



# POLITECNICO

## MILANO 1863

### Iqueue Project

RASD

Software Engineering for Automation (2022-2023)

Giacomelli Gianluca, 10615105  
Gottardini Andrea, 10617000  
Veronese Niccolò Enrico, 10620278

Professors: Rossi Matteo Giovanni  
Lestingi Livia

# Summary

1.	Introduction .....	2
1.1	Purpose.....	2
1.2	Reference Documents.....	2
1.3	Document Structure .....	3
2	Overall description .....	3
2.1	Product perspective .....	<b>Errore. Il segnalibro non è definito.</b>
2.2	Product functions .....	4
2.2.1	Data collection .....	4
2.2.2	Data analysis .....	5
2.2.3	QR code reading and generating .....	5
2.2.4	Special offer insertion .....	5
2.2.5	GPS system connection .....	5
2.3	User characteristics .....	5
3	Elicitation of requirements.....	5
3.1	Stakeholders, existing systems, documentation .....	<b>Errore. Il segnalibro non è definito.</b>
3.2	Scenarios and use cases .....	7
3.2.1	Scenarios .....	8
3.2.2	Use Cases.....	9
4	Specific requirements .....	13
4.1	External Interface Requirements .....	13
4.1.1	User interfaces .....	13
4.1.2	Customer interfaces .....	14
4.1.3	Shop owner interfaces.....	16
4.1.4	Hardware interfaces.....	18
4.1.5	Software interfaces.....	18
4.1.6	Communication interfaces .....	18
4.2	Functional Requirements.....	18
4.2.1	Requirements.....	19
	Appendix.....	21
1.	Scope .....	21
2.	Definitions, Acronyms, Abbreviations.....	22
3.	Assumptions, dependencies and constraints.....	22

# 1. Introduction

A RASD is a document that aims to present all the requirements of the system to be developed, explaining the domain in which it has to operate and what the application will do in a detailed way. In addition, it identifies with whom the system interacts with a particular focus on the involved stakeholders. A RASD should work as a baseline for the following tasks for the software development, in particular project planning, software evaluation and change control. This document has a wide audience, and hence it has to be written as clearly as possible.

The RASD has been identified considering all the information previously described in the Feasibility study.

## 1.1 Purpose

The main goal of the application Iqueue is to give the customers of small-medium size shops an efficient way to track the queue and the waiting time of those shops so that they can decide when to go to the stores, optimizing in this way their precious time. An additional feature of this app is that shop owners can easily keep track of the daily and even hourly number of customers and, at the same time, they have a useful instrument to advertise their activity and to create a lock-in effect in the clients. Therefore, Iqueue must be an application that can allow both the customers and the owners to register, with different options, and which can relate to a GPS environment such as Google Maps. Consequently, the goals of this project are:

Table 1: List of goals

G1	<b>Customers</b> can view the current queue status for the shop.
G2	<b>Customers</b> can book time slots in advance to visit the shop.
G3	<b>Customers</b> can get discounts or other benefits by using the Iqueue app to visit shops.
G4	<b>Customers</b> can provide feedback and ratings for the shops, helping other users make informed decisions.
G5	<b>Customers</b> can easily search and discover new shops based on their preferences and location.
G6	<b>Shop owners</b> can efficiently manage and organize the incoming flow of customers, improving their overall customer experience.
G7	<b>Shop owners</b> can better forecast and plan their inventory and staffing needs based on the expected footfall of customers.
G8	<b>Shop owners</b> can access analytics and insights on their business performance and customer behaviour, helping them make informed decisions.
G9	<b>Shop owners</b> can improve their brand awareness and visibility by being featured on the app and leveraging the app's marketing capabilities.
G10	<b>Shop owners</b> can offer personalized promotions or deals to customers through the app.
G11	<b>Shop owners</b> can build customer loyalty by offering a seamless and convenient experience through the app.

## 1.2 Reference Documents

- *IEEE 29148-2018* Requirements engineering, the IEEE specification document that “provides details for the construct of well-formed textual requirements, to include characteristics and attributes, in the context of system and software engineering”

- Course slides

## 1.3 Document Structure

This document complies with the SRS standard structure as it is defined in the *IEEE 29148-2018* Requirements engineering, section 9.6. Nevertheless, the order of the contents has been slightly changed in order to facilitate the readers in the reading of this specific RASD. Therefore, the document is divided into 3 main parts:

1. the first part (to which this section belongs) provides an introduction to the system to-be, Iqueue, making clear which are the goals it is required to achieve and in which context it is going to operate.
2. the second part provides a more detailed description of the functions that Iqueue has to implement relating them to the main concepts of the system and the user needs; it also provides the main assumptions under which Iqueue will work properly.
3. the third part treats with elicitation of requirements, presenting a stakeholder analysis, scenarios and use-cases.
4. the fourth part contains the complete requirements of the system, from both the functional and the non-functional points of view.
5. appendix with the scope of the project

It should be remarked that the structure of this document does not follow a logic or temporal order, but whoever is interested in the reading can jump from one section to another, because the purpose of it is to be a reference document.

## 2 Overall description

### 2.1 Product perspective

To describe what our application is expected to do, we employed the following statecharts. Since the main actors involved in the Iqueue application are Customers and Shop Owners, we choose to describe the main processes that they should face.

Once one chose to operate as a Customer, the first passage that the client should perform is to navigate to the “Search Shop” page and then select the desired shop. For that shop, the Iqueue app will give the possibility to book a time slot and, if this latter is available, the app will generate a QR code with the information related to the booking. For a graphical representation of this processes see figure 1 and 2. The other main actor for the Iqueue application is the Shop Owner.

Once logged in the application, he can perform many operations like register a new shop, add products related to his already signed shops, generate advertisements or special discounts. The Iqueue app will automatically implement these updates and it will show the results in the MyShops view of the Shop Owner. Figure 3 represents the pictorial description of the above process.

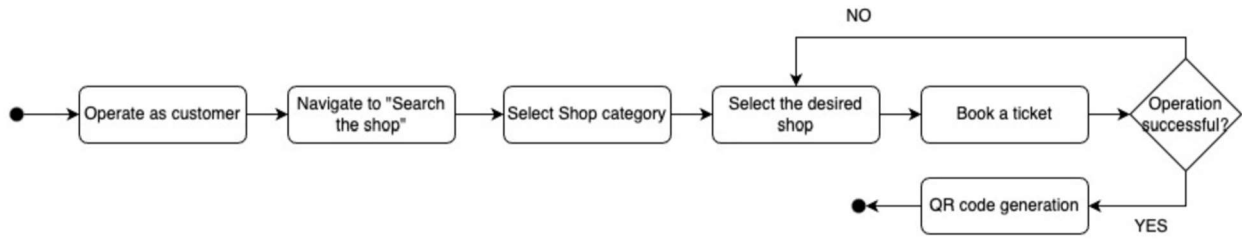


Figure 1: Customer main process flow.

