



Design Document

Design and Implementation of Mobile Applications

Valentina Politi – Martino Manzolini

A.A. 2020/2021 - Prof. Luciano Baresi

Contents

1 Document Structure	3
2 Introduction	3
2.1 Purpose.....	3
2.2 Scope 3	
2.3 Definition, Acronyms, Abbreviations	4
2.3.2 Acronyms used	4
2.4 Domain Assumptions.....	4
2.5 Functional Requirements.....	4
3 Use Cases Analysis.....	6
4 Architectural Design	12
4.1 Front End: Mobile Application.....	12
4.2 Back End: Firebase.....	13
4.3 Libraries used.....	13
5 Third Party interaction.....	14
6 User Interface Design.....	15
6.1 Colors and design	15
6.2 List of User Screens	15
6.3 List of Restaurateur Screens.....	16
7 Implementation and Integration.....	16
7.1 Front End implementation	16

7.2	Back End implementation.....	17
7.3	Front End and Back End integration.....	17
8	Testing.....	17
9	Software System Attributes	17
9.1	Reliability.....	17
9.2	Availability	17
9.3	Security.....	17
9.4	Compatibility.....	17
9.5	Scalability.....	18
9.6	Future developements.....	18

1 Document Structure

- **Introduction:** This section introduces the Design Document. It explains the Purpose, the Scope and the conventions of the document.
- **Architectural Design:** This section describes the components used for the system and the relations between them, providing information about their deployment and how they work. It also specifies the architectural styles and the design patterns chosen to design the system.
- **Third Party interaction:** This section introduces the interaction of APPetito with the listed Third Party APIs and services and their functionalities.
- **User Interface Design:** This section provides an overview on how the User Interface will look like. This section will be accurate enough to explain all our decisions about the design and the UI of the Mobile Application.
- **Implementation and Integration:** This section contains the order of the system's subcomponents implementation and their integration.
- **Testing:** This section describes the test cases submitted to the Application.

2 Introduction

2.1 Purpose

This document is the Design Document (DD) of the mobile application called APPetito. The objective is the one to provide general overview of the architecture of the software created, the interaction between all its different components and the way they function in order to pursue the requirements and the goals for which they were created.

2.2 Scope

APPetito is a native application for the offering or purchase of restaurant's meals depending on the role that the user will choose. As a restaurateur will be possible to open a profile for a restaurant, present a menu and manage incoming reservations. As client, instead, will be possible to browse with filters for different restaurants subscribed to the app, consult menus and complete reservations. A serie of missions, depending on the category of customer chosen, will allow the earning of points based on the reservations completed, to spend in discounts. The social objective of the app is also to promote real life moments to spend at restaurants, by inviting friends to subscribe, potentially increasing the incomes of the restaurants subscribed too.

2.3 Definition, Acronyms, Abbreviations

2.3.1 Defntitions

- User: a client customer of the service, whom can choose to belong to one of the following categories : Influencer or Adventurer
- Restaurateur : client of the service whom decides to advertise a restaurant using the application
- User Device: any compatible device with the APPetito application, mainly smartphones and tablets.
- App: abbreviation for the APPetito Mobile Application.

2.3.2 Acronyms used

- DD: Design Document.
- API: Application Programming Interface.

2.4 Domain Assumptions

- [D1] : Users' devices always have an internet connection available during the interaction with the system
- [D2] : User's devices have to support Mobile application
- [D3] : Restaurant users have to have at least an Instagram Account

2.5 Functional Requirements

The system provides to users a simple and user-friendly interface to:

For everyone :

- [R1]: Register and Login/Logout from the application.
- [R2]: Customize personal informations and profile picture.
- [R3]: Report missing/malformed/outdated informations or receive email support.

For the users :

- [R4]: Choose the role and category..
- [R5]: Review and complete missions.
- [R6]: Earn points from missions and use them for discounts in reservations.

- [R7]: Search for restaurants by area or specific name, refine the search with filters.
- [R8]: Complete reservations and review them.
- [R9]: Write a review on a restaurant
- [R10]: Invite a friend to subscribe to the app
- [R11]: Leave a review on a restaurant
- [R12]: Add a restaurant to the favourite list.

