POLITECNICO
MILANO 1863

Design and Implementation of Mobile Applications - Polaris project VINCENZO MANTO, ROBERT MEDVEDEC

Design Document

Deliverable:	DD
Title:	Design Document
Authors:	VINCENZO MANTO, ROBERT MEDVEDEC
Version:	0.1
Date:	28-12-2021
Download page:	https://github.com/robertodavinci/android-dev-travel-app/tree/main
Copyright:	Copyright © 2021, VINCENZO MANTO & ROBERT MEDVEDEC – All rights reserved

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

This document provides a detailed description, mainly of the architecture and the UI, of the 'Polaris' mobile application.
'Polaris' application is a system used to enhance travelling experience through idea sharing among people, exploration of other users' travels, documentation and easy manipulation of trip destinations and ideas. The system itself is made so the user has every single information regarding his travel in one place, without having to remember or worry about any details.
'Polaris' is mainly a mobile application, with a possibility of being also expanded as a web application in the future.

1.2 Scope

'Polaris' is an application that helps all travellers around the world to easily and efficiently manage their travels and thus enhance their travelling experience. The target audience for this application is everyone who uses a smartphone and has at least some experience in using mobile applications, as some of the patterns and application usages might not be intended for the novices in the field. Thus the target audience ranges from 15 to 50 years old, although there are no strict boundaries.

The application is mainly intended to be used when a user is planning and making their own trip and has a liberty to organize their free time and places to visit.

Another important usage of the application is sharing trip ideas with friends and other users around the world. Planning trips by itself is a daunting and time consuming task, and with the lack of adequate applications on the market, we wanted to create something that is going to allow users to easily share and modify their previous trips and therefore improve the overall travelling experience for others.
The most important functions of the application are arranging a trip, finding points of interest in the area, organizing visits to those points, exploring accommodation and restaurants in the area, and crafting your own views of the travel by providing additional comments on the whole experience.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **Application**: a computer (mobile) program that is designed for a particular purpose.
- **Smartphone**: a mobile phone that performs many of the functions of a computer, typically having a touchscreen interface, internet access, and an operating system capable of running downloaded apps.
- **Google Maps**: a web mapping service developed by Google, used both as a standalone app and as an integrated mapping solution in most of the apps.
- **iOS**: operating system developed by Apple, used by their portable devices like iPads and iPhones.
- **Android**: most popular operating system for smartphones and tablets, developed by Google and partners.
- **Backend**: the part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data.

1.3.2 Acronyms

- **API**: Application programming interface, computing interface which defines interactions between multiple software intermediaries
- **UI**: User interface
- **GUI**: Graphical user interface
- **DB**: Database
- **REST**: Representational state transfer - software architectural style used in web services

1.3.3 Abbreviations

- **App**: Application.

1.4 Revision History

- **Version 0.1**: First .tex document created and added all together; 28th December 2021

1.5 Reference Documents

- nothing

2 Overall Description

2.1 Product perspective

2.1.1 Internal structure

2.1.2 Scenarios

2.2 Product functions

Functions of the system provide easy and intuitive ways to use the app. They are somewhat connected and have overlapping features. Nevertheless, the users may use only certain parts of the system and still get the full functionality they need from the app. These functions are mentioned in several places in the document, but their most thorough explanation can be found here.

2.2.1 Adding a trip

Adding a trip is the most essential part of the system. The function features several parameters and allows for high level of customizability in order to create a fully unique travel experience.
Each point of interest is featured as a specific "destination", although even places that are not regarded as specific destinations can be inserted into a trip. Destination fetching is done through the Google Maps API, with the users also having the ability to add and edit their own destinations. The function features the following:

- Selecting a starting point, that is then connected to the nearest city on the map
- Adding other destinations to the trip
- Writing a trip description
- Adding preferred way of travel between destinations
- Identifying trip price level
- Adding comments to trip destinations
- .....

2.2.2 Editing a trip

Editing a trip can be done on two different types of trips - public or private. If the trip is public, either published by the user editing it or someone else, by starting to editing an exact copy of that trip is created, which can then be published again under different name and different trip ID.
If the trip is private and has not yet been published, then a new trip is not created, but rather the trip ID stays the same. If the trip is then published and edited again, the new edited version of the trip has a new trip ID and is a whole new entity.
Editing a trip features all of the same functions that adding a trip does, which means that everything from a small comment to the whole trip can be changed.

2.2.3 Finding a trip

Finding a trip can be done in several different ways. The first way features a search bar which then searches a trip by the starting point name or by the city which is the closest to that destination. The second way is a search by the trip ID or a hyperlink, which can be directly received from other users. The third way is by accessing a specific user's account page and scrolling through their published trips. Finally, trips can also be found by looking at the interactive map, selecting the area of the desired starting point of the trip, and finding the trip through the distinct trip name and photo.

2.2.4 Following a trip

This function mainly uses a specific trip and sets it as an active trip of the user. This makes it easily accessible by the user at all times by using the bottom navigation bar, and allows him to follow certain steps of the trip without losing progress. This function also allows the user to change the current active trip and still keep the progress of an old trip, so that the progress can be easily restored when that trip is again set as an active trip.