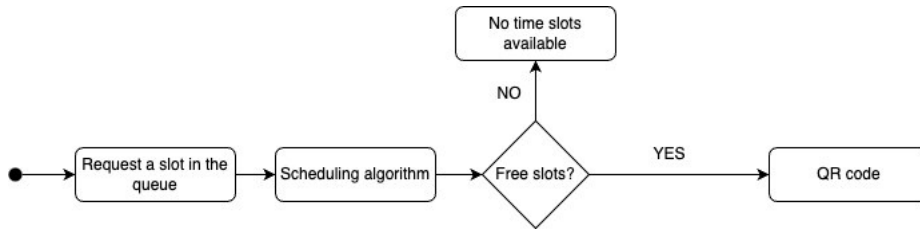


Figure 2: Booking and QR code process flow.

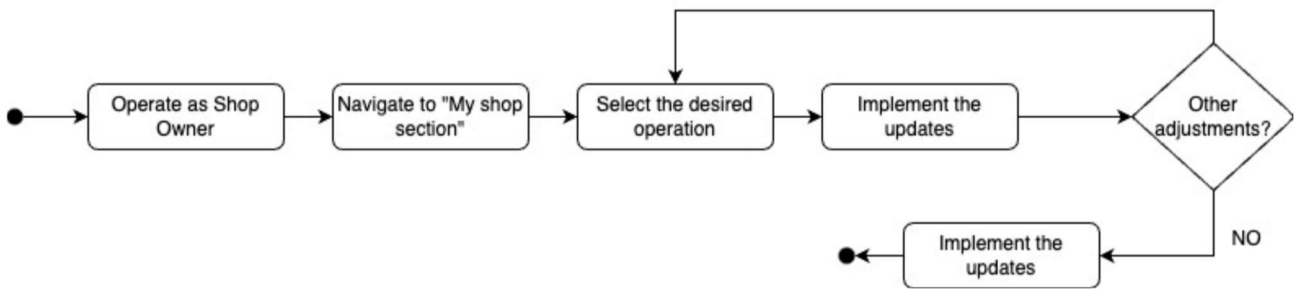


Figure 3: Shop Owner main process flow.

## 2.2 Product functions

In this section, a list of the most important requirements of the system is provided; notice that they are just briefly described since they will be analysed in-depth in the next chapters.

### 2.2.1 Data collection

Iqueue must be able to manage different kinds of data coming from various sources:

1. The customers who will book their presence in the shop through the application: at each customer will correspond a QR code and a position in the waiting list. The customers who have booked their time slot will have the priority over the ones without the application. The costumers without the application will be placed at the end of the queue.
2. The shop owners will insert into the system standard kind of information about the shop they own, but also other kinds of information like, for example, the types of sold products and eventually new products ready to be launched in the market.
3. The inc/dec system: every time a new customer enters the shop without having the application, the shop owner should increment the number of people in queue. In this way, Iqueue will be able to keep track also of the people on the waiting list without the application and jointly with the QR code strategy will be able to detect the number of people in the queue for the shop.

### 2.2.2 Data analysis

The raw data collected by Iqueue must be processed before being delivered to the end user. Therefore, starting with the big volume of information “ingested”, various kinds of analytics are performed to provide a more aggregate version of the data:

1. Associate the value coming from the booking of a certain shop to the correct number of people in the queue in front of the corresponding shop with the application.
2. Associate the value coming from the inc/dec system to a number of people in the queue in front of the corresponding shop without the application.
3. Associate the information from the shop owner to the corresponding class of shops (bakery, minimarket, clothes, ...)
4. Keep track of the number of people in a specific shop during the days, performing eventually some analytics like average, mode and so on.

### 2.2.3 QR code reading and generating

Iqueue generates a QR code once a booking has been done: this QR code must be created so that it links the customer information with the shop and so the shop owner. The application must be able to read the QR code for the customer with to allow the shop owner to verify the customer reservation.

### 2.2.4 Special offer insertion

Iqueue, in order to be economically advantageous also for the shop owners, must have a section in which the owner can insert special offers for the shop product/service so that he can promote the activity. This function will be fundamental also for the customers who will be encouraged to download our application to have this kind of promotion.

### 2.2.5 GPS system connection

Iqueue shall guarantee the connection with a GPS system so that to guide the Customers from their location to the desire Shop and to show the map with the available stores.

## 2.3 User characteristics

With regards to the possible actors of Iqueue, two different main user classes can be identified:

1. **Customers:** they access the system in order to see what the shops with the lowest number of people in the queue are, so that they can save time. In addition, having the Iqueue allows them to see special offers in terms of discounts or fidelity products.
2. **Shop owners:** they access the system in order to keep track of the number of people who enter the shop, depending also on the time. Another feature they can have is to provide special offers to the customers to advertise their activity and to create a lock-in effect in the clients.

## 3 Elicitation of requirements

*Elicitation of requirements* allows detecting the software requirements, in particular: with who or what the application interacts and its boundaries. We remember the fact that in the elicitation of requirements, we face some issues associated with stakeholders such as implicit knowledge and

documentation difficult to obtain, conflicting information and biases. These issues could be partially solved by applying observation by the requirement engineer also keeping into account that unfortunately a probing effect could be generated. These complexities are faced by adopting different approaches and strategies (listening, observing, studying, ...) and combining the results achieved with all of them and specifically being as close as possible to stakeholders and letting stakeholders describe their viewpoints.

### 3.1 Stakeholders, existing systems, documentation

- **Stakeholders**

The main stakeholders involved in this mobile application are all the people with a smartphone who are willing to go in a small/medium-sized shop like bakeries, hair salons, perfumeries and so on. Typically, these kinds of shop are present in the central part of cities so our focus will be on this perimeter. In addition to that, our stakeholders are people who own a small/medium-sized shop and who want to advertise it or have a better idea of the number of clients. An additional feature these stakeholders must have is that they need to know how to use an application and a QR code, so we need to keep particular attention to the simplicity of the user interface in order to avoid to lose a portion of the market (the shops managed by old people who are not familiar with apps, but potentially they are in strategic part of the city like in the centre). Another important stakeholder for our application is the GPS system provider, either Google Maps or Maps, which will allow Iqueue to show the location of the shops.

- **Existing systems**

On the market are present other applications which provide similar functionality compared to our app. However, they are characterized by defects or drawback which we aim at overcoming. The most important ones are:

1. *Qminder*: this application gives a service like ours, in fact it aims at keeping track of the number of people in queue and at giving analytics about the time and performance. However, it is characterized by a considerable higher price with \$429 per month for the base version up to \$1029 per month for the premium one. This cost is justified by the fact that their target are big companies or supermarkets, rather than small/medium ones like in our case. Another difference is that they focus more on clothes shops and in fact they have the possibility of booking also a shop assistant, but this is not the goal of Iqueue.