For the restaurateur :

- [R13]: Insert details about the restaurant, the cuisine, the location and social networks.
- [R14]: Insert a custom menu by adding dishes and specifying their characteristics.
- [R15]: Manage incoming reservations.
- [R16]: View reviews about him

3 Use Cases Analysis

In this section we will provide a list of the most relevant Use cases of the system, for a total of 6 of them:

Use Case 1

Name	Register as user to APPetito
Actor	Visitor
Requirement	[R1] , [R4]
Entry Condition	The Visitor has installed the Mobile Application on his/her device
Events Flow	<ol style="list-style-type: none">1. The Visitor taps on the button « I eat food ».2. The Visitor fills all the obligatory fields and then taps on the "Continue" button.3. The Visitor choose his/her rôle by tapping on one of the buttons.4. The Visitor taps on « Pick an image » to choose the profile picture and proceeds on tapping « Upload image » to complete the registration.5. The system receives and stores all the data of the registration
Exit Conditions	The Visitor has registered to APPetito as user
Exceptions	<ol style="list-style-type: none">1. The Visitor has not inserted a Name and a Surname

Use Case 2

Name	Register as restaurateur to APPetito
Actor	Visitor
Requirement	[R1] , [R13]
Entry Condition	The Visitor has installed the Mobile Application on his/her device
Events Flow	<ol style="list-style-type: none">1. The Visitor taps on the button « I sell food ».2. The Visitor fills all the obligatory fields, specify the type of restaurant, and then taps on the "Continue" button, for two windows.3. The Visitor taps on « Pick an image » to choose the profile picture and proceeds on tapping « Upload image » to complete the registration.4. The Visitor specifies the payment method, then tap on « continue »5. The Visitor has to choose the opening days and hours of the week, specifying if there is also an afternoon break. Then tap on « continue »6. The Visitor can fill the fields with the social networks links and then tap on « continue »7. The Visitor creates the menu tapping on « Add new dish » and filling all the fields to create a custom menu, then taps on « finish » to complete the registration.8. The system receives and stores all the data of the registration

Exit Conditions	The Visitor has registered to APPetito as restaurateur
Exceptions	<ol style="list-style-type: none"> 1. The Visitor has not inserted a Name and specified the kind of restaurant 2. The Visitor has not specified the location of the restaurant and the P.IVA 3. The Visitor has not uploaded a profile picture 4. The Visitor has not specified a payment method 5. The Visitor has not specified opening days and hours 6. The Visitor has not specified a social account link 7. The Visitor has not uploaded at least one dish in the menu

Use Case 3

Name	Search restaurant and use of filters
Actor	User
Requirement	[R7]
Entry Condition	The User is on the Home Screen
Events Flow	<ol style="list-style-type: none"> 1. The User, from the Search Page screen, taps on the Search bar and goes to Search Tool screen. 2. The User searches a restaurant or a place using one of the two Search bars, or press « search all ». 3. The User can filter the research by tapping « choose hour » or « filters » buttons on the top 4. The User can click on one of the results that appear in the list to proceed on the profile page of the selected restaurant
Exit Conditions	The User has successfully completed the research
Exceptions	<ol style="list-style-type: none"> 1. No results have been found with the name inserted 2. No results have been found with the location inserted

Use Case 4

Name	Complete a reservation
Actor	User
Requirement	[R8]
Entry Condition	The User is logged in
Events Flow	<ol style="list-style-type: none">1. The User click on the « card » of one of the restaurants and proceed to open the profile.2. The User tap on the button « make a reservation »3. The User fills all the fields required4. The User clicks on « Continue » to complete the reservation5. The system receives and stores all the data of the reservation
Exit Conditions	The User has successfully completed a reservation
Exceptions	<ol style="list-style-type: none">1. All the fields have not been inserted

Use Case 5

Name	Invite a friend
Actor	User
Requirement	[R10]
Entry Condition	The User is on the Home Screen after Login
Events Flow	<ol style="list-style-type: none">1. The User taps on the up-left corner the drawer icon2. The User taps on « invite a friend »3. The User insert the email of the friend that wants to invite4. The User taps on the « continue » button to complete the operation
Exit Conditions	The User has successfully invited a friend
Exceptions	<ol style="list-style-type: none">1. The email field has not been filled2. The email field has not been filled correctly

