Scuola di Ingegneria Industriale e dell'Informazione
Dipartimento di Elettronica, Informazione e Bioingegneria

Design and Implementation of Mobile Applications

POLITECNICO
MILANO 1863

AlGa

Alessandro Falcetta
Gabriele Guelfi

Academic Year 2019–2020

Title: Design Document

Authors: Alessandro Falcetta, Gabriele Guelfi

Version: 1.0

Date: 10-May-2020

Download page: https://github.com/GabrieleGuelfi/Project_DIMA

Contents

Chapter 1

Introduction

1.1 Purpose

The aim of the present document is to describe the AlGa application with its services, a deeply definition of the main assumptions, the goals and a list of requirements, and the proposed solution. The definition of use cases and the scenarios will provide to highlight the features that the software has to offer to the customers and to better specify the boundaries system.

AlGa is an Android application developed within the course of *Design and Implementation of Mobile Applications* ad Politecnico di Milano, Italy. The goal of this course is to design a "mobile"applications by considering both the problem of designing the user experience, that is, the screens used to interact with the user, and the problem of understanding the actual distribution of the components that constitute the application and their interactions.

AlGa provides electric cars' owners with a simple way to find and use nearby charging stations. The main goal of our application is to help electric cars owner in the entire process of recharging their car. Being able to find charging stations tailored to everybody's needs is the key to meet that goal.

1.1.1 Goals

[**G1**] The application should provide the user with a **clear** overview of the nearest charging stations and their vendor.

[**G2**] The application should offer the user the possibility to order them, in a list, according to specific criteria: price, charging speed, distance, vendor.

[**G3**] The application should make it easy and straightforward to start the

navigation toward any charging station.

[**G4**] Users should be able to check their statistics about time spent charging their car, the total amount of money spent, etc. with the minimum interaction required; statistics can be enriched if the user indicate their owned car

[**G5**] The application should provide an easy way to save the profile of the user, with the possibility to log-in, log-out, delete the account and change personal information like e-mail and the owned car.

1.2 Scope

Electric cars are one of the most interesting technologies of the last years, with a possible bright future. Nonetheless, the public is still reluctant to invest money into this kind of products because of many concerns about the autonomy and the charging system. With respect to thermic engines cars, electric cars require more time to be recharged, have less autonomy in terms of distance and the infrastructure of recharging stations is yet to be completed.This is the context in which applications like AlGa can improve the experience of electric cars owners.

An easy and adaptive way find the best charging stations, according to everybody's needs, can be an important boost to the confidence in this technology.

1.3 Definitions, Acronyms, Abbreviations

Definitions

- **Health data:** some vital parameters of the users, for example heartbeat and blood pressure
- **Runner:** athlete who participate in a run
- **Fiscal code:** a code used in Italy that uniquely identifies every citizen
- **Third party:** a usr like an organization that is allowed to request health data
- **Individual request:** request of data of a single user

- **Group request:** request of data of a group of users, based on some parameters
- **118:** Emergency Medical Service, it's the number to call an ambulance
- **112:** Public-safety Answering Point

Acronyms

- **RASD:** Requirement Analysis and Specification Document
- **API:** Application Programming Interfaces
- **GPS:** Global Positioning System
- **HTTP:** HyperText Transfer Protocol
- **TLS:** Transport Layer Security
- **OS:** Operating System
- **SQL:** Structured Query Language

Abbreviations

- [**Gn**]: n-th goal
- [**Dn**]: n-th domain assumption
- [**Rn**]: n-th functional requirement

1.4 Document Structure

This design document is composed by 6 chapter: The document is organized as follows: Chapter 2 states what the app is about and which are the requirement is must satisfy. Chapter **??** explains the reasons behind the choice of the used technologies. Chapter 3 details the architectural choices for the entire application, while Chapter **??** gives an overview on the application from the point of view of the user, including some screenshots and mock-ups. Chapter 5 explains the use of external APIs, while some business-driven considerations are proposed in Chapter 7. Conclusions are finally drawn in Chapter 8.

