Politecnico di Milano

Software Engineering 2 Project
a.y. 2020-21

Customers Line-up
Design Document
Version 0.0

Banfi Stefano Alessandro
Bresciani Matteo

Referent professor: Di Nitto Elisabetta

December 3, 2020

Contents

Chapter 1

Introduction

1.1 Purpose

The Design Document aims to give usefull information to help in software development by providing the details for how the software should be built. In particolar it should be detailed enough so that developers could code the project without having to make any significant decisions. This is done thanks to detailed description with graphical documentation of the software design for the project including different diagram types and other supporting requirement informations.

1.2 Scope

The main scope of the system is to provide users the possibility to make a booking in order to give access to the market. This could be done with two options: the first allows users to be inserted in the virtual queue; instead, the second give the possibility to schedule the booking in a precise moment in an particular day. So, the system have to reply users' requests in real time without waiting more than few seconds due to its reliability. To achieve this, the system is organized with a three tiers architecture which divides the systems in independent modules: presentation, application and a data tier. The detailed architecture will be described as well in the next chapter.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **Smart User** see the RASD;
- **Mobile User** see the RASD;

- **Visit**: see the RASD;
- **Reservation** see the RASD;
- **Application Server**: part of the application layer. It refers to the server needed to run the
- **Booking**: it's the generic appointment. It could be either a Visit or a Reservation;
- **Bottom navigation bar**: graphical object that allows the user to display different destinations at the bottom of a screen;
- **Horizontal fragmentation**: it consist in divide a table into a set of smaller table;
- **Cross-platform development**:;
- **Servelets**:;

1.3.2 Acronyms

- **DB**: Database;
- **ACID**: Atomicity, Consistency, Integrity and Durability;
- **DBMS**: Database Management System;
- **RDBMS**: Relational Database Management System;
- **SQL**: Structured Query Language;
- **HTTPS**: Hypertext Transfer Protocol Secure;

1.3.3 Abbreviations

- :;
- :;
- :;
- :;

1.4 Revision history

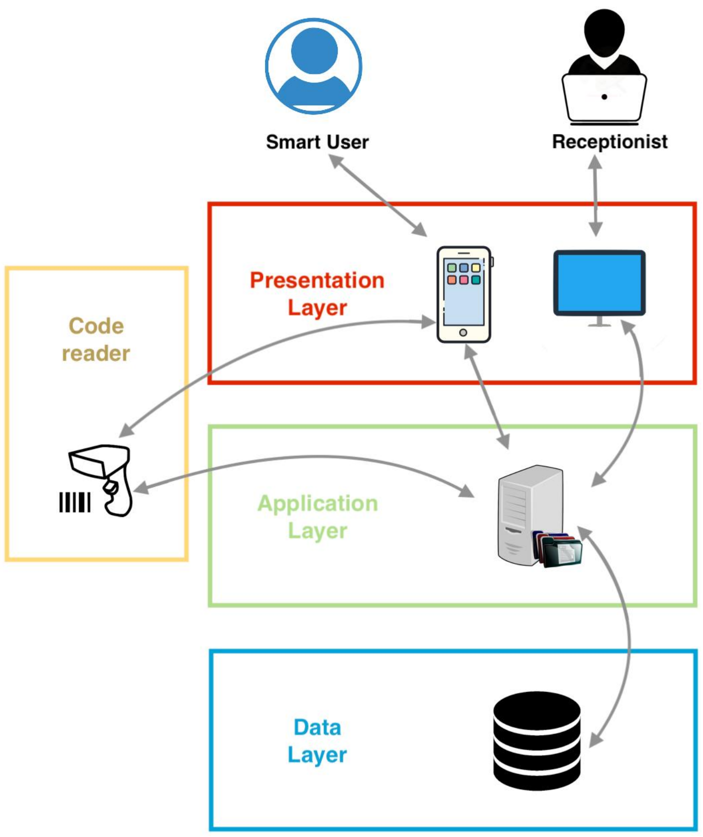
1.5 Reference Documents

1.6 Document Structure

Chapter 2

Architectural Design

Figure 2.1: Three tiers architecture of the system



2.1 Overview: High-level components and their interaction

The system is organized following the three tiers architecture. This aims to decouple logical layers in order to gurantee an horizontal scalability and an high fault tolerance. Graphically it's shown in the figure 2.1.

Presentation layer. It's the front-end layer which consists of the user interface. We have two types of user interface, depending on his functionality:

- **CLup**: It's the mobile application used by users who have a smartphone. They can manage their booking by themselves;
- **CLup Operator**: It's the desktop application used by receptionists that act as an intermediary to manage booking of users that have only a mobilephone.

Application layer. It deals with the model of the system, by containing the business logic of the application. In our system it consists in a remote server to which mobile and desktop applications have to connect due to manage any bookings.

Data layer. It's composed by a data storage system. It includes:

- User sensitive data asked during the registration process;
- Information about user's grocery shopping;

2.2 Component view

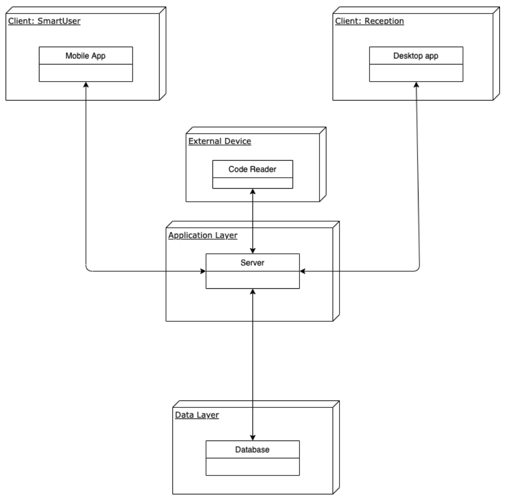


Figure 2.2: High-Level component diagram

In this section we describe the architecure with respect to its components and how they are wired together to form our system. Indeed, communication between Server and Clients is ensured by appropriate components which are called **Network Manager**. By an high-level point of view (figure **??**) the main subsysystems correspond almost to the 3 layers in the 3 tiers architecture, with the addition of the **Market's subsystem**. The last one includes the **QRCode reader**, which has different functionality:

- It checks if QRCode at the entrance is valid. If it's valid it will submit it. In addition a manual insertion of the QRCode is provided, due to allows Mobile Users to insert his code sent by SMS;
- It controls the **Shop door**. If a QRCode is submitted, the reader allows users to enter by opening it;

It follows a detailed description of the other significant components.

2.2.1 Application layer

It's the core of our system. It's composed by:

- **CLup**: In our model this component acts as a controller. It deals with the stochastic time estimations to control the flux of users inside and outside the market. Moreover this component [...] In order to do that, it's also linked to the DB which provides data that are going to be processed and analysed;
- **Auth User and Auth Reception**: these components grants respectively the access of users and receptionist by verifying their credential. In particular they are also accountable for users' registration;
- **Visit Schedule and Reservation Queue**: they manage respectevely Visit and Reservation requests of Users and Receptionists. In addition They provides QRCodes to the clients;
- **Notification Manager**: it's the component necessary to notify users about their turn in the queue;

2.2.2 Client's layers: CLup and CLup Operator

Each one of them is composed by components in a similar way, even if they have different functionalities. Both they've got, as we can see in the figure **??**:

- **GUI component**: it provides the graphical interface in which User and Receptionist can interact with the system;
- **Network Manager**: it's used to interact with the application server by sending or receiving informations;

In addition CLup is composed by a **Request Handler** which exposes methods to User to handle his booking. Then, it interract with the Network Manager to send request to the Server.

Instead, CLup Operator has the **Booking Manager**, which is a similar component to the previous one but with more functionalities. In particular it handles any booking of mobile users and their sensitive information. So it's allowed by application server to query all users' sensitive data with higher privileges than CLup.

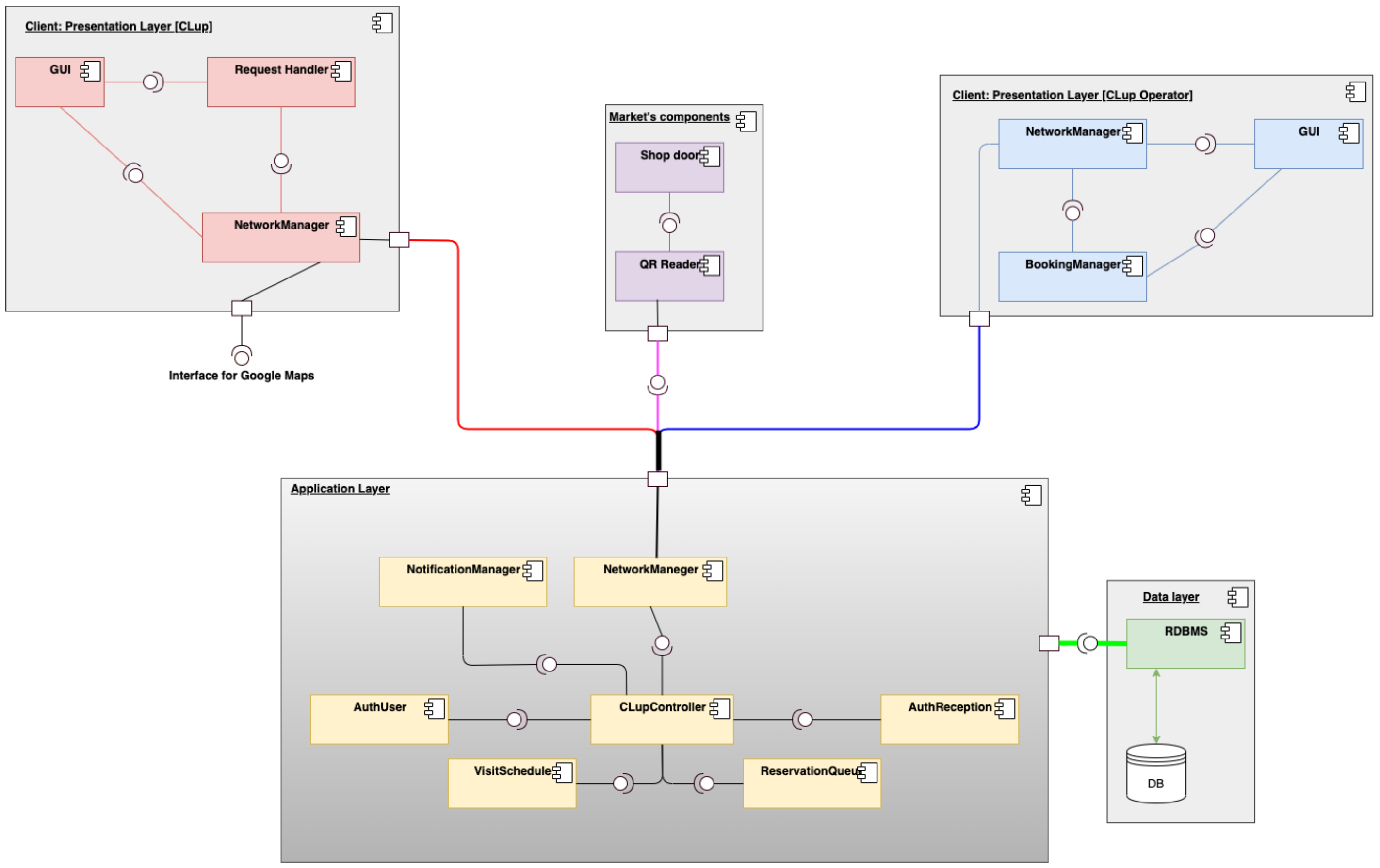


Figure 2.3: Component Diagram

2.2.3 Data Layer

In our architecture the data layer is composed by a relational DB needed to store informations about users and their dynamics in the market.

