to register as a user of the service and this can be performed via the custom application functionality or via external providers.

Figure 4.1: Splash Screen

Figure 4.2: Registration & Login

4.2 Home Page

These mockups show an idea of the *Home Page* of the application. The Home Page presents a *TabView* through which the user can navigate in order to inspect his history trips and a *Navigation Drawer Menu* through which the user can reach other pages of the application.

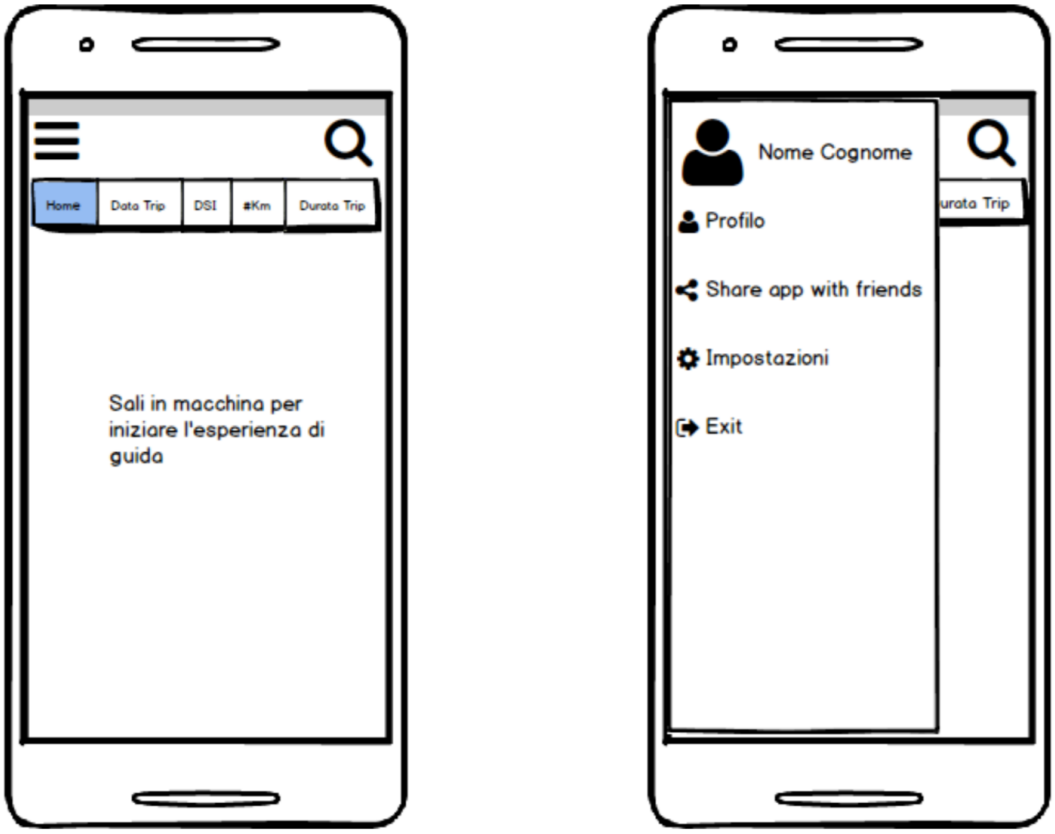

Figure 4.3: Home Page

4.3 During Trip Page

These mockups show an idea of the *During Trip Page* of the application. The During Trip Page presents a *View* where *Hints* produced by the application are loaded and viewable by the user. The user can interact with the application using two buttons: the first one is a *Pause/Resume* button which allow the user to pause the trip, without stopping it, if he wants to take a break from the driving session and, then, resume it; the second one, instead, is a *Stop* button which allow the user to end the driving experience.