Chapter 2

Idea and Requirements

2.1 Product perspective

A simple diagram of the common use of AlGa is here provided:

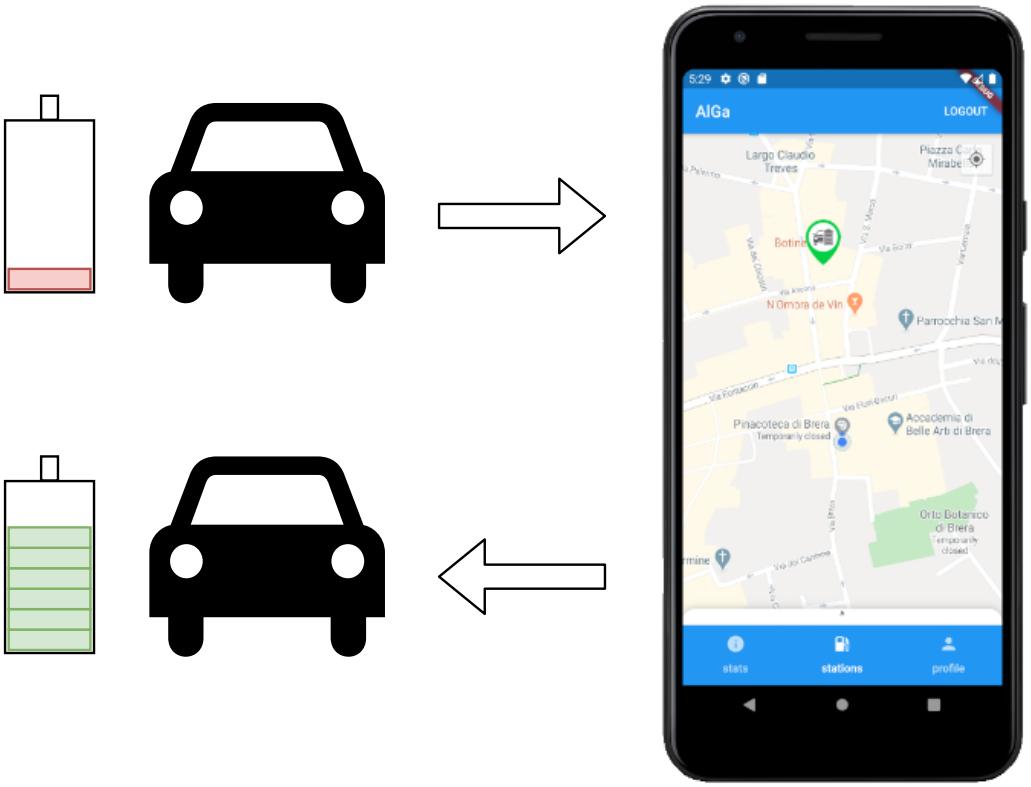

Figure 2.1: Use diagram of AlGa.

As we can see, AlGa is designed to help users directly on the road, in the fastest and easier way possible. Statistics useful for the users are another key part of the application: in fact, during the use, a set of useful metrics are collected and then displayed to the user. They can help her or him understand better the way in which they use their car, and fix some problems and/or wrong behaviors which may arise in the process of switching from a traditional car to an electric one. However, they are not strictly necessary to use the app; users can decide to opt-in to this functionality.

2.2 Product functions

According to the goals of the project, a more detailed description of the various functionalities is here provided:

Stations

This is the default screen. It shows a map, centered on user's position, together with the charging stations. The user can select every station to see its properties like cost, position, vendor, etc (Requirement 1). If the user decides to utilize that station, a simple click on the "GO" button will open the Google Maps application, with the destination already defined on that station (Requirement 3). Moreover, with a simple scroll menu, the user can also visualize a list of the nearest stations, with the possibility to order them by price, distance and speed and vendor (Requirement 2).

Stats

The statistics screen provides the user with data about the use of their car and of the application. That is, the amount of time spent at charging stations, the amount of money spent in energy, the distance traveled and other aggregate information (Requirement 4). Leveraging the possibilities offered by sensors installed on every smartphone, AlGa can collect these statistics in an effortless way for the user.