In particular it should be connected to the server placed in the application layer. To reach the goal we plan to adopt a **RDBMS** in order to deal with a relational database. Indeed, it ensures consistency of data due to the *ACID* properties.

In addition, it can process a large amount of data, which is suitable for our application.

Besides, due to a better efficiency, RDBMS provides *horizontal fragmentation*. It allows to fragment entities of our model with respect to each different market. This is why tuples regarding different markets will never be queryied together. The only which won't be fragmented is the Market entity.

Moreover, it includes a software program which is designed to capture request over a network of the application server. From this each user and receptionist in fact could retrieve informations needed through SQL Query by connecting to it.

An other important aspect is the security of the data due to mitigate any risks of violation. In order to do that we limit its access exclusively only to the application server. Communication between them will be also encrypted and accounts' passwords will be hashed.

2.3 Deployment view

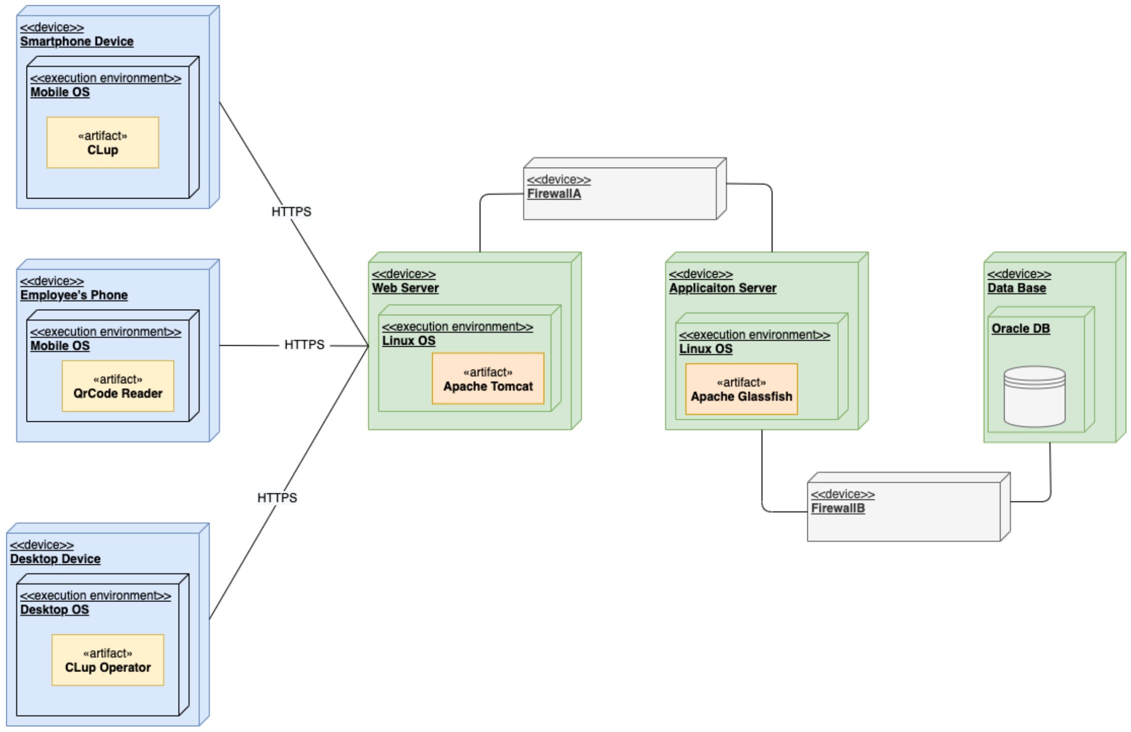


Figure 2.4: Deployment Diagram

In this section we'll explain the structure of our run-time system and the interaction between the hardware and the software parts. This is shown in the deployment diagram in the figure **??**.

For the client's side, CLup and CLup Operator will be hosted respectevely on a mobile and desktop device. Softwares will be designed both following a cross-platform development. We choose cross-platform to ensure an easier and quicker implementation; moreover it includes a better uniformity and maintenability. The main OSs choosen are respectevely iOS and Android for CLup and macOS and Windows for CLup Operator. In addition is provided a third device used by a generic employee of the market which, interfaced with a QRCode scanner, sends to the server any request for entering or leaving the market. Instead, the server is organized as well following a 3 Tier Server Architecture. Each tier is composed of:

1. **Web Server**: it's the node to which every clients connect. It handles each request from the clients connecting by HTTPS. It's built using **Apache Tomcat** This, supported by Apache Software Foundation, a nonprofit

corporation, is an open source software which is lightweight and suitable for our web server;

2. **Application Server**: this node hosts the core of our business logic. In particular we set **Apache Glassfish** which is leading of the Java EE standard to run servlets. Compared to Tomcat, is a full-blown Java EE application server, because it includes more features. Indeed it provides backup and recovery services due to a better availability a reliability;

3. **Data Base**: this machine contains each data of our system. In fact it's composed by a RDBMS in order to store and retrieve data. Besides, we choose **Oracle DB** from Oracle Corporation as management system on it. We choose it because of its high performances, which are necessary to process data quickly. In particular it can handles large volume of data since it have to manage a large number of users' requests simultaneously;

Each tier is divided from the others with **2 Firewalls** which aim to filter any incoming and outgoing network traffic. In particular:

- **FirewallA**: it filters traffic coming from the WS to AS. It allows requests from receptionist, logged user and employees' devices. This is due to mantaining the application server's integrity;

- **FirewallB**: this protects the DB by filtering any packets, except for the application server. In this way we gurantee an high security against any data violation;