2.2.5 Updating account settings

Only a few settings can be changed in the user account. List is the following:

- Changing a username
- Changing a profile picture
- Switching between dark and light colour scheme
- Setting preferred price level
- Switching between offline mode and online mode

2.2.6 Offline function

The system allows the users to be disconnected from the Internet and still use the app fully. Individual trips can be downloaded and store in the phone internal storage, and then accessed at any time. If any changes are made to the trip during the offline time, a new iteration of the trip gets created and published as soon as the connection is restored and the user has come back to the online mode. Multiple trips can be downloaded and kept in the storage of the phone at all times.

3 Architectural Design

3.1 External Interface Requirements

3.1.1 User Interfaces

The CLup application interface will have to serve two types of users: customers and store managers. Opening screen of the application allows the user to pick a store or to login as a store manager. Depending on the choice, another screen is presented. For a customer, a screen with the options to retrieve a ticket, book a visit or change the store. For a store manager, a screen containing their camera view, and buttons to confirm the scanned ticket, notify the system about a customer exit and to log out of the application.

3.1.2 Hardware Interfaces

Depending on the current user the CLup application will require access to some hardware interfaces. If the current user is a customer, the application will require the device's camera and if the current user is the store manager the GPS location data will be needed. The application will require no further hardware interfaces.

3.1.3 Software Interfaces

The CLup application will not require any specific software interfaces.

3.1.4 Communication Interfaces

The most important communication will occur between the device and the database. The decision on the specific communication interface which will be used depends on the database, and is, therefore, left to the developers.

3.2 Functional Requirements

G1 Allow a User to "line up"/retrieve a number.

G1.1 Allow a User to retrieve a number through the application.

R1 The user must be able to select a specific store in which they want to do the shopping.

R2 The user must be able to request a number and a ticket.

R3 The user must be able to receive a number and a ticket.

G1.2 Allow a User to retrieve a number physically from the printer.

R4 The user must be able to physically retrieve a ticket from the printer containing a number and a QR code.

D8 The user has internet connection for the device at all times.

G2 Allow a Store Manager to control the entrance of a User via QR code scanning.

R5 The store manager must be able to scan a QR code.

R6 The store manager must be informed by the application if a user tries to enter the store out of order.

R7 The store manager must be informed when the capacity of the store is full.

R8 The store manager must be able to alert the system whenever a customer exits the store.

D6 The system can correctly save data about enter and exit times of anonymous customers, in order to calculate estimated wait time.

D8 The user has internet connection for the device at all times.

G3 Allow a User to get precise calculations of the wait time.

R9 Allow the user to receive a precise estimation of wait time when retrieving a number.

R10 The system must provide the user with an estimation of wait time based on data.

D4 The system can use data about the registered user to calculate estimated wait time.

D6 The system can correctly save data about enter and exit times of anonymous customers, in order to calculate estimated wait time.

D8 The user has internet connection for the device at all times.

G4 Allow a User to get updates/notifications on the estimated wait time.

R11 The system must be able to update its estimated wait time in real time.

R12 The system must be able to send an update to the user in specific intervals regarding estimated wait time until it's their turn.

D4 The system can use data about the registered user to calculate estimated wait time.

D6 The system can correctly save data about enter and exit times of anonymous customers, in order to calculate estimated wait time.

D8 The user has internet connection for the device at all times.

G5 Allow a User to "book a visit" to the store.

G5.1 Allow a User to "book a visit" to the store without indicating the expected duration of the visit.

R13 The user must be able to request to see all the available timeslots in that specific store.

R14 The system must be able to provide the user with the list of all available timeslots upon the request.

R15 The user must be able to select a specific timeslot.

R16 The user must be able to receive a confirmation of his timeslot reservation, along with a number and a ticket.

R17 Allow the user to be at most five minutes late for his reservation before canceling his ticket.

G5.2 Allow a User to "book a visit" to the store with indicating the expected duration of the visit.

R18 The user must be able to specify expected duration of his visit to the store.

D3 The user's device provides accurate GPS information.

D7 The system can correctly save data to and pull data from available time slot schema in the database.

D8 The user has internet connection for the device at all times.

G6 Allow a Store Manager to login to his store manager account with credentials.

R19 The store manager must be provided with the login credentials upon request to the system administrator.

D1 The store manager's username must be unique.

D2 The store manager's password must be secure.

Requirement, [Rn]	Goals, [Gn]	Domains, [Dn]
R1	G1.1	D8
R2	G1.1	D8
R3	G1.1	D8
R4	G1.2	D8
R5	G2	D6, D8
R6	G2	D6, D8
R7	G2	D6, D8
R8	G2	D6, D8
R9	G3	D4, D6, D8
R10	G3	D4, D6, D8
R11	G4	D4, D6, D8
R12	G4	D4, D6, D8
R13	G5.1	D3, D7, D8
R14	G5.1	D3, D7, D8
R15	G5.1	D3, D7, D8
R16	G5.1	D3, D7, D8
R17	G5.1	D3, D7, D8
R18	G5.2	D3, D7, D8
R19	G6	D1, D2

Table 1: Mapping table

3.3 Performance Requirements

4 User Interface Design

Provide here information about how much effort each group member spent in working at this document. We would appreciate details here.