Profile

The profile screen lets the user customize their account on AlGa. They can choose a simple username, change their e-mail and password, and select their electric car. AlGa offers a list of cars from which the user can choose; every

car has some statistics about the consumption, the autonomy etc. This leads to more accurate usage data (Requirement 5).

2.3 User characteristics

AlGa User: An individual who has downloaded AlGa.

Registered User: An AlGa User who has created an account on AlGa platform.

Vendor: A company which offers electric recharging stations.

2.4 Assumptions, dependencies and constraints

Domain assumptions

[**D1**] Users can be identified through a couple email/password, unique for every user.

[**D2**] Users' devices can provide precise and correct data on location.

2.5 Use case diagrams

Name	Sign up
Actor	User
Entry conditions	User has AlGa app installed on their smartphone
Events flow	1. Click the "Sign Up" button 2. Fill the form providing the requested information 3. Click the "Confirm button" 4. User is now enrolled to AlGa.
Exit conditions	The app shows the main screen to the user.
Exceptions	1. The e-mail is already taken. 2. The format for e-mail or password is wrong. 3. All the exceptions are handled by notifiying the user and taking him back to the main screen.

(a)

Name	Login
Actor	User
Exit conditions	The app shows the main screen to the user.
Events flow	1. The user opens the app 2. The user fills the “E-mail” and “Password” fields 3. The user clicks on the “Login” button
sxit conditions	The app starts to show the collected data to the user
Exceptions	1. The e-mail is not associated with a registered user 2. The password is not correct for the given e-mail 3. All the exceptions are handled by notifiying the user and taking him back to the main screen.

(b)

Chapter 3

Architecture

The architecture for the application is composed by two main components: the Flutter application that represents the client-side and a server, hosted on Google Firebase, that represents the server-side.

In the diagram below the main components, that are described later, and their interaction are shown.

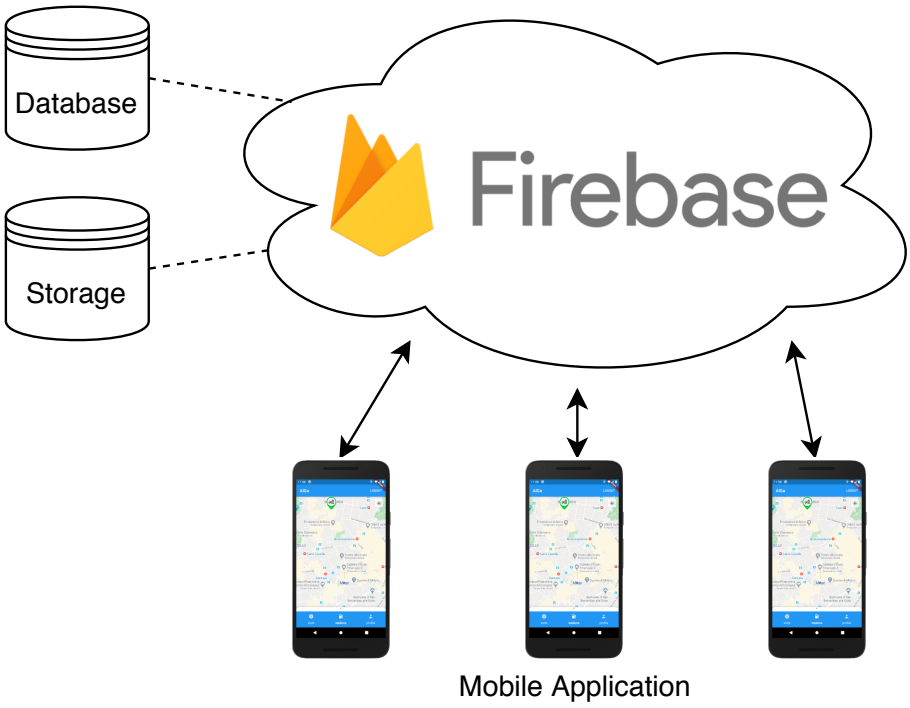

Figure 3.1: Basic architecture of the application.

3.1 Client-side application

This part is represented by the Flutter application and is most of the written code. The functional components and the graphical ones are not strictly separated, everything is built with the official Widgets provided by Flutter.