2.4 Runtime view

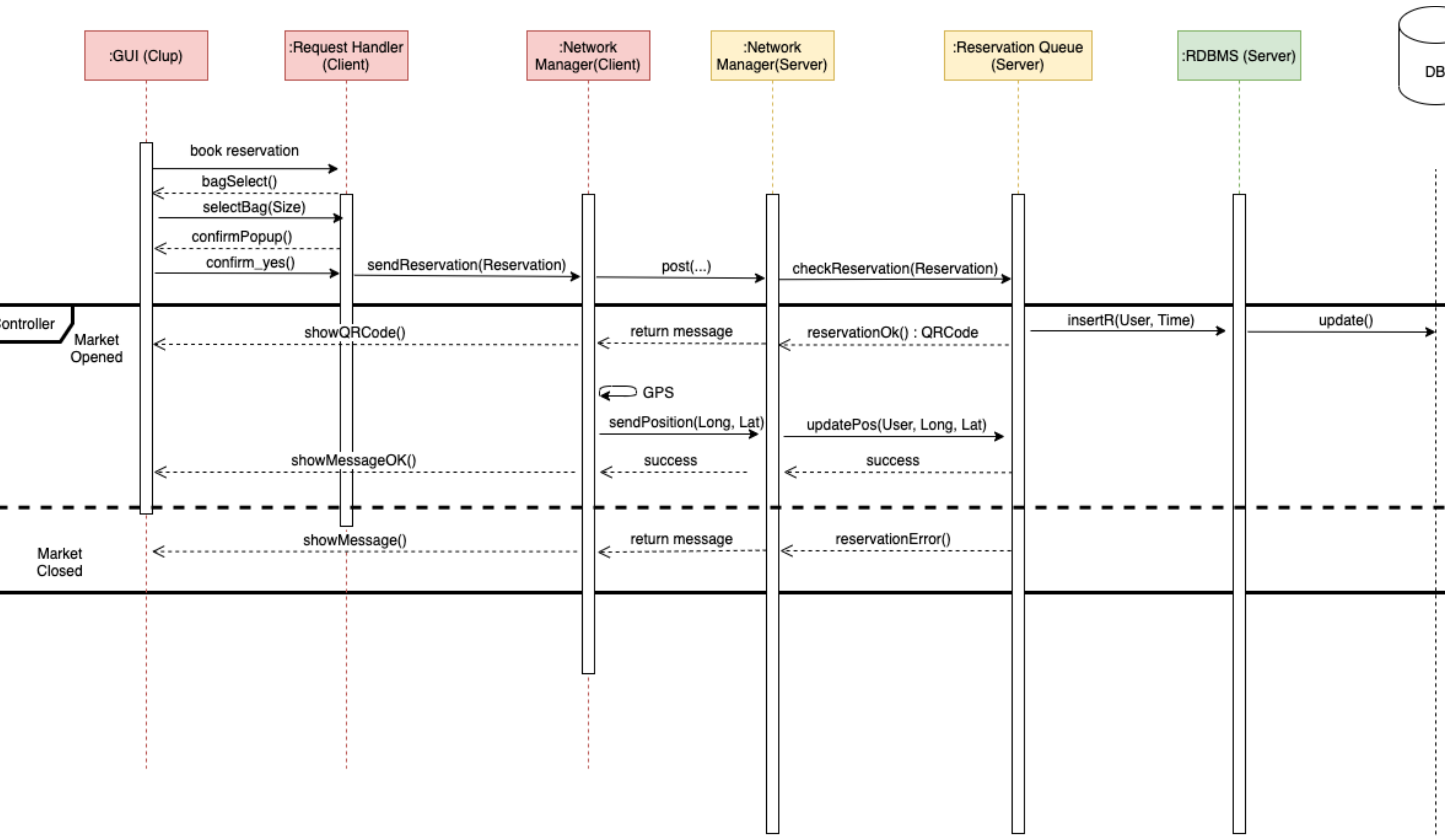


Figure 2.5

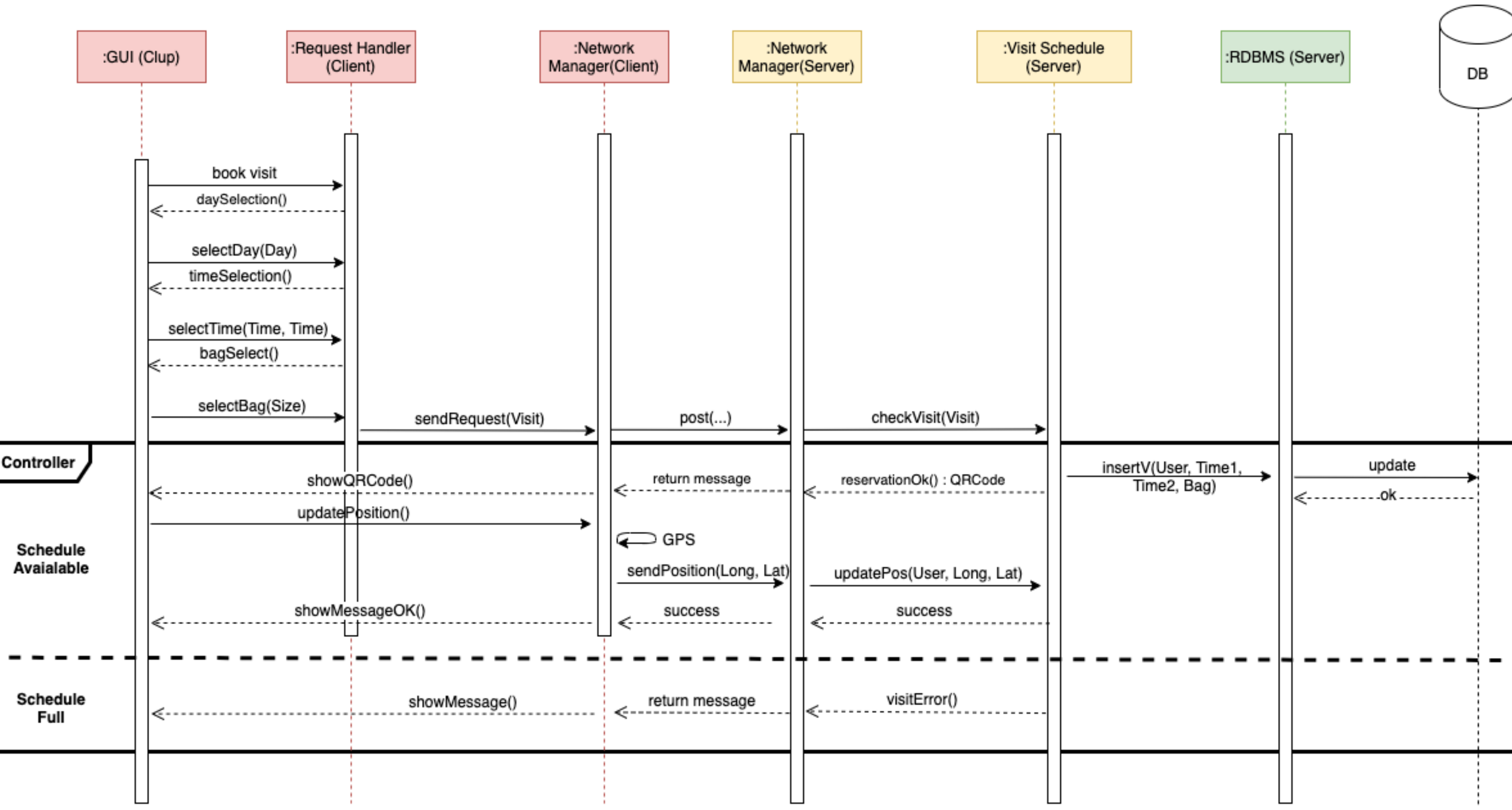

:GUI (Clup)
:Request Handler (Client)
:Network Manager(Client)
:Network Manager(Server)
:Visit Schedule (Server)
:RDBMS (Server)
DB
book visit
daySelection()
selectDay(Day)
timeSelection()
selectTime(Time, Time)
bagSelect()
selectBag(Size)
sendRequest(Visit)
post(...)
checkVisit(Visit)
Controller
showQRCode()
return message
reservationOk() : QRCode
insertV(User, Time1, Time2, Bag)
update
ok
updatePosition()
GPS
Schedule Avaialable
sendPosition(Long, Lat)
updatePos(User, Long, Lat)
showMessageOK()
success
success
Schedule Full
showMessage()
return message
visitError()

Figure 2.6

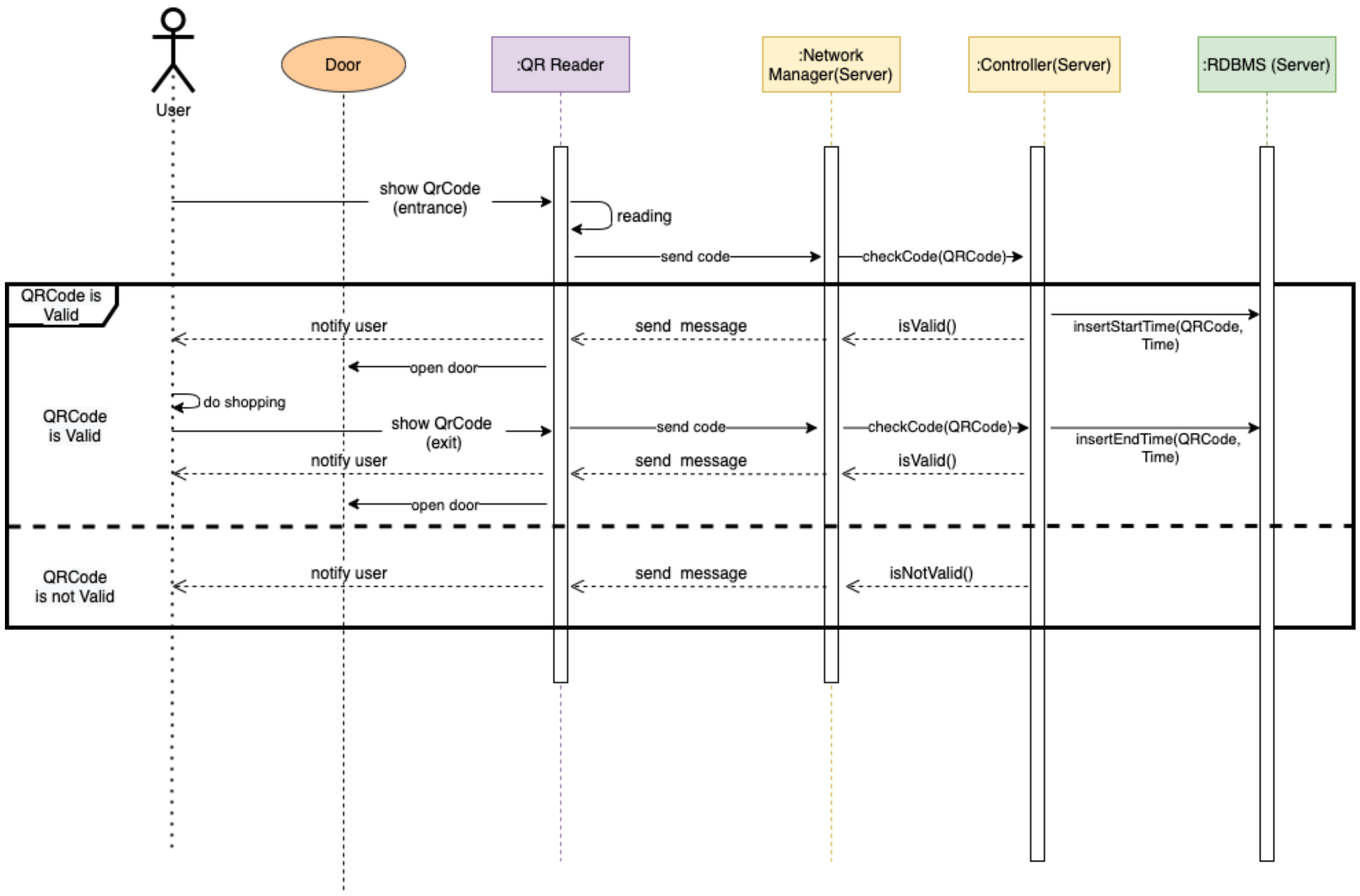

User
Door
:QR Reader
:Network Manager(Server)
:Controller(Server)
:RDBMS (Server)
show QrCode (entrance)
reading
send code
checkCode(QRCode)
QRCode is Valid
notify user
send message
isValid()
insertStartTime(QRCode, Time)
open door
do shopping
QRCode is Valid
show QrCode (exit)
send code
checkCode(QRCode)
insertEndTime(QRCode, Time)
notify user
send message
isValid()
open door
QRCode is not Valid
notify user
send message
isNotValid()