Source: [www.qminder.com](http://www.qminder.com)

2. *Waitwhile*: this application has some features like Iqueue. In fact, it focuses on the queue management and appointment scheduling of the shops, giving the possibility of book a time slot and thus generating a corresponding QR code. In addition to that also the cost is comparable to our price since the base version is free and the premium one is \$156 per month, which is quite affordable also for our possible clients. However, the main difference between Waitwhile and Iqueue is that our application will also have a interface with a GPS system so that people can see where the interested shops are located, rather than just have the address like in Waitwhile. Even for this application, although in a smaller way compared to Qminder, the focus is on clothes shops.

Source: <http://www.waitwhile.com>

3. *Qwaiting*: this application has a service which is again very similar to our since it keeps track of the number of people in queue by generating a QR code once a booking has been formulated. The possible drawback of this application is the price in fact the base version costs \$89 per month while the premium version costs \$699 per month. In addition, they will create an infrastructure to keep track of the number of people inside every shop so the cost is intended for each location.

Source: <http://www.qwaiting.com>

- **Documentation**

The documentation has been found with a search on Internet. The information regarding the competitors and the stakeholders are present on their website. For what concerns the market according to *MarketsAndMarkets* the global queue management systems in terms of revenue was estimated to be worth \$0.5 billion in 2020. Due to the Covid 2019 pandemic the interest into this field is grown because it is very important to avoid that people stay in queue in front of crowded places. In addition to that, nowadays there is a growing attention into the time management and efficiency, thus shop owners want more and more to give a great experience to their clients. For these reasons, the queue management market is expected to grow at a rate of 4% per year and to reach a value of \$0.6 billion by the end of 2026.

Source: [www.marketsandmarkets.com/Market-Reports/queue-management-system-market-23961354.html](http://www.marketsandmarkets.com/Market-Reports/queue-management-system-market-23961354.html)

## 3.2 Scenarios and use cases

Scenarios are an important tool with which we can summarize situations we derive from observations, interviews and document analysis. According to M. Carrol, a scenario is “*a narrative description of what people do and experience as they try to make use of computer systems and applications*” (M. Carrol, Scenario-based Design, Wiley, 1995).

To build scenarios, we ask ourselves the following questions:

- Which user groups are supported by the system to perform their work?
- What are the primary tasks that the system needs to perform?
- What data will the actor create, store, change, remove or add to the system?
- What external changes does the system need to know about?
- What changes or events will the actor of the system need to be informed about?

Scenarios have not to be identified only through questions but relying also on task observation if the system already exists and on talking with end users, not just to the software contractor.

Scenarios are concrete, specific and describe a single feature of the system, but the application has to be exhaustive with respect to many situations not only some specific ones. Hence, to abstract from details and specificities, we consider *use cases*, a generalization derived from scenarios. We choose to enclose in use cases the following aspects:



- Participating actors
- Entry Condition: condition to be verified to enter the use case.
- Flow of Events: By identifying in it which actors take the action, we can detect which parts have to be realized by the application and which ones have to be only observed.
- Exit Condition: it describes the system states after the use case.
- Exceptions
- Special Requirements: constraints, nonfunctional requirements.

### 3.2.1 Scenarios

In this section, we will analyze some important scenarios and from them, we will derive the use cases. First, we provide a detailed list of the Actors:

- Iqueue
- Client: someone who will visit the shop.
- GPS system: external system (e.g. Google Maps)
- Shop Owner: who beholds the shop.

**Scenario name: Difficulties in finding the pastries for the party.** Joanna is a girl who will join a university party at 20:00. Every guest of the party has to bring some food (e.g. drinks, pizzas, pretzels, ...) and Joanna proposes herself for the pastries. Joanna is very strict on her schedule because an exam at her university has terminated at 18.30 later than expected. He has planned to visit her favourite bakery Scaringi, but unexpectedly when she arrives, she sadly discovers that a long queue is outside the shop. As a consequence, she would arrive later at the party. To avoid this, Joanna opens Iqueue and scans the available bakeries around her. She found out that another renovated bakery is open and that its queue is not very long. In addition, this bakery also has her preferred kind of pastries at hand. Hence, Joanna decides to book a slot at this bakery compatible with her schedule and choose to be guided here, hence Iqueue directs her on a third-party GPS system. Thanks to this, Joanna arrives at the party on time and her pastries will be appreciated by all the guests. Joanna proposes Iqueue app to her friends because she saw the potential of it for busy people as her.

**Scenario name: Difficulties in counting the visitors.** Helena beholds a very renovated hairdresser saloon that after gaining success, started to receive many visits daily. The shop team and she find difficulties with the requests to the suppliers for what concerns primary products to perform their activity, such as lacquer or balms. She thinks that by knowing the exact number of shop clients, she could better schedule the supply requests. After discovering Iqueue, Helena can now employ its functionality related to the visitor count: when a client visits the shop booking from Iqueue, the count is performed automatically after scanning the client QR code whereas, for non-Iqueue clients, she has to increment the count by herself. Based on this, she is now able to hardly ever miss a product to carry out her activity because she can access the client counter on Iqueue whenever she wants.

**Scenario name: Poor number of clients.** Mario owns a small bakery in a strategic position in the city but the shop has open since very few weeks so the number of customers is still low. He would like to have something to advertise his activity and his products without spending too much. For this reason, he downloads the Iqueue application on his mobile phone via Play Store or App Store. He registers his grocery from a specific menu for the shops, accepting to give the position of the shop to Iqueue. Then, he inserts his shop logo and some pictures. Then he opens another tab and writes the special offers for the clients who will come to his bakery with the Iqueue app. He hopes to increase the number of

customers rapidly. The Iqueue app is now able to show the position of the bakery on the map, including the logo and the special offers.

**Scenario name: New in town.** Laura has recently moved from her original city to a new town in a different country. She would like to see what the shops near his new house are and especially wants to see whether they are crowded or not in order to save time. She is interested in small/medium shops like bakeries or perfumeries, and she would like also to know some of the promotions offered by them. Thus, she downloads Iqueue from the App store or Play Store on her phone. She registers as client on the app, and she accept the fact that the app uses its position. Iqueue is able now to track the position of Laura. From a menu she selects the category of activity she is interested in (e.g. bakery). After that passage Iqueue shows her the map with the closest bakeries and with the number of people in queue in that moment. Laura is particularly interested in one specific bakery and thus she clicks on the corresponding icon. Iqueue opens a menu with the special offers of the shop and again the number of people in queue.

### 3.2.2 Use Cases

From the scenarios are now identified the following use cases. To better understand them, they are matched with sequence diagrams representing the flow of events. Note that a more detailed description of the features of the above sequence diagram is presented in the UML requirement document.