Widgets are elements of interaction in the graphical interface, they help to build UI, from basic to more complex one just combining them.

In the client-side there are also all the functionalities to interact with the Database and Google Maps, to handle notifications and collect statistics.

3.1.1 Code package

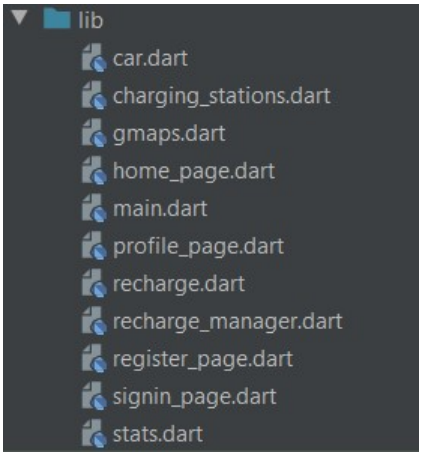

Figure 3.2: The lib package.

The figure above shows the lib package, where all the dart class are collected, basically each one represent a different page in the mobile application and it is the real core of the entire implementation.

The following list briefly describes the main classes:

- **main**: it is the entry point of the application that leads directly to the home page if the user is already logged in
- **signin_page and register_page**: they handle the authentication process(log in and registration)
- **home_page**: it is the core of the application and where the user will spend most of the time; it is divided in three tab; statistics, maps and profile

- **stats**: it shows and collect all the recharges of the user, showing interesting statistics about them
- **gmaps**: it shows the map with the charging stations, handling navigation and ordering of the stations
- **profile_page**: it shows the settings and information about the user, who can modify them here
- **recharge_manager**: it handles the recharging process calculating time and cost of it and the notification system when a recharge is finished
- **car, charging_stations, recharge**: they are Java-like classes that define the respective object

3.2 Server-side application

The server-side of the application is entirely hosted on Google Firebase and it is represented by Database and Storage, both of this components are fully integrated on Firebase and can be accessed and modified through the built-in console. This server aplication has many advantages: it offers a lot of services such as authentication, hosting and analytic, so it would be easy to add one of them in the future; as Flutter it is developed by Google so the APIs to made them work together are reliable; it is easy scalable starting from a free plan and then upgrading it.

Chapter 4

Design

AlGa is implemented in Flutter.

Chapter 5

External services and Libraries

We used many external services and libraries, some of them are useful to implement the core of the application, such as Firebase, while others just enhance the user experience.

Integrating these external services we didn't need to reimplement that functionalities by our own and since they are specialised in their field their efficiency is much higher.

5.1 Firebase

5.1.1 Authentication

This service of course represent the main entrance of the application since it handles the authentication process of users. It was very simple to implement because it requires just client-side code and it allows users to register or login with the classic email and password method or GitHub OAuth. The server-side of the application simply verify that the credential inserted are corrected and allows the user to log in.

5.1.2 Database

The Database provided by Firebase is of the NoSQL type and has several advantages: data can be updated directly in th Web Console or through a simple program, Flutter provides simple APIs to interact with it asynchronously and everything is stored in the Firebase's cloud.

We use it to store different type of cars, charging stations with their location and properties, users with their saved settings(i.e. name and car) and all the recharges made.

5.1.3 Storage

We use this service just to store the profile pictures of each users that eventually uploaded it.

5.2 Google Cloud Platform

5.2.1 Google Maps

It is probably the most important feature of this application, it allows to see and select the charging stations, visualised with a custom pin, check the own position and navigate on the map. Using Google Maps keeps the UI simple for the users because most of them already use it.

The navigation point-to-point is kept outside the application because the free plan doesn't permit that feature.

5.3 Minor Packages

- **image_picker**: it allows to upload images from the smartphone gallery, specifically for users that want to upload their profile pictures
- **email_validator**: it is used to validate the email inserted by a user during registration process
- **url_launcher**: it launches an external URL, we used it to launch the point-to-point navigation in Google Maps
- **sliding_up_panel**: we implemented a sliding up panel to visualise the charging stations by different order
- **flutter_local_notifications**: it is used to display push notification when a recharge is finished
- **background_locator**: it gets location updates even when the application is killed, it's useful to check if the user has arrived at the selected charging station

Chapter 6

Test Cases

AlGa has been tested using both emulators and real devices. Obviously, the use of an emulator can speed up the testing and debugging of the application; for example the possibility to change GPS location, battery, check notifications etc.