Figure 2.7

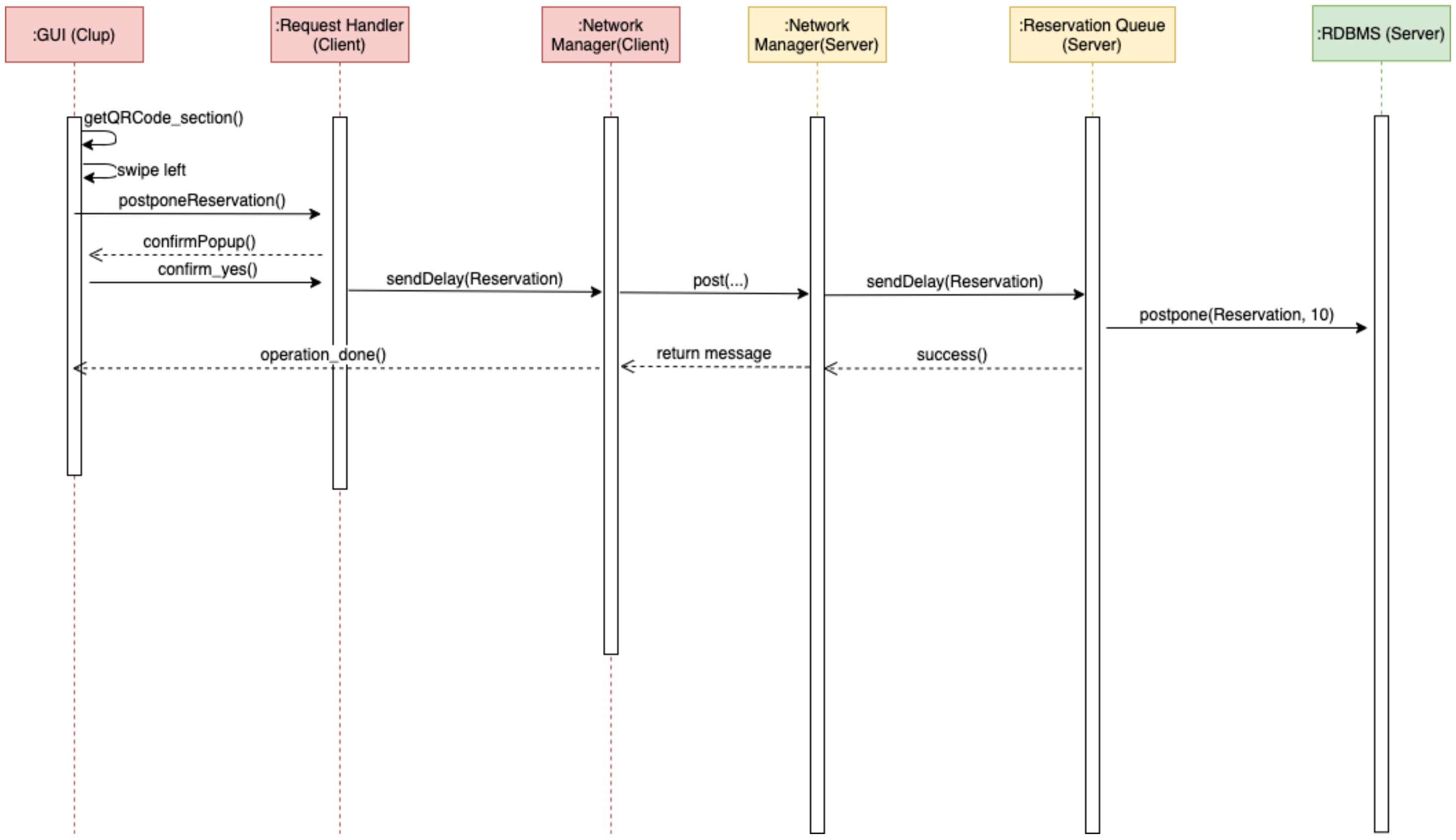

:GUI (Clup)
:Request Handler (Client)
:Network Manager(Client)
:Network Manager(Server)
:Reservation Queue (Server)
:RDBMS (Server)
getQRCode_section()
swipe left
postponeReservation()
confirmPopup()
confirm_yes()
sendDelay(Reservation)
post(...)
sendDelay(Reservation)
postpone(Reservation, 10)
operation_done()
return message
success()

Figure 2.8

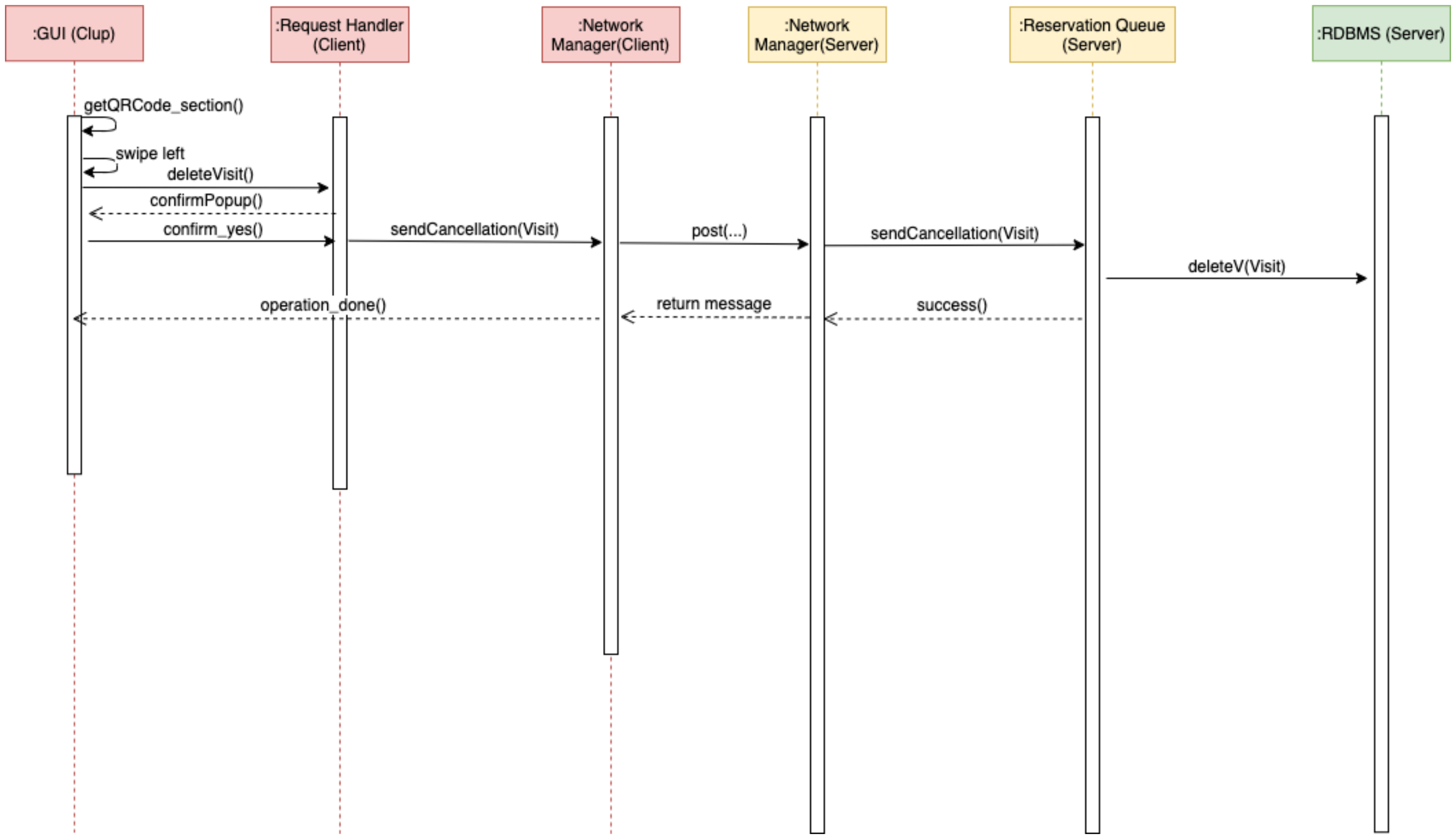
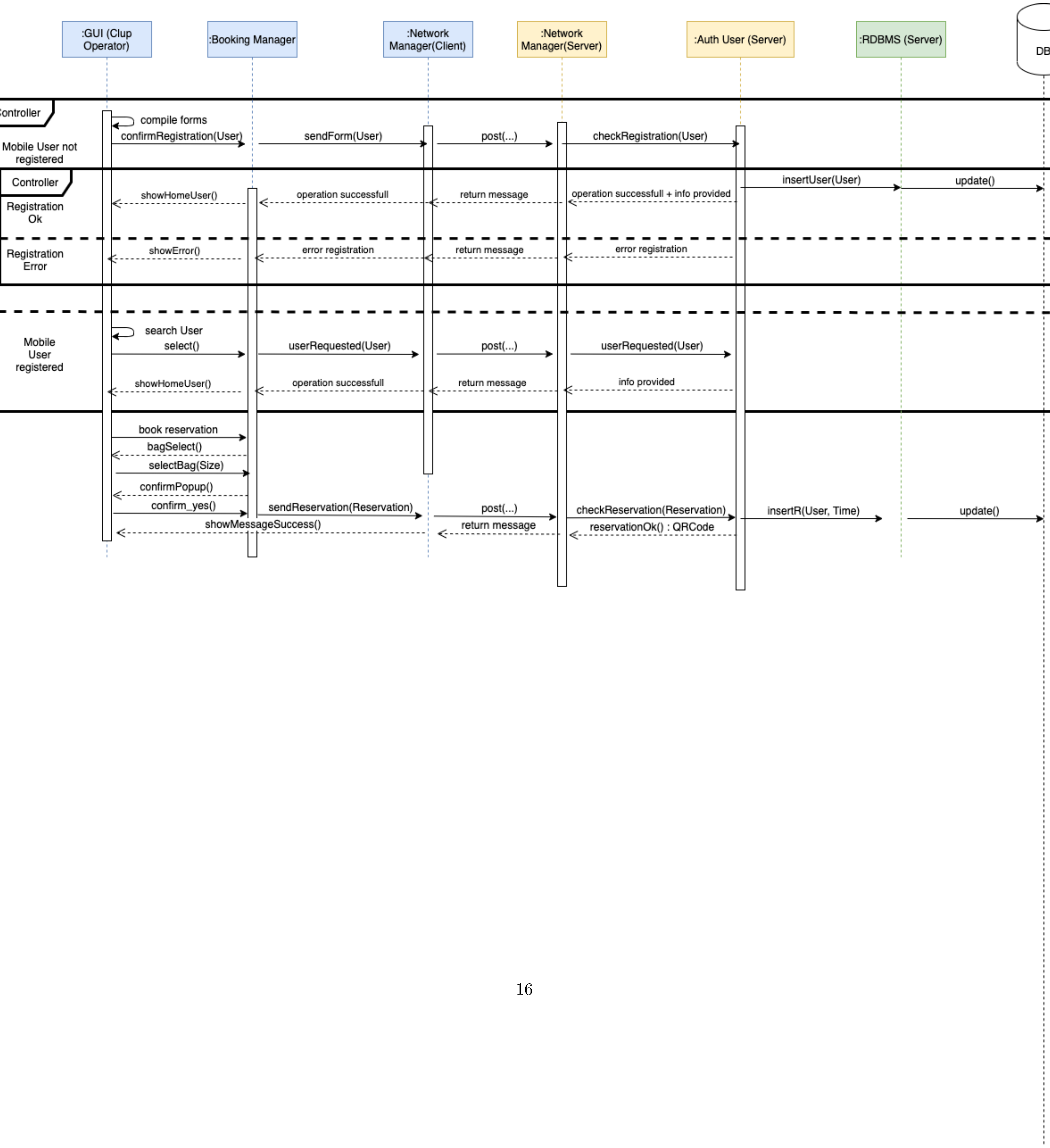