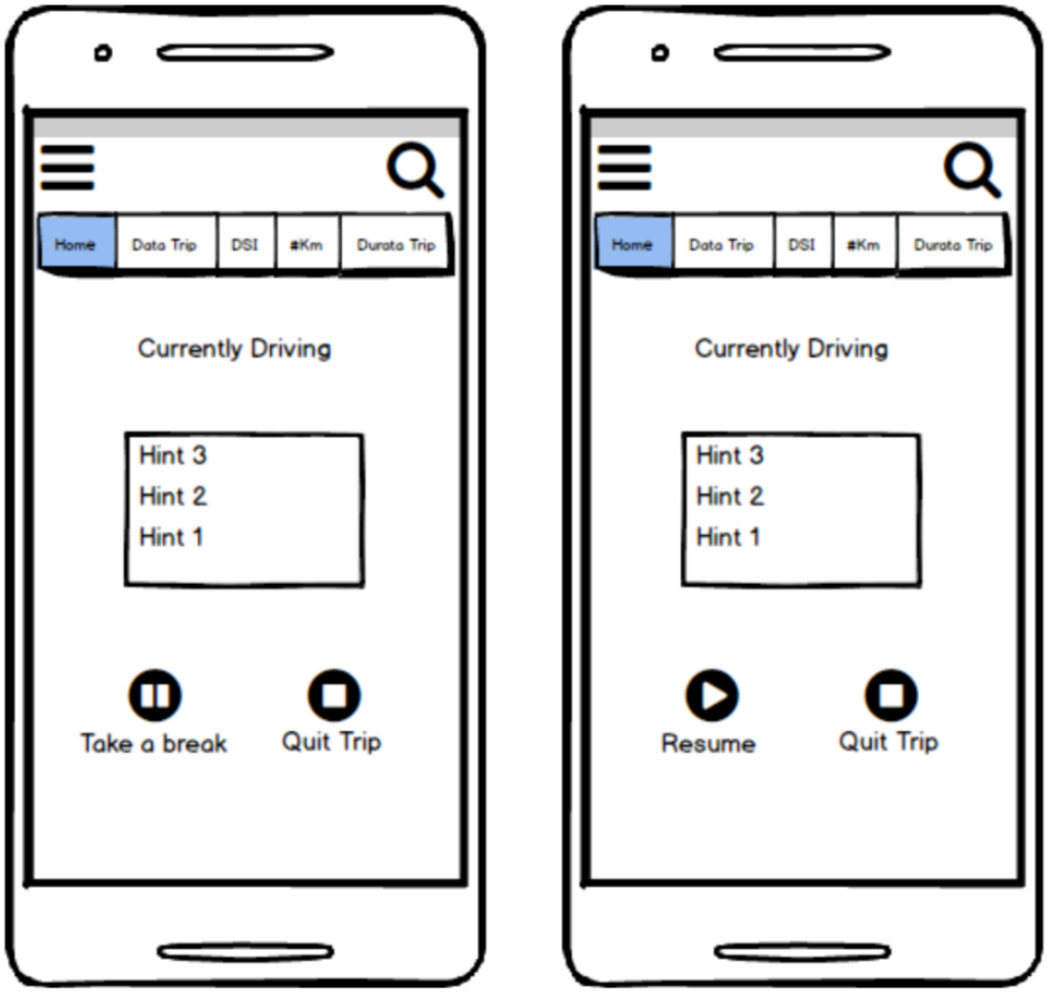

Figure 4.4: During Trip

4.4 Report Page

This mockup show an idea of the *Report Page* of the application. The Report Page is showed when the user inspect one of his history trips or when he complete a driving session pushing the *Stop* button. This page allow the user to see all the details about his trip including data like: the trip's date, duration and length, the DSI score and the path he made.