However, mobile applications are made to be used on real devices. Thus, is very important to check that everything works well on smartphones of different vendors, given the problem of fragmentation which afflicts the Android environment. In particular, we used Sony and Xiaomi devices, with Android 9 and 10.

These are the use cases tested for correctness.

6.1 Login and registration

Name	Registration
Entry conditions	User has AlGa app installed on their smartphone and it is not already registered
Events flow	1. Open the application 2. Click on the Register button 3. Fill the Email, username, password form and choose a Car 4. Click on ”Submit”
Exit conditions	The app shows the main screen to the user and the user is enrolled in the system.
Notes	The app correctly checks for the validity of the various fields of the registration

(a)

Name	Sign in
Entry conditions	User has AlGa app installed on their smartphone and it is already registered
Events flow	1. Open the application 2. Fill the email and password forms 3. Click the “Sign In” button
Exit conditions	The user is logged-in with their credentials
Notes	The sign-in is rejected if the credentials are not correct

(b)

Name	Sign in with GitHub
Entry conditions	User has a GitHub account and has AlGa installed on their phone
Events flow	1. Open the application 2. Click on the ”Login with GitHub” button 3. Login into GitHub in the browser page which is opened 4. The user is redirected into AlGa
Exit conditions	User is logged-in with their GitHub account
Notes	None

(c)

Name	Logout
Entry conditions	The user is logged-in
Events flow	1. Press on the Logout button in any application screen
Exit conditions	The user is logged-out and the app accepts a new user
Notes	None

(d)

6.2 Stats page

Name	Add a recharge
Entry conditions	User is logged-in
Events flow	1. Click on the "Add a recharge button" 2. Choose a date and time with the selector 3. Fill the "Cash spent" and "kw recharged" forms 4. Click on the tick
Exit conditions	The recharge is correctly added and immediately shown to the user
Notes	The app checks for the validity and range of Cash spent and kw recharged

(e)

Name	Modify a recharge
Entry conditions	The user is logged-in and has already done/added a recharge
Events flow	1. Click on the modify button near a station 2. Fill the Cash spent and kW recharged forms 3. Click on the tick
Exit conditions	The recharge is updated
Notes	The values are checked for validity and range. The initial values are the original ones.

(f)

Name	Delete a recharge
Entry conditions	The user is logged-in and has already done/added a recharge
Events flow	1. Click on the delete button near a station
Exit conditions	The recharge is deleted and the updated list is shown to the user
Notes	None

(g)

Name	Stats granularity
Entry conditions	The user is logged-in and has already done/added a recharge
Events flow	1. Click on the Week/Month/Year selector under "Statistics"
Exit conditions	The statistics are updated taking into consideration only the recharges in the range selected
Notes	

(h)

6.3 Stations

Name	Explore the map
Entry conditions	The user is logged-in
Events flow	1. Click on the stations tab 2. Move around the map
Exit conditions	The map shows the recharge stations and the other elements of Google Maps
Notes	The stations marked in Green are available, the ones marked in Red are not

(i)

Name	See a station's details
Entry conditions	The user is logged-in
Events flow	1. Click on the stations tab 2. Click on a recharge station
Exit conditions	A window is shown with the details of the station
Notes	None

(j)

Name	Navigation toward a station
Entry conditions	The user is looking at a station's detail
Events flow	1. Click on a station in the map 2. Click on the GO button in the details window
Exit conditions	A Google Maps link is launched and AlGa starts monitoring the user's position
Notes	None

(k)

Name	Order the stations
Entry conditions	The user is logged-in
Events flow	1. Click on the stations tab 2. Swipe up the stations panel
Exit conditions	The stations are shown ordered by the characteristic chosen (price, speed, distance)
Notes	Stations are colored in red/green according to their availability

(l)