Use Case 6

Name	Send an email for support or report
Actor	User and Restaurateur
Requirement	[R3]
Entry Condition	The User or Restaurateur is on the Home Screen after Login
Events Flow	<ol style="list-style-type: none">1. The User or Restaurateur taps on the up-left corner the drawer icon2. The User or Restaurateur taps on « contact us »3. The User or Restaurateur, if wants to report a bug, taps on the checkbox « report bug » and chooses the correct answer.4. The User or Restaurateur fulfills the message field and press the « continue » button5. The system receives and stores all the data of the Report
Exit Conditions	The User or Restaurateur has successfully sent a report or a help request
Exceptions	<ol style="list-style-type: none">1. The User or Restaurateur didn't fulfill the message field

Use Case 7

Name	Add dish to the menu
Actor	Restaurateur
Requirement	[R14]
Entry Condition	The Restaurateur is on the Home Screen after Login
Events Flow	<ol style="list-style-type: none">1. The Restaurateur taps on « Menu » from the bottom bar2. The Restaurateur taps on the button « Add new dish here »3. The Restaurateur fulfills all the fields required4. The Restaurateur upload the picture of the dish pressing « Add picture » button and then tapping on « Pick an image » and , once chosen, « Upload image ». By pressing on « Continue » the upload is completed.5. The Restaurateur press on « continue » button to complete the action.6. The system receives and stores all the data of the Dish inserted.
Exit Conditions	The Restaurateur has successfully added a new dish to the menu
Exceptions	<ol style="list-style-type: none">1. The Restaurateur didn't fulfill all the required fields2. The Restaurateur didn't choose a picture for the dish

Use Case 8

Name	Adding opening hours and days
Actor	Restaurateur
Requirement	[R13]
Entry Condition	The Restaurateur is on the Home Screen after Login
Events Flow	<ol style="list-style-type: none">1. The Restaurateur taps on « Profile » from the bottom bar2. The Restaurateur taps on the card « opening hours »3. The Restaurateur adds opening hours and days by clicking thr specific day from the below list, specifying if there will be an an afternoon break by using the dedicated checkbox.4. The Restaurateur taps on the « confirm »button to conclude the operation.5. The system receives and stores all the data of the Opening time inserted
Exit Conditions	The User has successfully added an opening time table
Exceptions	-

Use Case 9

Name	Manage reservations
Actor	Restaurateur
Requirement	[R15]
Entry Condition	The Restaurateur is on the Home Screen after Login
Events Flow	<ol style="list-style-type: none">1. The Restaurateur taps on « Bookings » from the bottom bar2. The Restaurateur can tap on the buttons « Decline » , « Accept » , or « Complete » in order to manage the incoming reservations.
Exit Conditions	The Restaurateur has successfully managed the incoming reservations
Exceptions	-

Use Case 10

Name	Login with social network
Actor	User
Requirement	[R7]
Entry Condition	The User taps on the social button on the Login Page
Events Flow	<ol style="list-style-type: none">1. The User taps on the social button and he will be redirect to the Login Page of the social2. The User insert his social credentials and accepts terms and conditions3. The User will be redirected to the Login Page of the App4. The system check if the User was already registrated or not and if the credentials are correct5. The User will be redirect to the HomeScreen if he was already registrated, to the ChooseRole Page instead
Exit Conditions	The User has insert correct credentials
Exceptions	<ol style="list-style-type: none">1. The User inserts wrong credentials2. The User was already registrated in a different method

4 Architectural Design

Our application is composed mainly by two components : a FrontEnd part, developed with the React Native framework, and a BackEnd that relies on Firebase, in particular we used Cloud Firestore Database and Firebase Storage.

These two components are connected to each other and display on the device all the information needed to complete the operations required, adapting the showed results on the actions and behave of the User/Restaurateur .

4.1 Front End: Mobile Application

APPetito is a React Native application made by several screens which are all listed in the section « User Interface Design Section ».

The purpose of the app is to be runned on mobile devices or tablets in order to give the possibility to the user to complete a reservation in whatever moment of the day, and for the restaurateur, or to some employees, to check the activities of the restaurant even without being phisically in the place and to reach a higher amount of potential customers.

For this reason we choosed to use React Native Framework because it allowed us to have a « real time » way to implement the activities of APPetito, and because by being a **cross-platform** framework it allows to reach a large number of users.