:GUI (Clup)
:Request Handler (Client)
:Network Manager(Client)
:Network Manager(Server)
:Reservation Queue (Server)
:RDBMS (Server)
getQRCode_section()
swipe left
deleteVisit()
confirmPopup()
confirm_yes()
sendCancellation(Visit)
post(...)
sendCancellation(Visit)
deleteV(Visit)
operation_done()
return message
success()

Figure 2.9


:GUI (Clup Operator)
:Booking Manager
:Network Manager(Client)
:Network Manager(Server)
:Auth User (Server)
:RDBMS (Server)
DB
Controller
Mobile User not registered
compile forms
confirmRegistration(User)
sendForm(User)
post(...)
checkRegistration(User)
Controller
Registration Ok
insertUser(User)
update()
showHomeUser()
operation successfull
return message
operation successfull + info provided
Registration Error
showError()
error registration
return message
error registration
Mobile User registered
search User
select()
userRequested(User)
post(...)
userRequested(User)
showHomeUser()
operation successfull
return message
info provided
book reservation
bagSelect()
selectBag(Size)
confirmPopup()
confirm_yes()
sendReservation(Reservation)
post(...)
checkReservation(Reservation)
insertR(User, Time)
update()
showMessageSuccess()
return message
reservationOk() : QRCode

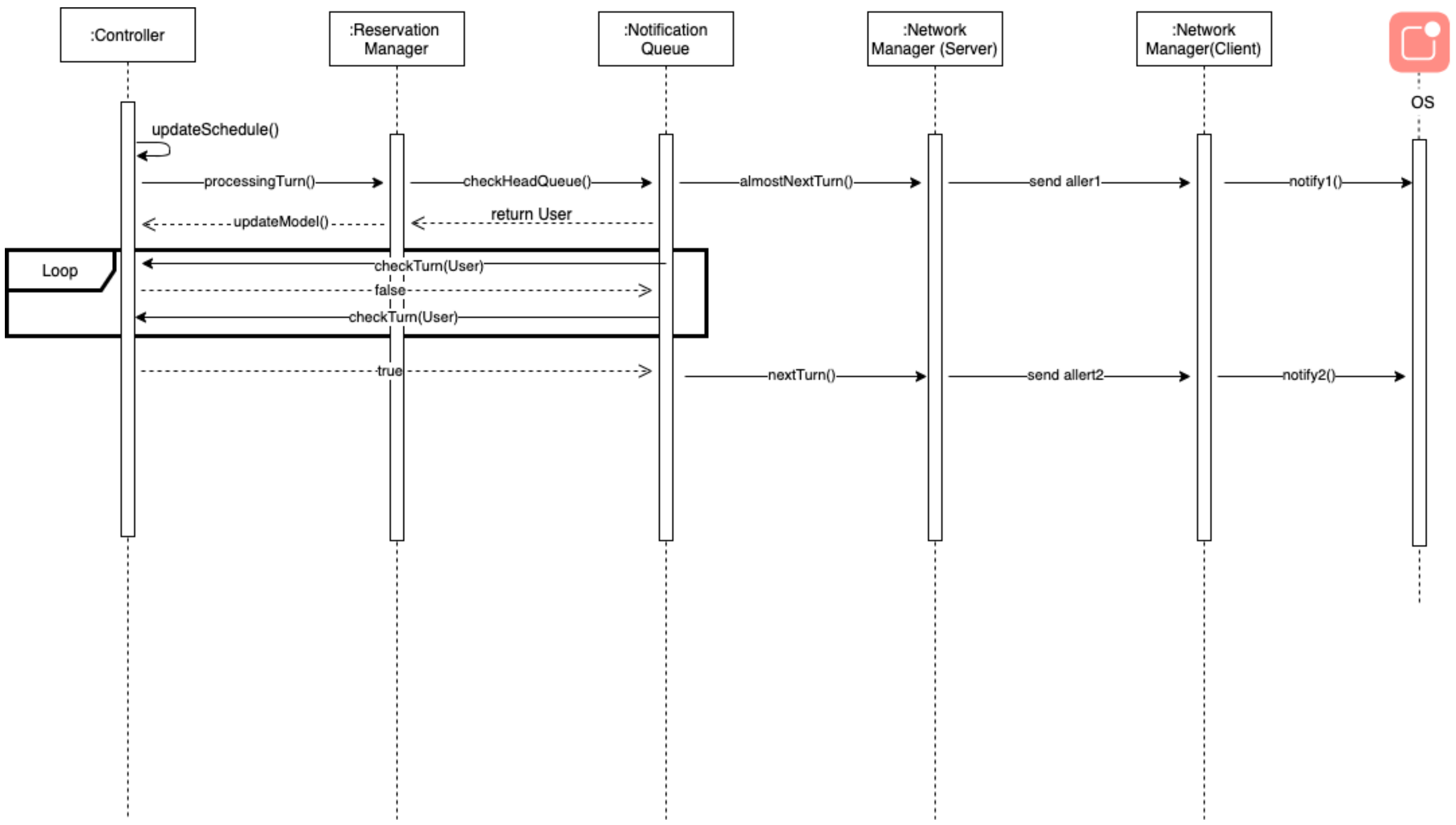


Figure 2.11

2.5 Component interfaces

ogni componente app server+db laptop receptionist

2.6 Selected architectural styles and patterns

mvc + tier + ..

2.7 Other design decisions

security+google api asyncrnous coomunication because eccc

Chapter 3

User Interface Design

In this chapter we will illustrate most of action allowed in CLup and CLup Operator using UI flowchart digrams. UI flowchart diagrams are used to model the interactions that users have with the software, by understanding how the system is expected to work. Diagrams are built using the mockups already put in the RASD, but in a more detailed way.