Name	Locate a station from the list
Entry conditions	The user has opened the stations panel
Events flow	1. Click on the geolocate button near a station
Exit conditions	The panel is closed and the map is centered on the selected station
Notes	None

(m)

Name	Launch the navigation from the stations panel
Entry conditions	The user has opened the station panel
Events flow	1. Click on the GO button near a station
Exit conditions	A Google Maps link is launched
Notes	None

(n)

Name	Start a recharge
Entry conditions	The user has launched a navigation toward a station and the user is near the station
Events flow	1. A notification is shown 2. Click on the notification 3. Fill the Current battery percentage form 4. Click on the tick
Exit conditions	The recharge is saved into the database and a notification is scheduled for the end time
Notes	The notification for the end of the recharge is shown either with AlGa open or closed. The endtime is correctly calculated based on the user's car values

(o)

6.4 Profile

Name	Check the profile
Entry conditions	The user is logged-in
Events flow	1. Click on the profile tab
Exit conditions	The user's details are shown
Notes	None

(p)

Name	Change profile pic
Entry conditions	The user is logged-in
Events flow	1. Click on the profile tab 2. Click on the change picture button 3. Select an image from the selector
Exit conditions	The profile pic is updated
Notes	None

(q)

Name	Change the username
Entry conditions	The user is logged-in
Events flow	1. Click on the profile tab 2. Click on the modify button near the username 3. Fill the username form 4. Click on the tick button
Exit conditions	The username is updated
Notes	The new username's length is checked

(r)

Name	Modify user's car
Entry conditions	User is logged-in
Events flow	1. Click on the profile tab 2. Click on the car selector 3. Choose a new car
Exit conditions	The car and its values for battery and range are updated
Notes	None

(s)

Name	Modify car values
Entry conditions	User is logged-in
Events flow	1. Click on the modify button near the battery/range values 2. Fill the form 3. Click on the tick
Exit conditions	The values are updated
Notes	The values are checked for validity

(t)

Name	Modify e-mail
Entry conditions	The user is logged-in
Events flow	1. Click on the modify button near the email 2. Fill the form with a new email and the current password 3. Click on the tick
Exit conditions	The user's email is updated
Notes	The email is checked for validity; the password is checked; this test case is not available in case of GitHub login

(u)

Name	Modify password
Entry conditions	The user is logged-in
Events flow	1. Click on the Edit password button 2. Fill the form with the current and new password 3. Click on the tick
Exit conditions	The user's password is updated
Notes	The password is checked for validity; the current password is checked; this test case is not available in case of GitHub login

(v)

Name	Delete account
Entry conditions	The user is logged-in
Events flow	1. Click on the Delete account button 2. Fill the form with the current password 3. Click on the tick
Exit conditions	The user's profile is deleted and the user is logged-out
Notes	The current password is checked

(w)

Chapter 7

Cost estimation

Our plans made sure to keep into account all possible costs that our team will incur during development. As such we planned to recruit new team members only when needed in order to keep costs as low as possible. Our first research phase won't be very expensive and it will be mainly led by our core team. The main source of cost will be the recruitment of new team members as well as the planning and launching of the marketing campaign in order to ensure a successful launch, which we think is paramount to the long term success of the entire project. Technology wise the infrastructure won't have a great cost in the development and beta phase and we plan to scale according to the demand, in such a way we can be sure to avoid scaling over the demanded size and use the budget for possible extensions of the product or improving upon the existing services offered.

Role	Number	Estimated Cost (monthly)
Core Team Member/Chief	4	2,000€
Senior Project Manager	1	2,000€
Junior Back-End Developer	2 to 3	1,500€
Junior Front-End	1 to 2	1,500€
Security Analyst	0 to 1	2,000€
Junior Graphic Designer	1 to 2	1,200€
Marketing Supervisor	1	3,000€
Social Media Manager	1	1,500€
Digital Marketing Manager	1	1,500€

Chapter 8

Conclusions

Name	Alessandro Falcetta
Effort spent	20 hours
Task	• Group work • Introduction • Deployment view • Runtime view

Name	Gabriele Guelfi
Effort spent	20 hours
Task	• Group work • Overview • Component interfaces • Requirements traceability • UML