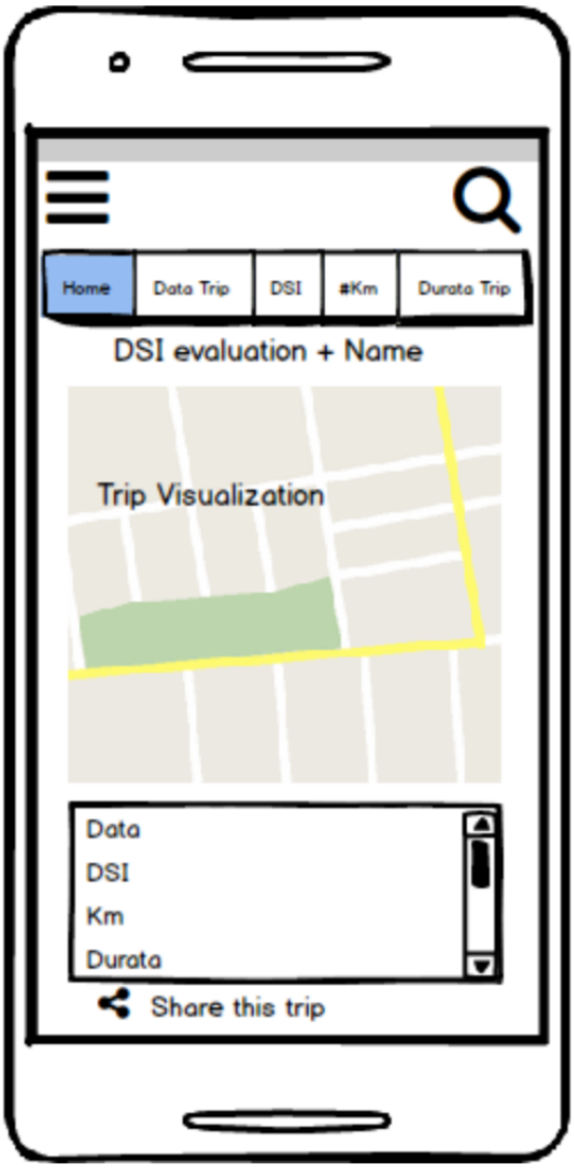

Figure 4.5: Report Page

4.5 Profile Page

These mockups show an idea of the *Profile Page* of the application. The Profile Page shows information about the user like his name, surname, email address, driver level and also some badges that he can unlock meeting specific conditions. Each badge is clickable and let the user know about its locked/unlocked status.

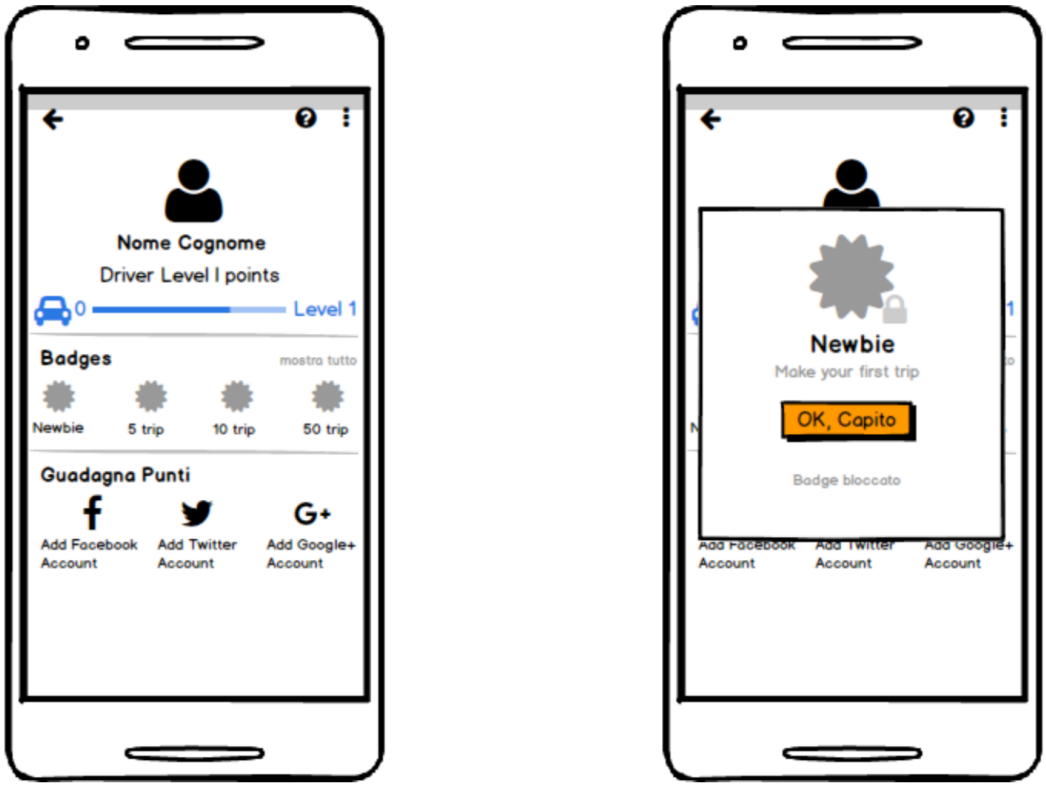

Figure 4.6: Profile Page

4.6 Settings Page

This mockup show an idea of the *Settings Page* of the application. The Settings Page allow the user to manage some settings of the application like push notifications and smart objects. This page also allow the user to see some info about the application and send feedback in order to improve it.