**Use Case name:** Book (derived from scenario *Difficulties in finding the pastries for the party*)

- Participating actors: Client, Shop Owner, Iqueue, GPS system
- Entry Condition: A long queue outside a shop.
- Flow of Events:
  1. The Client selects a certain type of shops (bakeries in the scenario).
  2. Iqueue show on his map a list of the available shops of the indicated type with their queues.
  3. The Client selects a shop.
  4. Iqueue shows the available products, time slots and special offers.
  5. The Client scans the products and select a free time slot.
  6. Iqueue notifies the Shop Owner of the time slot booking.
  7. The Client chooses to be guided to the shop.
  8. Iqueue redirects the Client to a third-party GPS system and provides to it the shop address.
- Exit Condition: The use case terminates when the Client is guided by the GPS system. The selected time slot is now unavailable to any other Client. The client receives the QR code with the booked time slot.
- Special Requirements: The queue of a shop has not to exceed in time the opening hours of the shop, otherwise the time slot is unavailable. The Client must have a GPS system on his device.

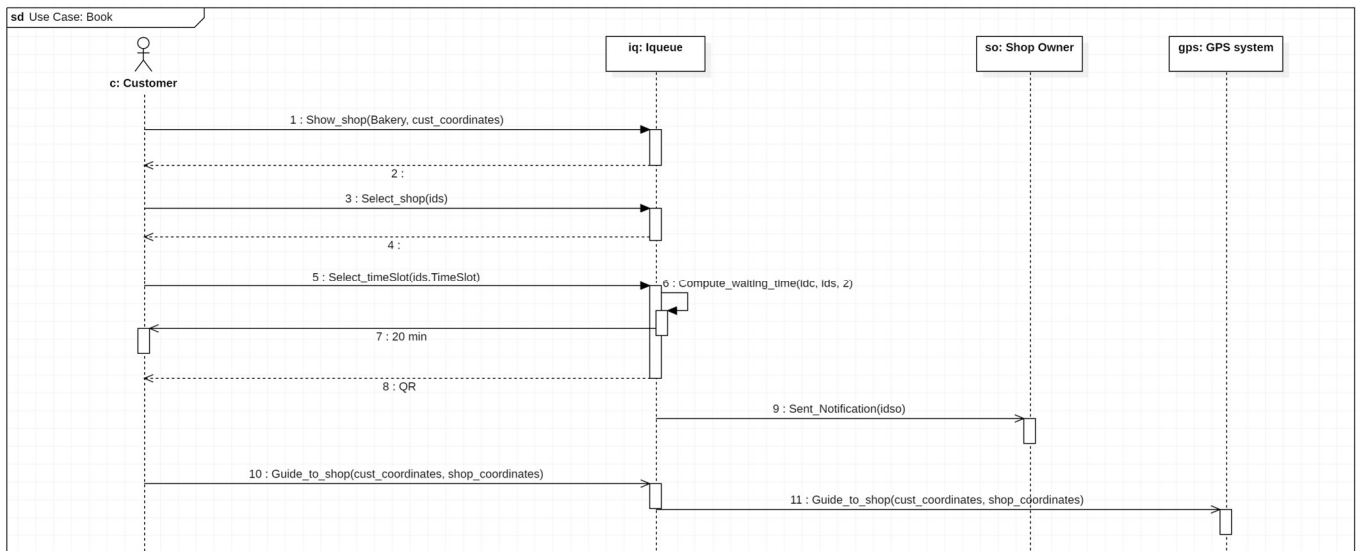


Figure 4: Book sequence diagram

**Use Case name:** Count clients (derived from scenario *Difficulties in counting the visitors*)

- Participating actors: Client, Shop Owner, Iqueue
- Entry Condition: Client enters in a shop.
- Flow of Events:
  1. Shop Owner asks the Client if he has done a reservation with Iqueue.
  2. The Shop Owner scans the client QR code.
  3. Iqueue increments the counter client for the shop of the relative shop opening.
- Exit Condition: The use case terminates after the increment of the client counter. Shop Owner can consult the client counter value on different shop opening accessing Iqueue.
- Exceptions: If the Client has not done his registration on Iqueue, the Shop Owner increments the client counter on Iqueue by himself with the inc/dec system.

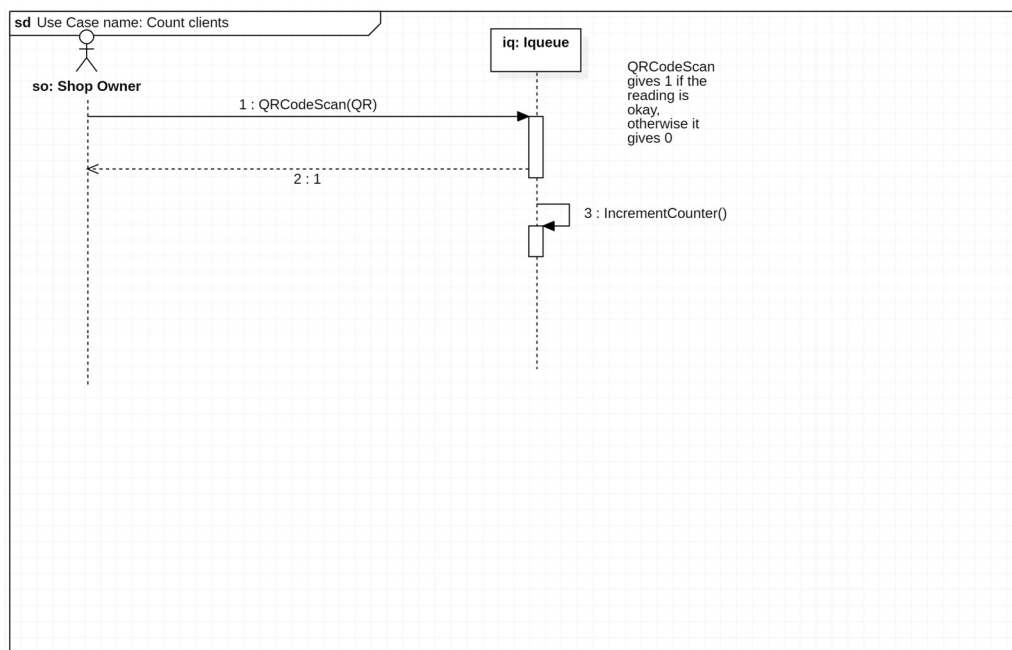


Figure 5: Count clients sequence diagram

**Use Case name:** Register Shop(derived from scenario *Poor number of clients*)

- Participating actors: Shop Owner, Iqueue
- Entry Condition: Shop Owner has a shop.
- Flow of Events:
  1. Shop Owner starts register Shop procedure on Iqueue.
  2. Iqueue asks shop position to the Shop Owner.
  3. Shop Owner inserts the shop position.
  4. Iqueue asks shop logo and/or pictures to the Shop Owner.
  5. Shop Owner inserts the shop logo and/or pictures (optional).
  6. Iqueue asks Shop Owner to insert products of his shop.
  7. Shop Owner inserts the products.
  8. Iqueue asks Shop Owner to insert special offers on products.
  9. Shop Owner inserts the special offers on products (optional).
- Exit Condition: The use case terminates after the product discounts entering. Shop registration is now terminated, and the shop is now visible on the Iqueue map with its products.
- Exceptions: The insertion of shop position and products is mandatory to conclude the shop registration on Iqueue; The Shop Owner could not insert immediately pictures, the logo of his shop and the special offers on products to register his shop. They could be inserted in a second moment.