This Application has being created to be **multi-thread**, by the need to handle user Client actions and Restaurateur actions at the same time, to update informations in real time such as reservations, reviews and missions, score systems and so on.

The Mobile Application is composed of a total of X screens (better described in Section Y), and the most important of these are:

(for the Client)

- Search Screen
- Goals
- Profile

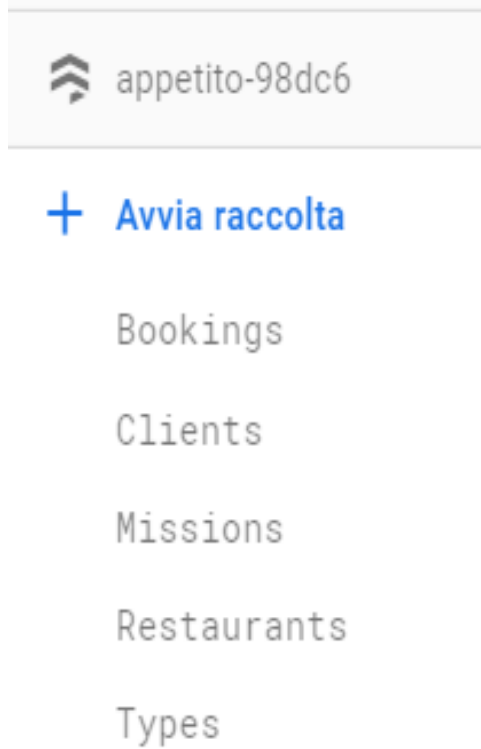
(for the Restaurateur)

- Menu
- Bookings
- Profile

4.2 Back End: Firebase

The Firebase back end is where all the users, restaurants, missions, reviews and reservations data are stored. The back end is also encharged of authentication and user session management (a well implemented and tested functionality of Firebase).

Since Firebase is a NoSQL Database we have a collection style structure, where we have 5 main collections: **Clients**, **Restaurants**, **Bookings**, **Missions** and **Types**.



Bookings : In this collection are stored all the bookings data made by each user. It has a field called ID_Client with the ID of the user that made the reservation, and a field called ID_Branch with the path of the branch in which the user made the reservation.

Clients : In this collection are stored all the Clients data that the user inserts when he make the registration as Client. It also contains all the missions associated to the same type of the user and all the generic missions. For each mission the user has a status that can be 'To Do', 'In Progress' or 'Completed'.

Missions : In this collection are stored all the missions that we implement.

Restaurants : In this collection are stored all the Restaurants data that the user inserts when he make the registration as restaurateur. Every restaurant can have more than one branch and each branch has his own bookings, reviews, opening hours, allowed

payment methods etc.