3.1 Mobile Interface: CLup

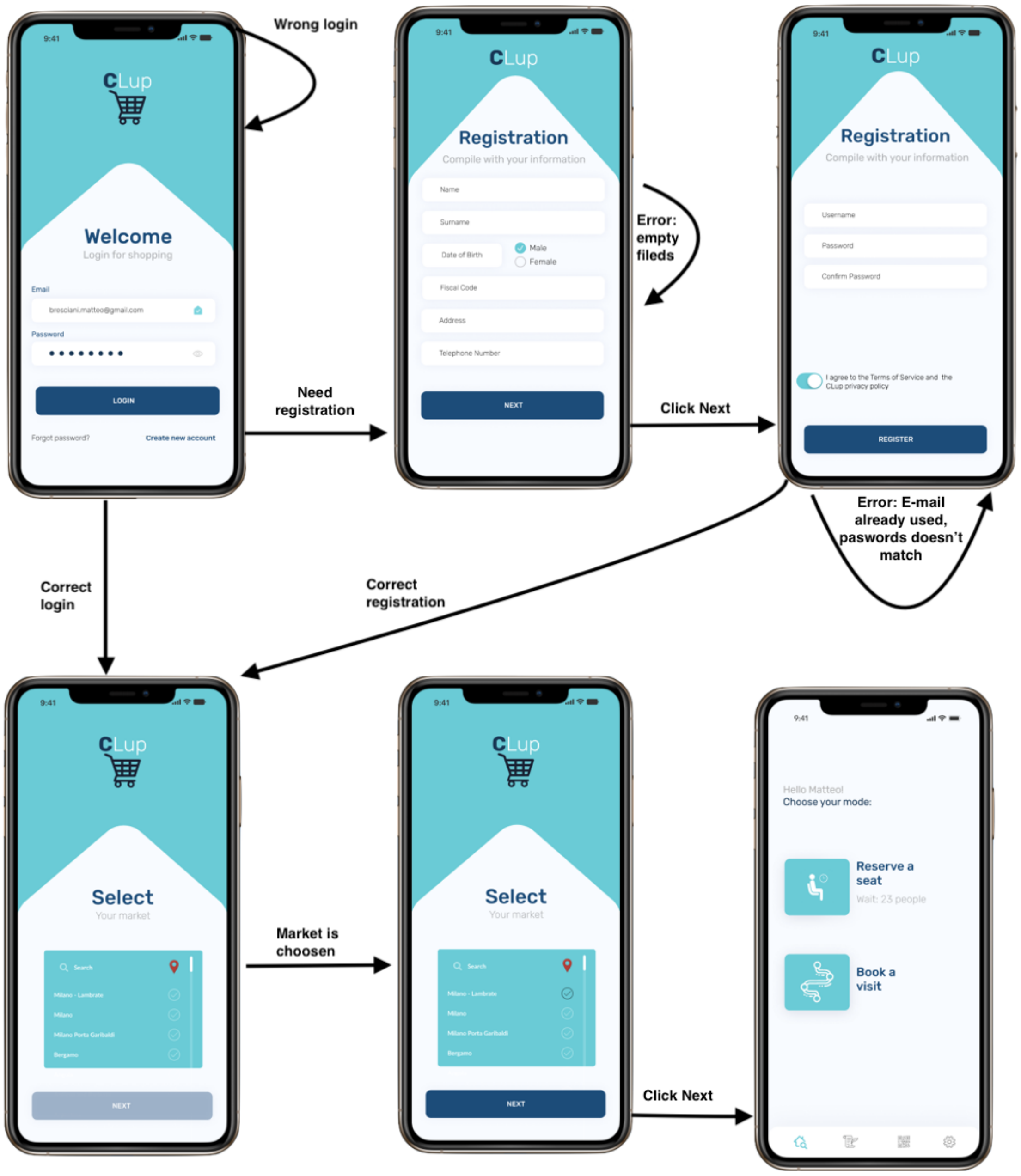
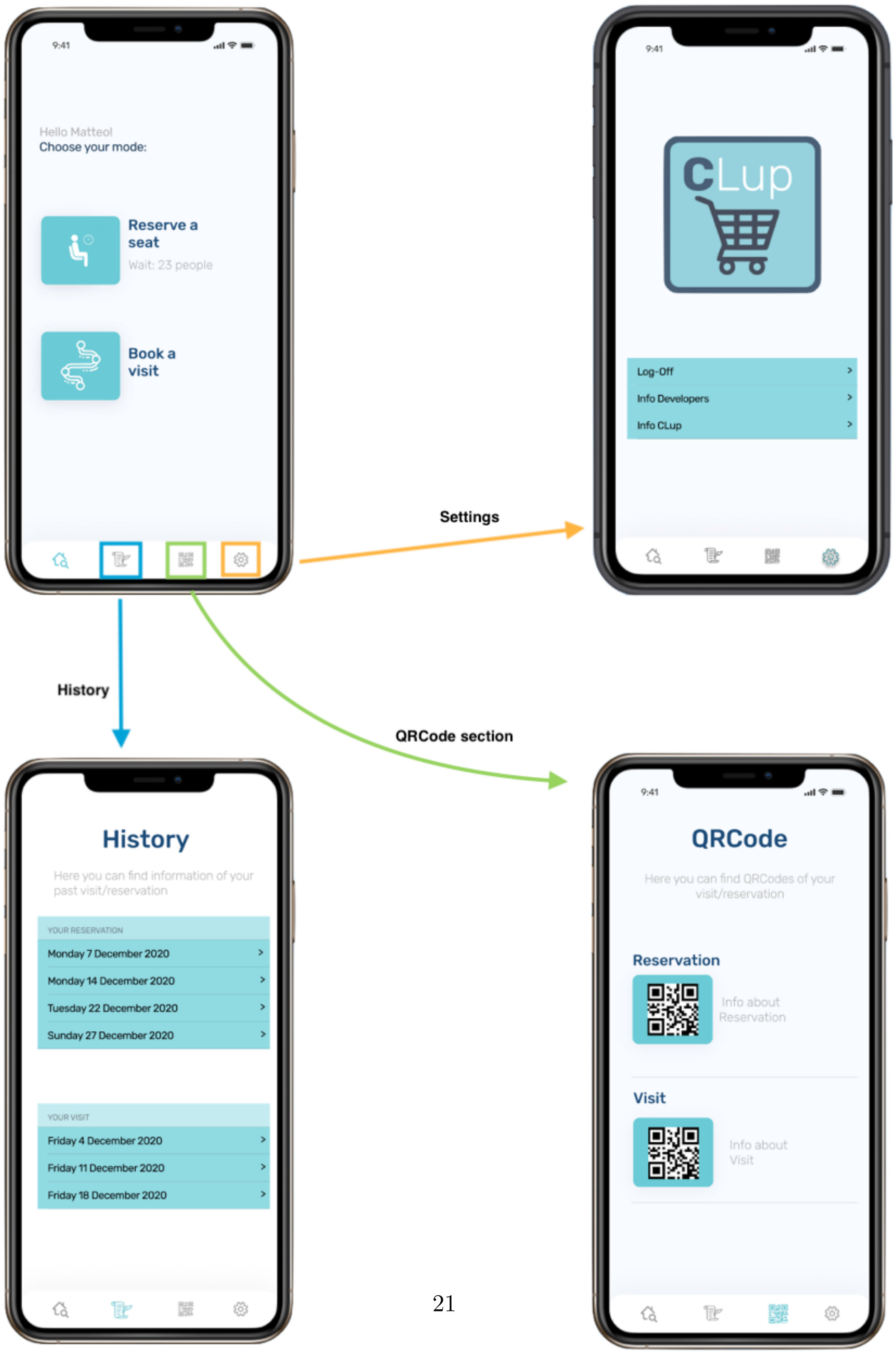


Figure 3.1: Login and registration interface: starting from the first screenshot the user must authenticate or register himself to use CLup. Hypothetically if a User signs up for the first access, he will be already logged in. After procedure the home screen will be displayed.

Figure 3.2: From the home screen it’s possibile to move in the other three app section by selecting them in the Bottom Navigation Bar



Hello Matteo!
Choose your mode:

Wait: 23 people

Here you can find information of your past visit/reservation

Monday 7 December 2020
Monday 14 December 2020
Tuesday 22 December 2020
Sunday 27 December 2020

Friday 4 December 2020
Friday 11 December 2020
Friday 18 December 2020

Here you can find QRCodes of your visit/reservation

Info about Reservation

Info about Visit

Figure 3.3: Procedure needed to book a Visit.

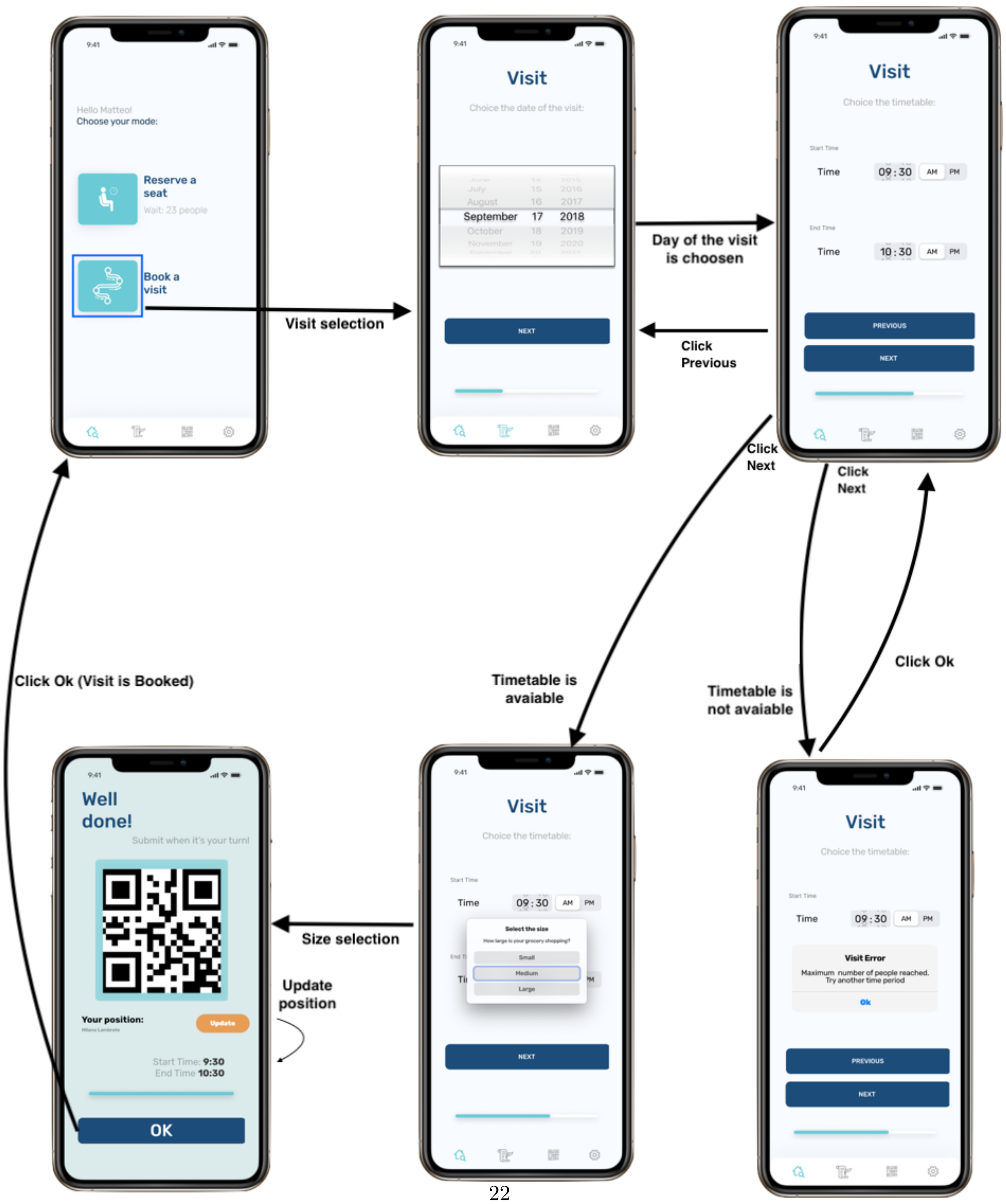


Figure 3.4: Procedure needed to make a Reservation.