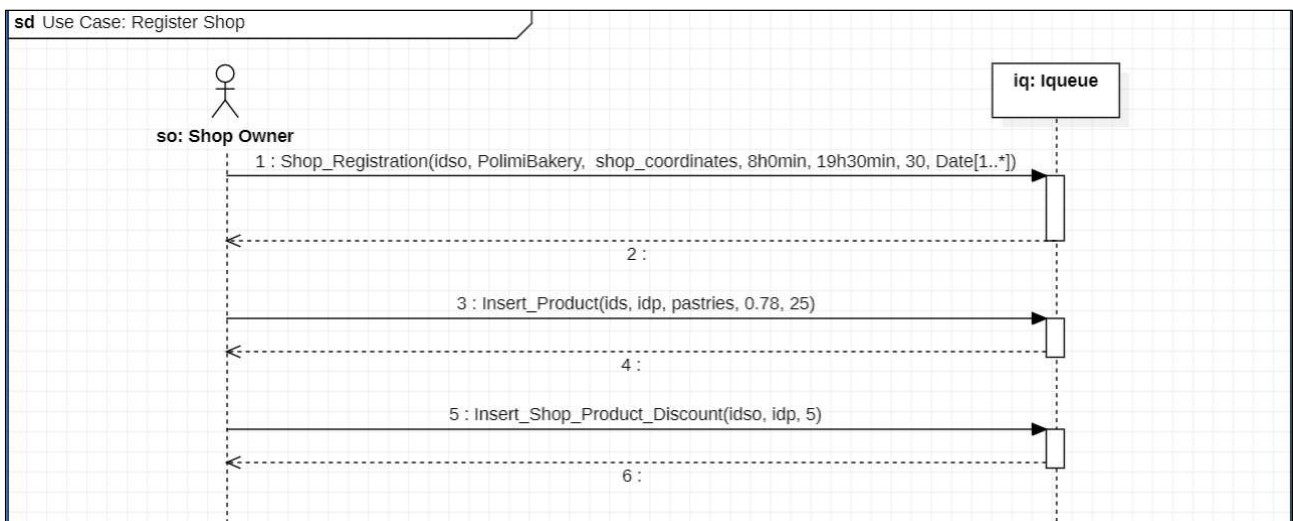


Figure 6: Register Shop sequence diagram

**Use Case name:** Search(derived from scenario *New in town*)

- Participating actors: Client, Iqueue
- Entry Condition: Client likes to know shops in his surroundings.
- Flow of Events:
  1. Client allows Iqueue to know his position.
  2. Client selects a certain type of shops.
  3. Iqueue show on his map a list of the available shops of the indicated type with their queues.
  4. Client selects a maximum distance from the shops.
  5. Iqueue shows the shops accordingly with the selected distance.

6. The Client selects a shop.
  7. Iqueue shows the available products and special offers.
- Exit Condition: The use case terminates after the Iqueue shows the products and special offers. Client could now book a time slot or simply exit the application.
  - Special Requirements: The Client must have a device which is able to provide the position to the Iqueue.

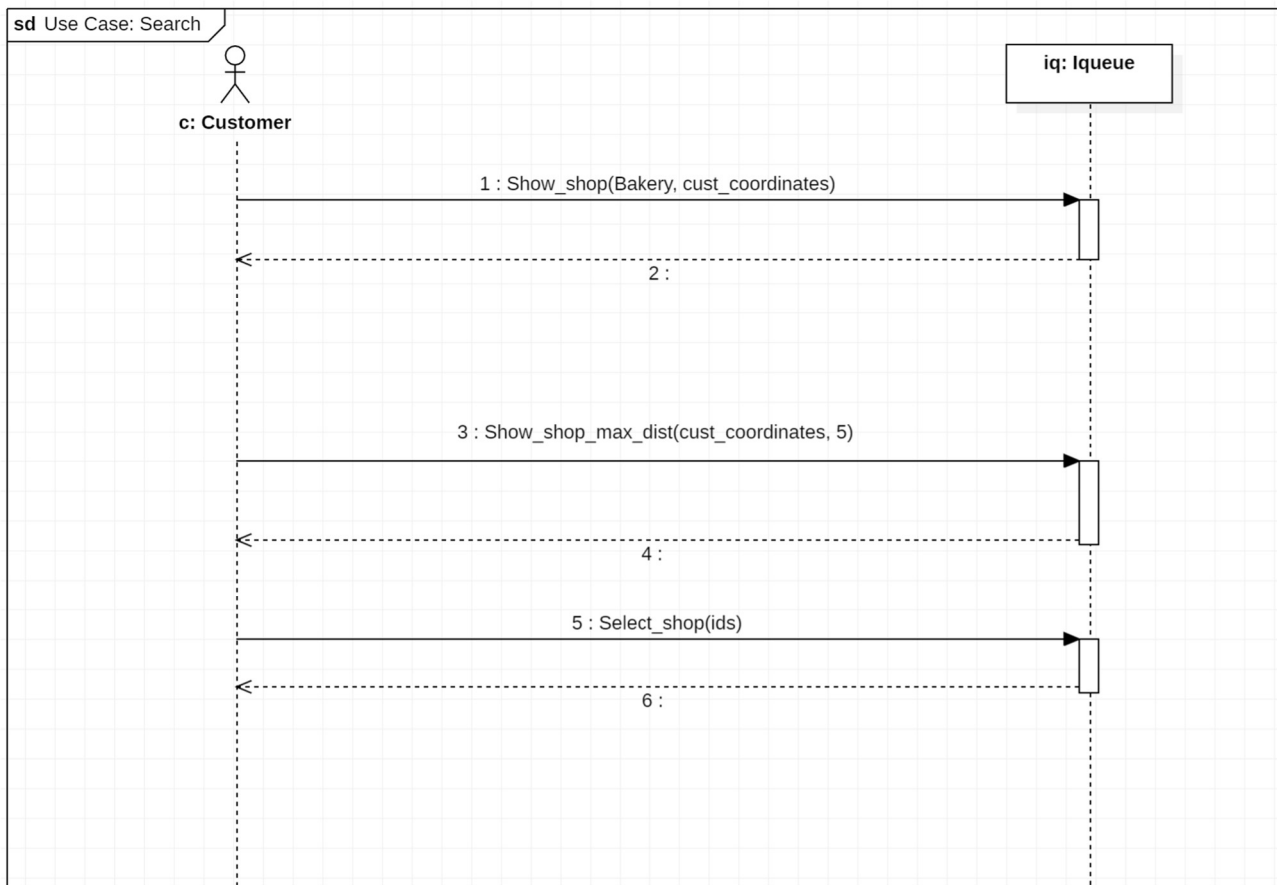


Figure 7: Search sequence diagram

## 4 Specific requirements

### 4.1 External Interface Requirements

#### 4.1.1 User interfaces

Iqueue is provided to the users, namely customers and shop owners, as an application, accessible from web browser. Iqueue is not given with a CLI, Command Line Interface but only with a GUI Graphical User Interface. The rest of this subsection contains some mockups that show an example of suitable user interface for Iqueue.