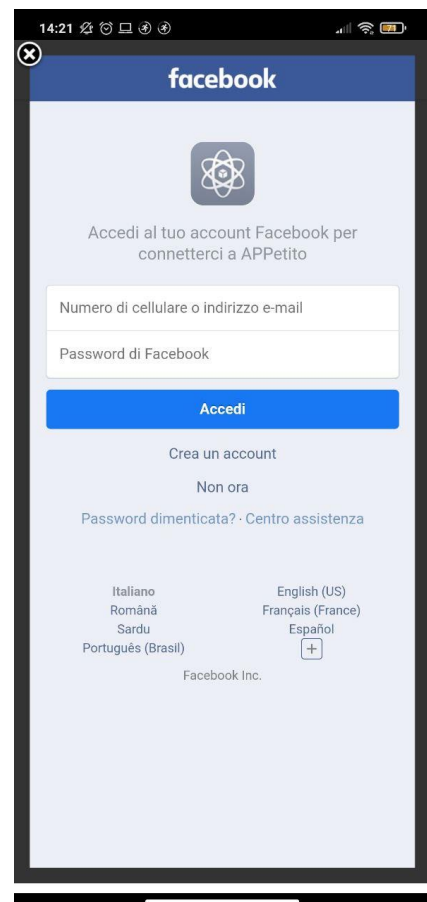
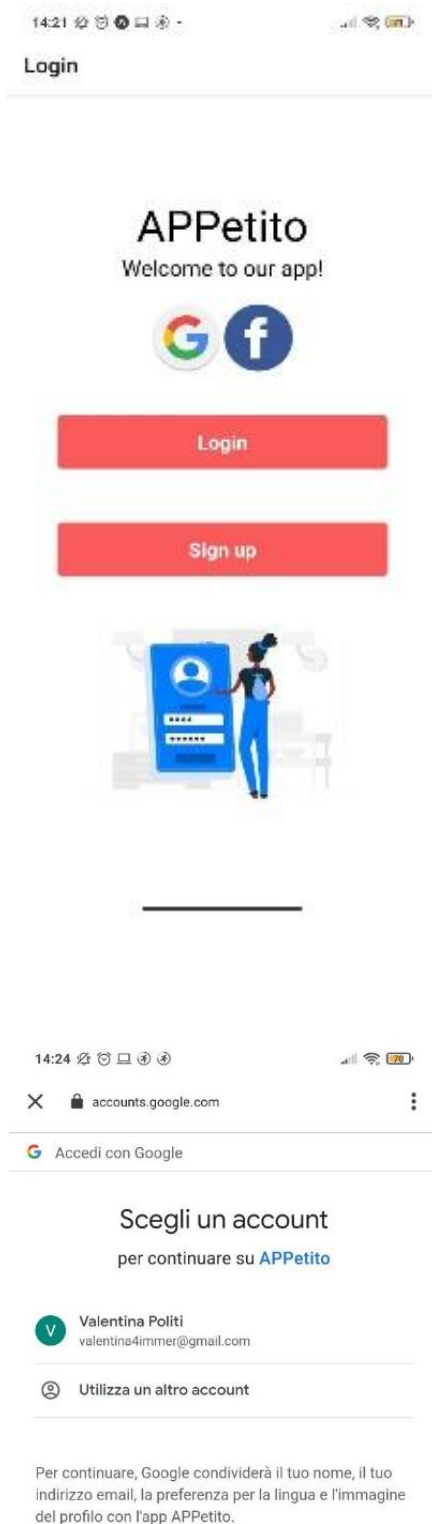
Types : In this collection are stored the types information which the Client can choose.

4.3 Libraries used

'firebase'	'expo-image-picker'
'firebase/auth'	'react'
'firebase/storage'	'react-native'
'firebase/firestore'	'react-native-gesture-handler'
"expo"	'@react-navigation/native'
"expo-google-app-auth"	'@react-navigation/stack'
'expo-facebook'	'galio-framework'
'@expo/vector-icons'	

5 Third Party interaction

Google and Facebook login are implemented as follow:



6 User Interface Design

In this section we list all the Screens of the Mobile Application, explaining briefly their functionalities , along with the decisions we made to provide a nice and pleasant User Experience.

6.1 Colors and design

The color are kept through all the application which, along with the simple and intuitive design, have been thought to create a user friendly GUI and aesthetically likeable.

6.2 List of User Screens

User registration screens

Name	Short description
ChooseRole	To choose to be a User or a Restaurateur
UserNameSurname	To insert name, username and if the user has been invited by another user
ChooseCategory	To choose what « category » of user
UserProfilePicture	To upload the profile picture

User navigation screens

Name	Short description
Search	To open search tool and show suggested restaurants
SearchTool	To search by name, place or everything
Booking1	Show the results of the research that can be filtered
RestaurantPage	Shows informations and pictures of the restaurant
Menu	Shows informations and pictures of the dishes of menu
MakeReservation	To complete a reservation in a restaurant
MakeReview	To write a review about a restaurant
MyReviews	To checkout all the reviews written by the user
MyFavourites	To checkout all the favourited restaurant by the user
InviteAFriend	To invite a friend , by email, to join the app
ContactUs	To report a bug or request support
EmailFormat	To write an email to report a bug or ask for support
Profile	Show an overview about the profile of the user
Settings	To activate the newsletter or read more informations
ProfilePicture	To change the profile picture
FriendProfile	To see the friend profile

6.3 List of Restaurateur Screens

Restaurateur registration screens

Name	Short description
ChooseRole	To choose to be a User or a Restaurateur
InsertRestaurant	To insert the name and the type of the restaurant
LocationRestaurant	To insert location and P.IVA of the restaurant
RestaurantProfilePicture	To upload the profile picture
PaymentMethod	To choose the payment method
OpeningDays	To choose the opening time during the week
InsertSocials	To insert the link of socials pages
InsertMenu	To upload a custom menu
InsertDish	To insert informations and picture of a new dish