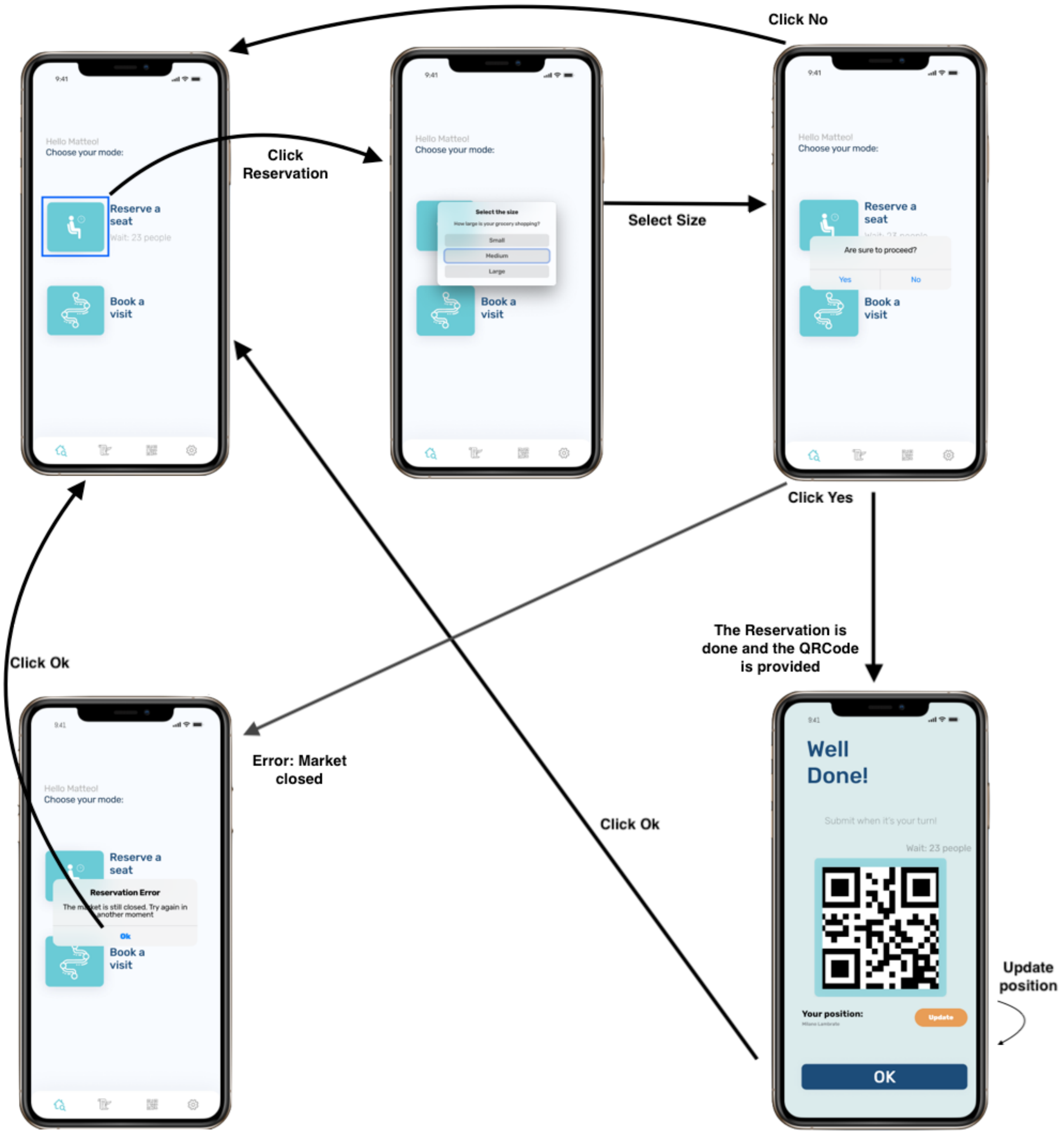
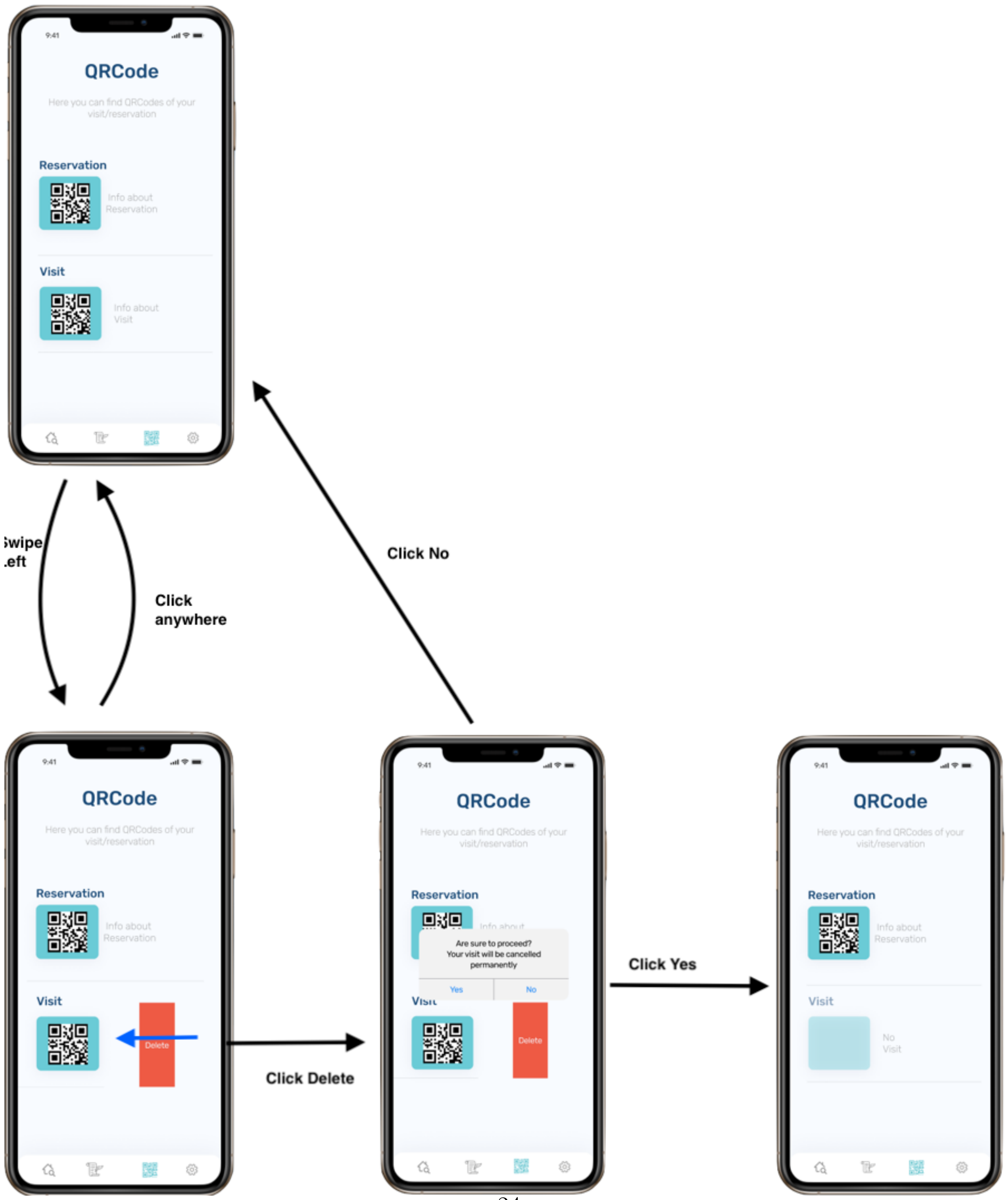


Figure 3.5: The diagrams shows how it's possibile, from the QRCode section, to cancel a Visit. The procedure will be the same also for a Reservaiton cancellation. In particular the following screenshoots illustrate the procedure in a scenario in which a user has booked both Reservation and Visit.