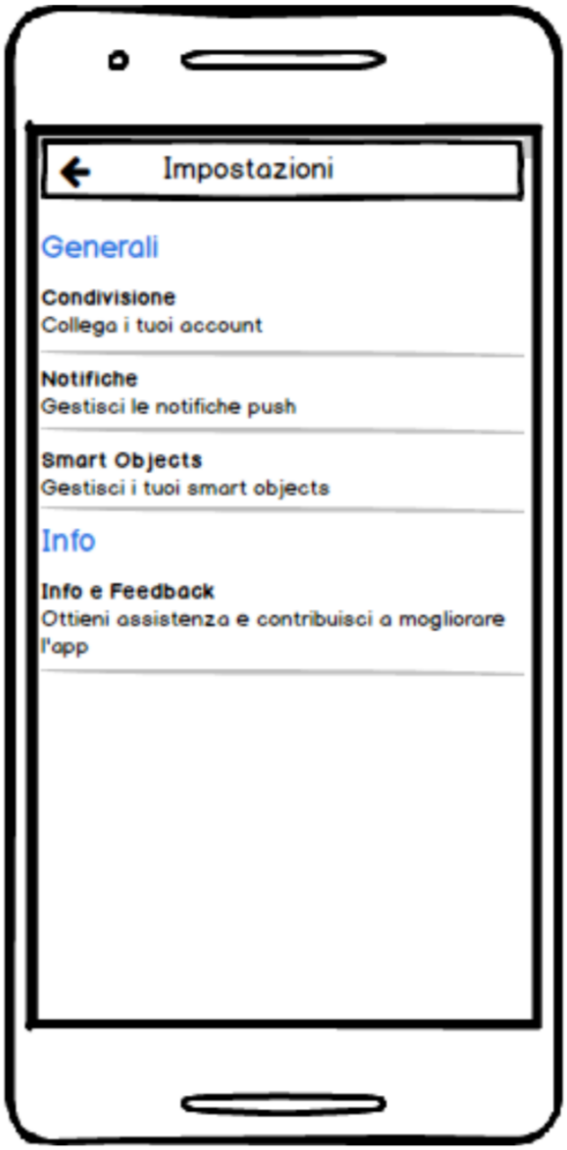

Figure 4.7: Settings Page

4.7 Smart Objects Page

These mockups show an idea of the *Smart Objects Pages* of the application. The Smart Objects Pages allow the user to manage smart objects which are necessary for the application. Using a bluetooth scanner the user can pair his device with a plug used to retrieve relevant information about the current driving session and he can also manage all the paired plugs, inspecting or deleting them.

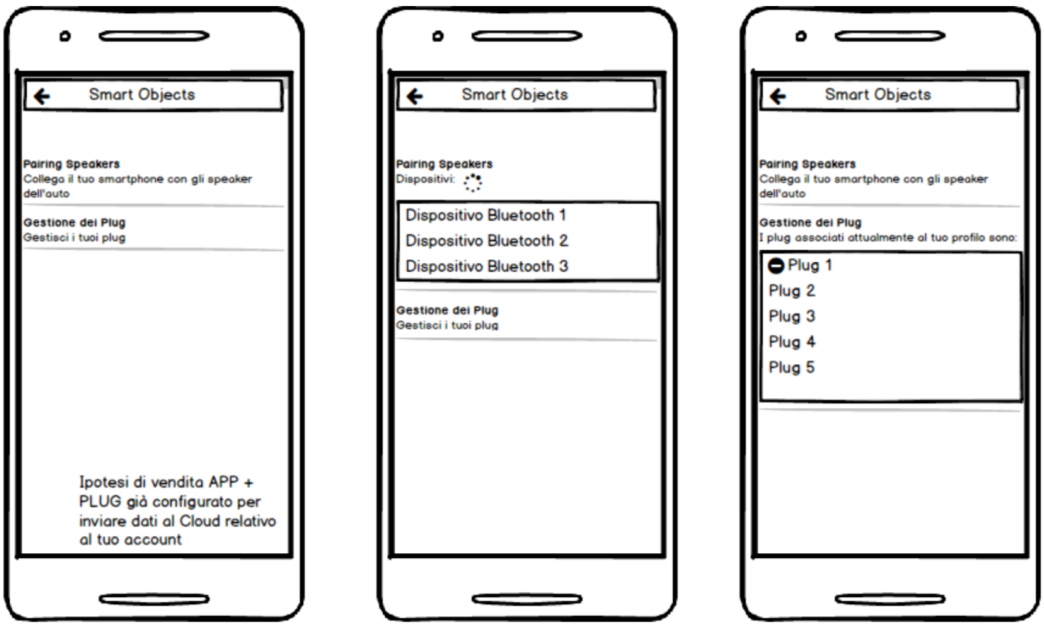

Figure 4.8: Smart Objects Pages