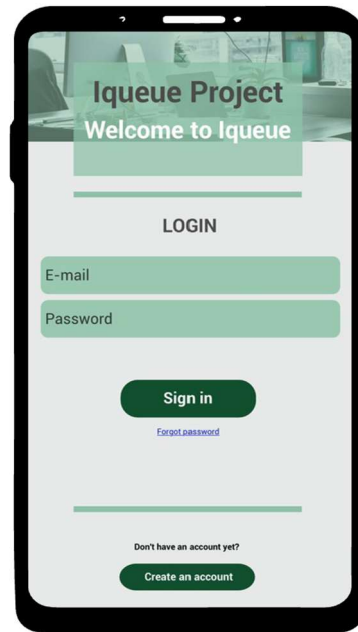


Figure 8: Initial page of the system

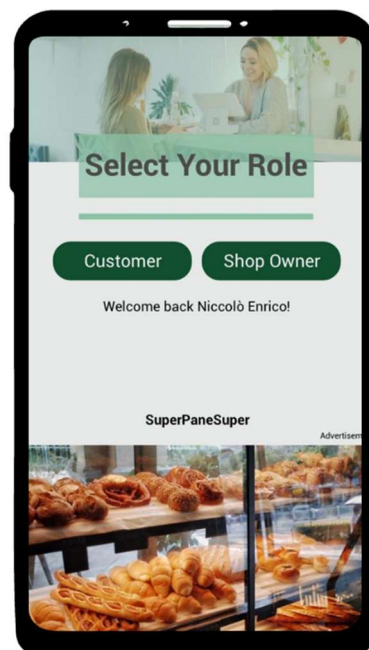


Figure 9: Page in which the user selects whether to act as a customer or a shop owner. It is also possible to see the shops which have been advertised.

### 4.1.2 Customer interfaces

This subsection shows some mockups of Customer graphical interfaces.

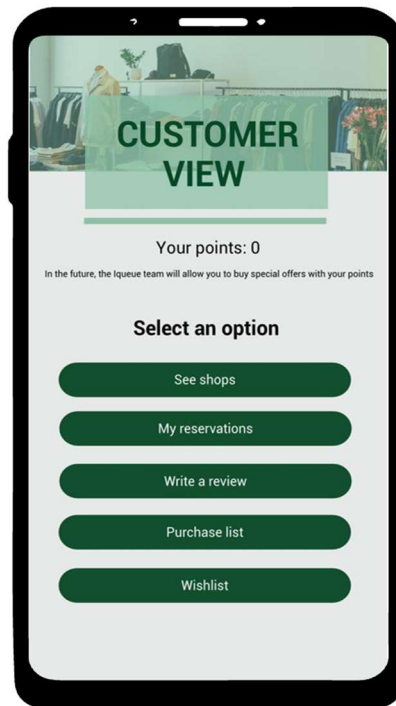


Figure 10: Customer view: page in which the Customer can select to see the Shops logged in lqueue, see its reservations, write a review for the Shop, see its purchase list or its Wishlist On top of that, the view shows also the points earned by the Customer which can be used to buy special offers.

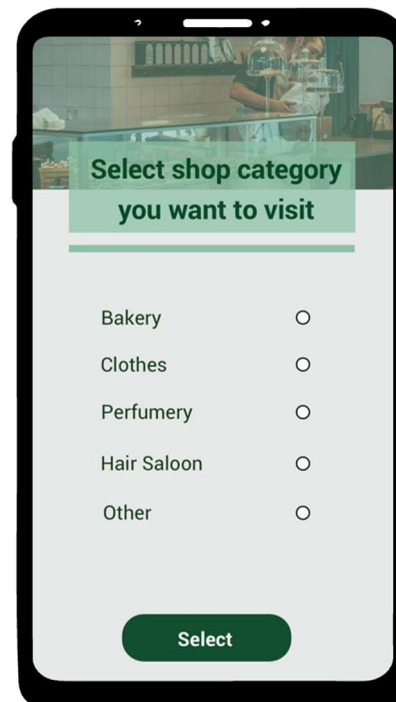


Figure 11: Page in which the Customer selects the shop category he wants to visit

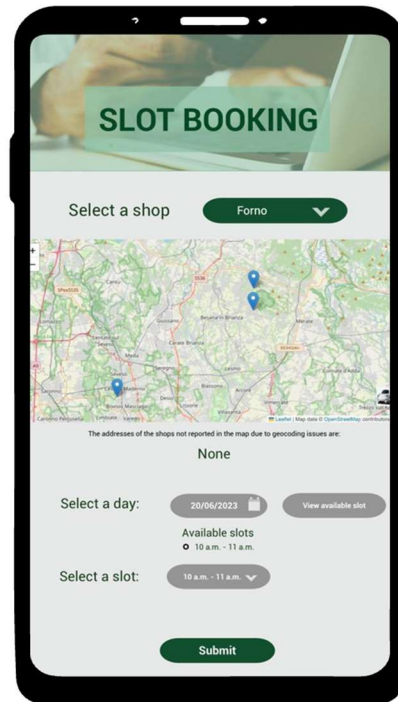


Figure 12: Page that allows the Customer to select the desired Shop, based on the category selected before, shown in a Map interface. In this page the client can also book its corresponding day and time slot.

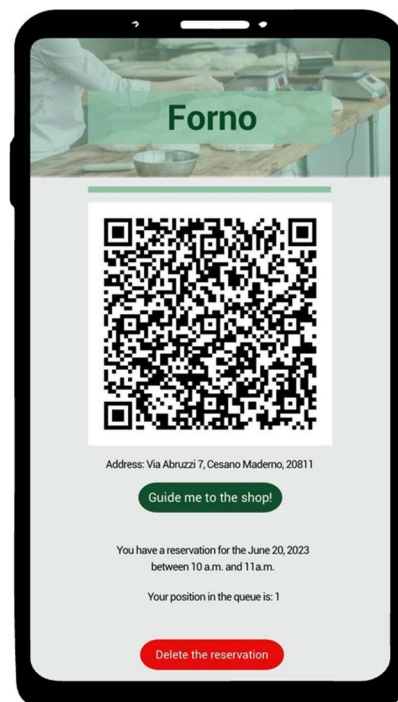


Figure 13: Page that allows the customer to see the QR code corresponding to his reservation. Once pushed the button “Guide me to the shop” the Iqueue app will open the GPS system to guide the customer to the desired location.