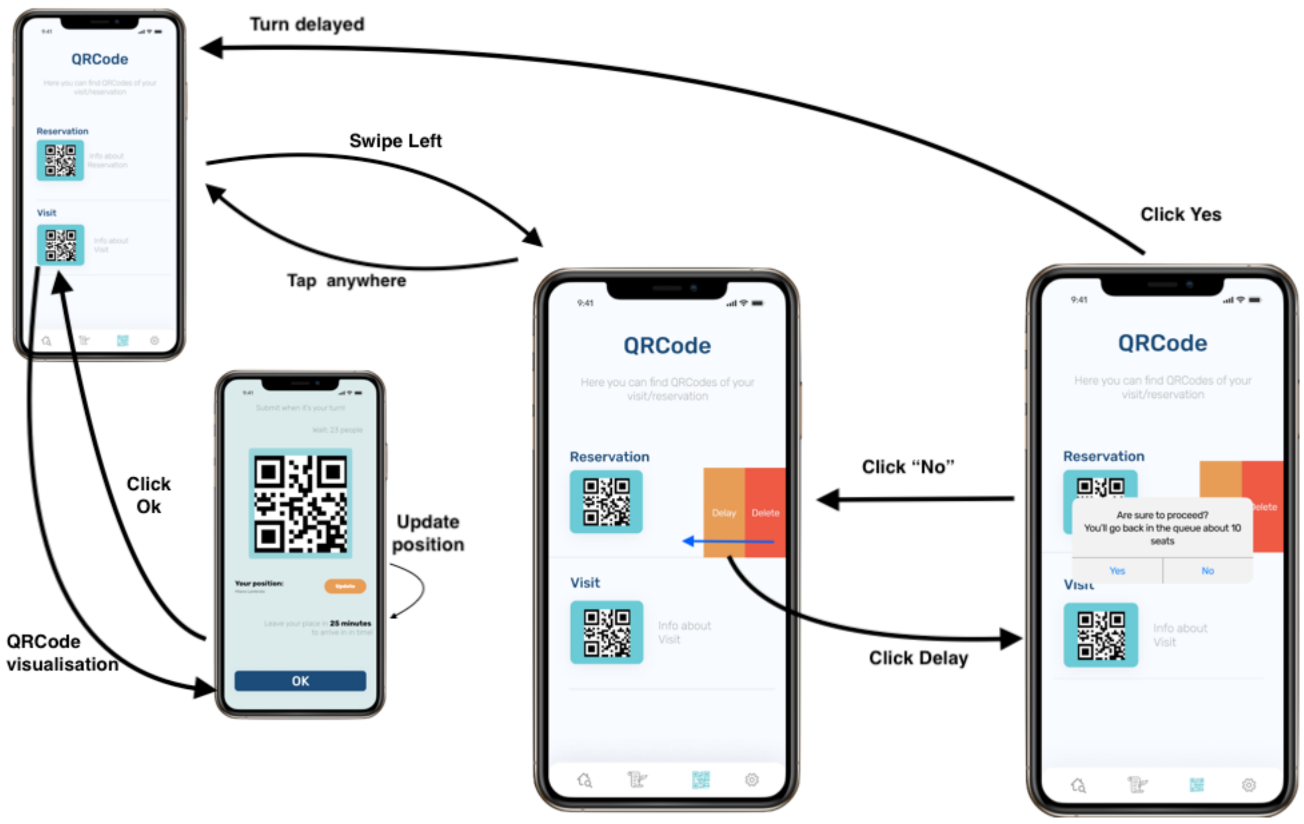
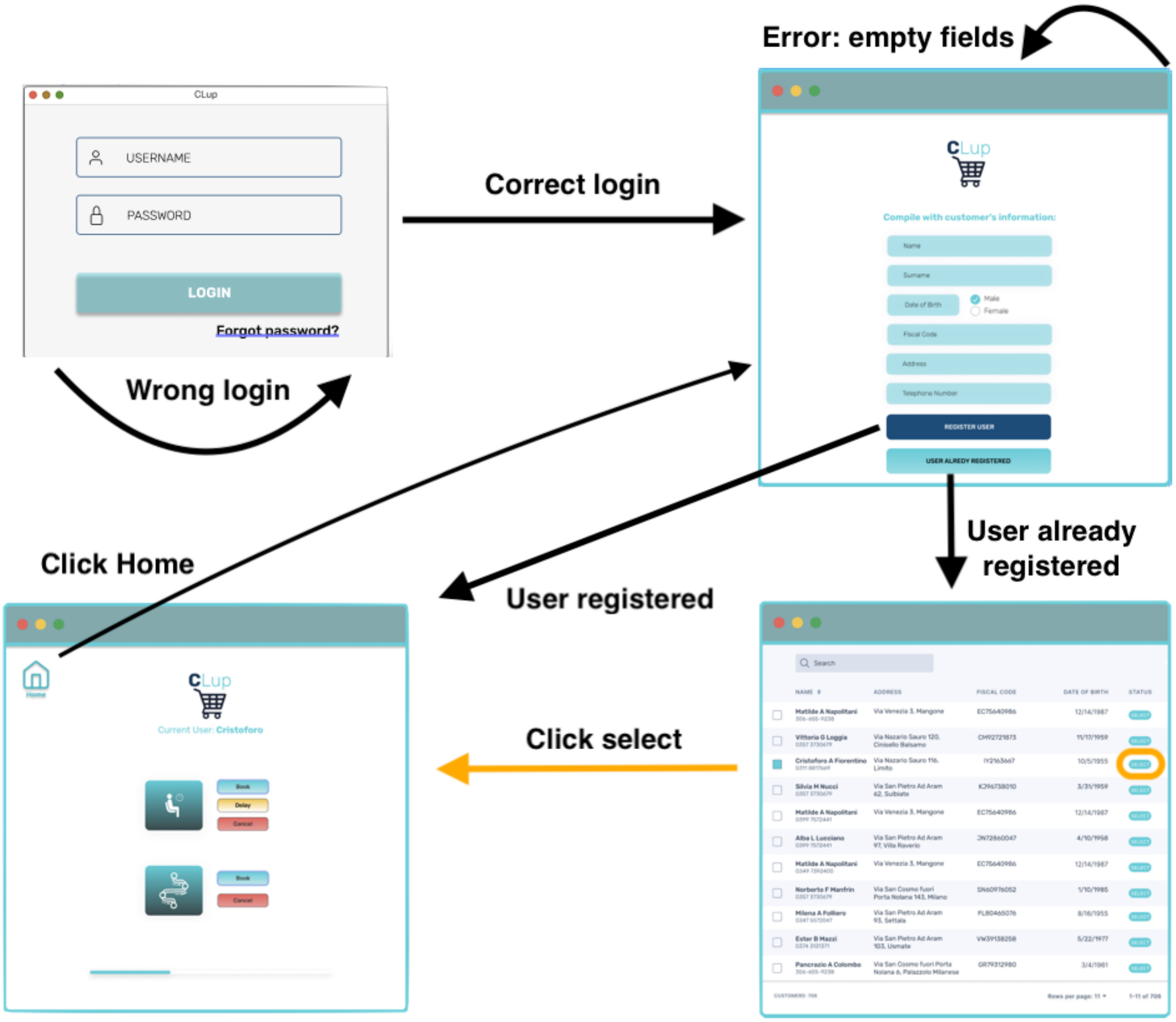


Figure 3.6: The diagrams shows how it's possibile, from the QRCode section, to postpone the own turn in queue for a Reservation. In particular the following screenshoots illustrate the procedure in a scenario in which a user has booked both Reservation and Visit.

3.2 Desktop Interface: CLup Operator



	NAME	ADDRESS	FISCAL CODE	DATE OF BIRTH	STATUS
☐	Matilde A Napolitani 306-655-9238	Via Venezia 3, Mangone	EC75640986	12/14/1987	SELECT
☐	Vittoria G Loggia 0357 3730679	Via Nazario Sauro 120, Cinisello Balsamo	CM92721873	11/17/1959	SELECT
■	Cristoforo A Fiorentino 0311 8817669	Via Nazario Sauro 116, Limito	IY2163667	10/5/1955	SELECT
☐	Silvia M Nucci 0357 3730679	Via San Pietro Ad Aram 62, Sulbiate	KJ96738010	3/31/1959	SELECT
☐	Matilde A Napolitani 0399 7572441	Via Venezia 3, Mangone	EC75640986	12/14/1987	SELECT
☐	Alba L Lucciano 0399 7572441	Via San Pietro Ad Aram 97, Villa Raverio	JN72860047	4/10/1958	SELECT
☐	Matilde A Napolitani 0349 7392400	Via Venezia 3, Mangone	EC75640986	12/14/1987	SELECT
☐	Norberto F Manfrin 0357 3730679	Via San Cosmo fuori Porta Nolana 143, Milano	SN60976052	1/10/1985	SELECT
☐	Milena A Folliero 0347 5572047	Via San Pietro Ad Aram 93, Settala	FL80465076	8/16/1955	SELECT
☐	Ester B Mazzi 0374 3131371	Via San Pietro Ad Aram 103, Usmate	VW39138258	5/22/1977	SELECT
☐	Pancrazio A Colombo 306-655-9238	Via San Cosmo fuori Porta Nolana 6, Palazzolo Milanese	GR79312980	3/4/1961	SELECT

CUSTOMERS: 708 Rows per page: 11 ▾ 1-11 of 708

Figure 3.7: A receptionist must authenticate himself before taking into account the user's request. After this the receptionist is able to select an existing or register a new user in order to satisfy his request.

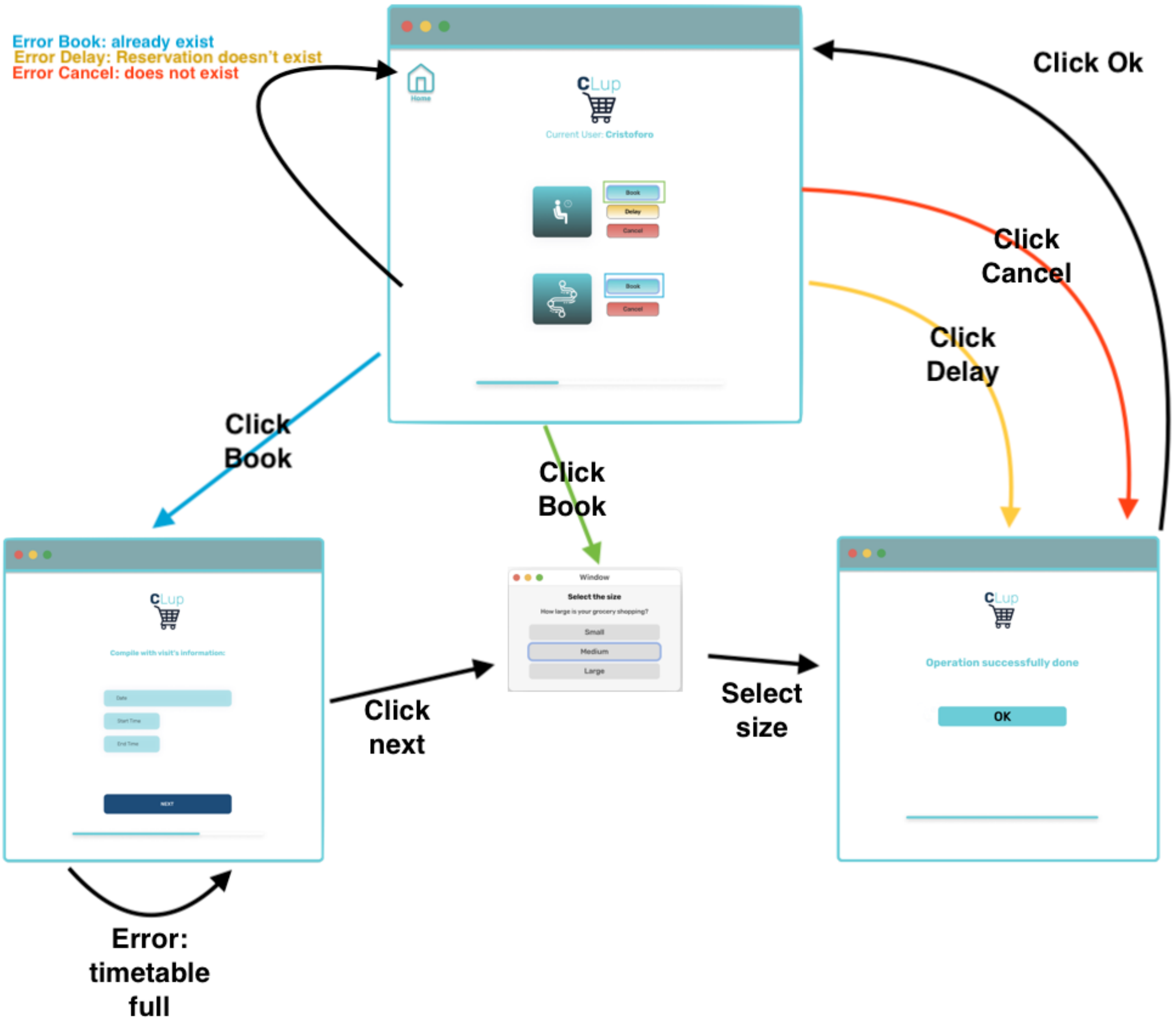


Figure 3.8: The following screenshots illustrate how a receptionist can manage user's request through some steps.

Chapter 4

Requirements Tracebility

requirement of rasd in relation to components

Chapter 5

Implementation, Integration and Test Plan

how to (dividing each parts)

testing how to integratio plan

Chapter 6

Effort Spent

Chapter 7

References