Restaurateur navigation screens

Name	Short description
Profile	An overview on restaurant informations
ManagePhotoRestaurant	To change the restaurant profile picture
ManageOpeningHours	To change the opening time during the week
ManageSocialLinks	To insert social networks links
ManageMenu	To review and insert new dishes in the menu
ManageDish	To insert a new dish in the menu
MyReviewsPage	To view the reviews of the clients on the restaurant
Bookings	To manage incoming reservations
ContactUs	To report a bug or request support
Settings	To activate the newsletter or read more informations

7 Implementation and Integration

We proceeded in the implementation and integration in three distinct phases, written here below in order of execution

7.1 Front End implementation

We started by creating a prototype alpha version of the user interface, designing what were the most important interaction in order to have a intuitive and simple user experience. Once we figured out what was the best structures we moved on a second step where we thought about colors, shapes and positions of the different functions, in order to provide a user friendly interface.

7.2 Back End implementation

We created the Back end skeleton of the system, by configuring Firebase Cloud Firestore in order to store every needed information. In particular, we started by registrate some restaurants, and then some clients.

7.3 Front End and Back End integration

In this phase we integrated the front end and back end testing every function, considering exceptions handling e fixing the last errors, sometimes modifying a little the previous designed front end in order to simplyfing even more the interaction and turn it even more fluid and dynamic.

8 Testing

The testing done so far was the one strictly necessary in order to allow us to present a working demo. This is a project which is supposed to be develop with more features and the developement will require a longer time both for design and implementing phase and also for the testing phase.

9 Software System Attributes

9.1 Reliability

The system guarantees a 24/7 service. Even if some Back end data has to change the system can be updated without downtimes, because every change in the Database will be immediately received and correctly updated from the Mobile Application.

9.2 Availability

The system is built and implemented using highly tested and safe libraries used by most the other mobile applications developed using React Native framework. Because the application doesn't include features which needs critical safety measures, we Conclude that the system availability fits with the standards of all the other applications Commonly available on the market, which means at least about 99%.

9.3 Security

The only Security aspect that the system must handle is the credentials transmission and storage: Firebase does that, and obviously no password data is stored in the Database.

9.4 Compatibility

The system id designed with the target of being used by a large amount of people. For this reason we created it in order to fit a high number of different devices. Even if for now is only compatible with Andorid, is our intention to also make it available for iOS, to increase the public. The application doesn't require particular technologies which are not commonly

already installed on every modern smartphone and devices like tablets, which makes APPetito even more versatile.

9.5 Scalability

The architecture is simply scalable as the number of users grows during the time. Enlarging the structure of the system is be an easy task: Firebase helps us a lot in this case, because the number of possible users is huge and more users registrate and more restaurants will be added by simply adding them one after the other in the Database (also thanks to the NoSQL structure).

9.6 Future developements

The application APPetito is supposed to be developed to become a bigger project implementing a number of features that we could not yet implement due to the short time available for the presentation. Here follows a short list of future functions that we have the intention to create.

- **Stats page** : a page which shows an overview of the different paramètres modified by the user through his/her own expériences, such as number of missions not yet completed, graphs to show the increment of missions completed in a certain time, count of the scores earned and used in a certain time span etc...
- **Unlockable functions** : items or activities that can be unlocked through scores, for example particular filters (trendy dishes, dishes that follows particular diets...), premium or category missions, reputation badges.
- **Reputation system** : reputations scores that can be given by a restaurateur arbitratly if the behave of the customer have been particularly appreciated (for example if has been generous with tips, or kind in ordering the meal). With such scores is possible to buy badges to display on the profile, which will be seen by other restaurateurs during the bookings management which will suggest them to give you a priority in the reservations in case of queues.
- **New user categories** : for example a Business category, in order to customize new missions and features to incentivize meals for working purposes. A Loyal category for those customers who prefer eating in a limited number of restaurants, in order to improve their experience in such places.
- **Choose the seat** : allowing the Restaurateur to insert 360° pictures specifying each seats on them, registering it with a unique number. The user will be able not only to complete a reservation but also to choose where to seat and have a 3D overview of the place around.
- **Social networks** : connecting more social networks to the application, giving the possibility to share contents easier in order to advertise even more the experience of the user and the activity of the restaurant.