### 4.1.3 Shop owner interfaces

In this subsection, some mockups of the Shop owner interfaces are present.

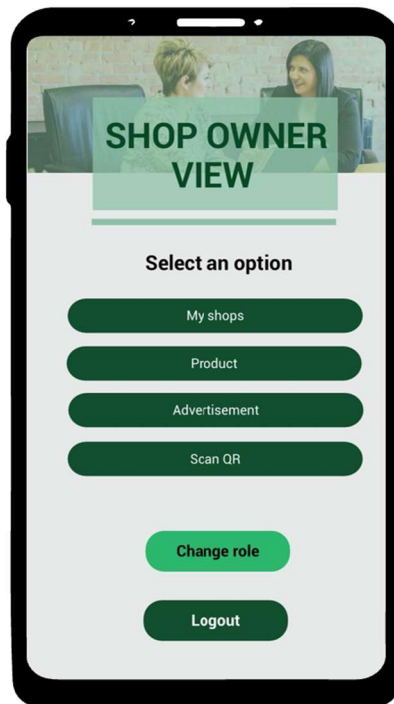


Figure 14: Page that allows the Shop Owner to choose what type of information insert into the Iqueue app. The possible choices could be: Shop, Product, Advertisement or Scan the QR code of an upcoming client.

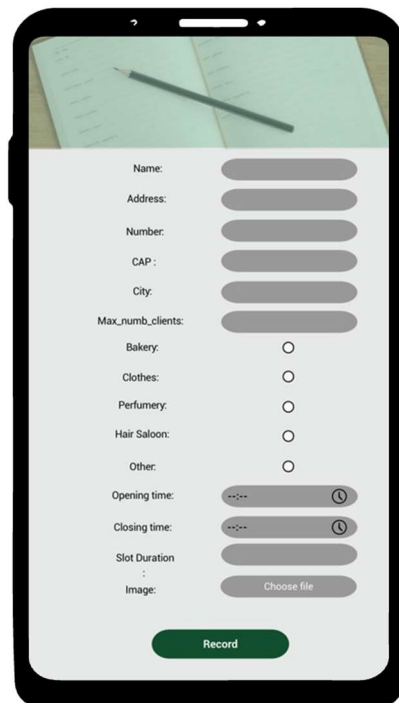


Figure 15: Page that allows the Shop Owner to insert the shop data

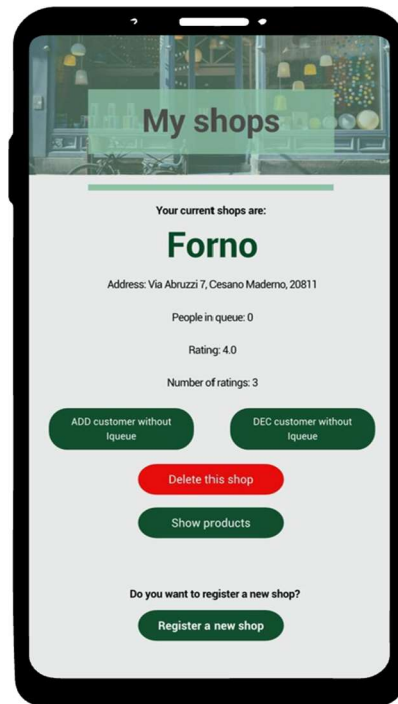


Figure 16: Page that allows the Shop Owner to see its current registered Shops. He can see the current people in queue in front of the Shop and with two buttons he can also add or remove manually customers without the Iqueue app. He can also see the products related to the Shop or eventually remove it from the app.

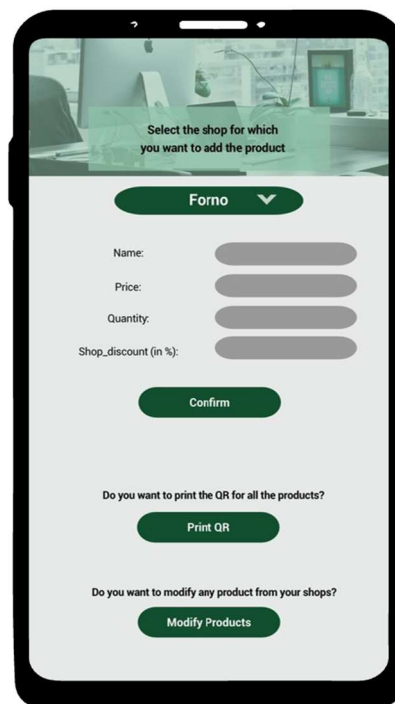


Figure 17: Page that allows the Shop Owner to add products related to his Shop.

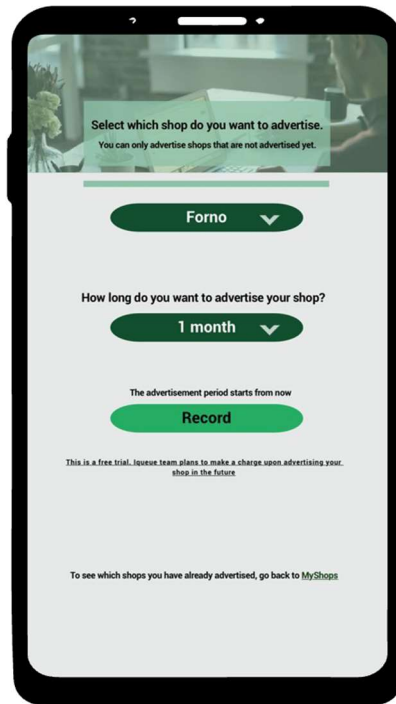


Figure 18: Page that allows the Shop Owner to generate an advertisement for his Shop. The advertisement will be shown in the initial user page of the Iqueue app.

#### 4.1.4 Hardware interfaces

Since Iqueue is to be implemented as a webapp, every user can access it through the device they prefer, that is personal computers, smartphones, tablets . . . and the only requirement for the app is to be responsive (make the website scale properly to different devices' sizes). Every device of this kind suffices to achieve the goals. To read the QR code, shop owner shall have proper technologies (e.g. phone camera, Google Lens,...).

#### 4.1.5 Software interfaces

The following software interfaces are required to make Iqueue work properly:

- 1 Every user's device must have browser (e.g. Google Chrome, Firefox, Opera, ...) installed on it through which the user can access the app; no other software requirements are requested for these kinds of devices;
- 2 Every user's device need a GPS system such as Google Maps or Maps so that Iqueue can interface with it furnishing additional functionalities.

#### 4.1.6 Communication interfaces

For what concerns the communication interfaces, Iqueue uses the HTTP protocol at the application layer (layer 7 of the ISO OSI stack) to exchange information, access GPS systems (e.g. Google Maps) and manage QR code communications.

### 4.2 Functional Requirements

Iqueue allows its users to perform many tasks and so establish targeted requirements to be accomplish is pivotal in the application development. The main system requirements are provided along with a

summary of the possible situations in which Iqueue is involved and used. In this paragraph, a first list of all the requirements of the system is given.

#### 4.2.1 Requirements

Table 2: List of requirements

R1	Iqueue must allow the customers to see the number of people in the queue at a specific shop
R2	Iqueue shall allow the shop owners to insert the data of their activity
R3	Iqueue shall allow the shop owners to insert special offers for their shop
R4	Iqueue must read the QR code of the clients with the app
R5	The app shall allow customers to track their rewards
R6	Iqueue must allow a customer to book a time slot on the app to visit the shop
R7	Iqueue must keep track of the customer position
R8	Iqueue allow the shop owner to associate a mean time for every person in the queue.
R9	Iqueue calculates the waiting time of a customer, based on the mean time for every person in the queue.
R10	The app shall allow customers to create and save shopping lists or wish lists.
R11	The app shall show the customer's purchase history and receipts
R12	The app will be able to divide the registered facilities into categories
R13	The app will permit the shop owner to insert their products/services
R14	The app will enable shop owners to insert the price of their products/services
R15	The app will enable shop owners to modify the price of their products/services
R16	Iqueue must be able to generate the client QR code
R17	Iqueue must show the available time slots of a shop
R18	Iqueue must permit a customer to delete his booked time slot
R19	The app will show the registered shops on a map/list
R20	Iqueue shall allow the shop owners to modify the data of their activity
R21	Iqueue shall permit the shop owners to delete the data of their activity
R22	Iqueue shall allow the shop owners to modify special offers for their shop
R23	Iqueue shall allow the shop owners to delete special offers for their shop
R24	The app must be able to track customer rewards and loyalty points
R25	Iqueue must allow user registration
R26	Iqueue must allow the user to operate as customer or as shop owner
R27	Iqueue must allow log-in as customer
R28	Iqueue must allow log-in as shop owner
R29	Iqueue must allow log-out
R30	The app will enable shop owners to remove their products/services
R31	The app will enable shop owners to count their available products/services
R32	Iqueue will be able to realize to queue counting by means of the QR codes
R33	Iqueue will allow the shop owner to manually increment or decrement the queue counter

R34	Iqueue shall allow the customer to select the category of the shops he is interested to visit
R35	Iqueue shall allow the customer to select a specific shop
R36	Iqueue shall allow the customer to select a specific shop product/service
R37	Iqueue establishes counters to keep track of the client visits during the shops opening hours.
R38	The app shall allow the shop owner to advertise their shop
R39	The app shows the user of the interested queue variation
R40	Iqueue shall give the shop address to a third-party GPS system to guide the client to the shop
R41	Iqueue will allow the client to provide feedback and rating to the shops

In this first development of the application, we initially assume that a client can only buy a product/service in a presence when he arrives at a shop: further improvements to the app will include the possibility to acquire a product/service through a credit card directly on the app.

# Appendix

## 1. Scope

Iqueue is a software system that has to work in a World where the following phenomena occur:

Table 3: List of World phenomena

WP <sub>1</sub>	A <b>customer</b> enters a shop
WP <sub>2</sub>	A <b>customer</b> joins a queue at a shop
WP <sub>3</sub>	A <b>shop experiences</b> high demand and long queues during peak hours
WP <sub>4</sub>	A <b>shop experiences</b> low foot traffic and sales during off-peak hours
WP <sub>5</sub>	A <b>shop owner</b> adjusts pricing or sales strategies to attract more customers
WP <sub>6</sub>	A <b>shop owner</b> restocks inventory based on sales data and demand forecasts
WP <sub>7</sub>	A <b>customer</b> provides feedback to a shop owner about their experience
WP <sub>8</sub>	A <b>shop owner</b> updates their store layout or design to improve customer flow and experience
WP <sub>9</sub>	A <b>shop owner</b> launches a marketing campaign to increase brand awareness and attract new customers
WP <sub>10</sub>	A <b>shop owner</b> hires or trains new staff members to improve customer service and efficiency.

*Shared phenomena* are what is perceived by the machine from the external world.

The shared phenomena, which are the intersection between the World phenomena  $W$  and the Machine phenomena, are:

Table 4: List of shared phenomena

SP <sub>1</sub>	A customer registers an account on the app
SP <sub>2</sub>	A customer views nearby shops on the app based on their current location
SP <sub>3</sub>	A customer selects a shop on the app to visit
SP <sub>4</sub>	The app shows the estimated waiting time for the selected shop
SP <sub>5</sub>	A customer books a time slot in advance on the app to visit the shop
SP <sub>6</sub>	The app sends a notification to the customer when their turn in the queue is approaching
SP <sub>7</sub>	A customer enters the shop and checks-in on the app
SP <sub>8</sub>	The app updates the estimated waiting time based on the customer's check-in
SP <sub>9</sub>	A customer cancels their booking on the app
SP <sub>10</sub>	A customer provides feedback and ratings for the shop on the app
SP <sub>11</sub>	The app shows recommended shops to the customer based on their previous visits and ratings
SP <sub>12</sub>	A customer redeems rewards and offers on the app when making a purchase
SP <sub>13</sub>	The app allows customers to track their rewards and loyalty points
SP <sub>14</sub>	A customer views the shop's menu or product catalog on the app
SP <sub>15</sub>	The app allows customers to place orders for pickup or delivery from the shop
SP <sub>16</sub>	A customer pays for their order using the app

SP17	The app shows the customer's purchase history and receipts
SP18	The app suggests new shops or products to the customer based on their preferences and activity on the app
SP19	A customer contacts the shop directly through the app for inquiries or support
SP20	The app provides live chat or messaging support to the customer for assistance
SP21	The app allows customers to create and save shopping lists or wish lists.
SP22	The app provides a map with the shops.
SP23	The app shows the shop's hours of operation, contact information, and other details
SP24	The app allows customers to report issues or problems with the shop or their experience
SP25	A customer shares their experience or purchase on social media through the app
SP26	The app allows customers to connect and follow their favorite shops or brands for updates and promotions
SP27	The app provides personalized recommendations to the customer based on their browsing and shopping history
SP28	A customer earns badges or achievements on the app for frequent visits or purchases
SP29	The app offers exclusive discounts or promotions to customers who refer their friends to use the app
SP30	The app allows customers to easily switch between different languages or currencies based on their preferences.

## 2. Definitions, Acronyms, Abbreviations

The World is the portion of the real world affected by the machine. Michael Jackson. 1995. The world and the machine.

## 3. Assumptions, dependencies and constraints

Table 5: List of domain assumptions

D1	The customers with the application own at least one mobile phone with a screen and an internet connection.
D2	The customers without the application own at least one mobile phone with internet connection and camera.
D3	The shop owners own at least one mobile phone with an internet connection and camera.
D4	The QR code reading works correctly.
D5	The QR code generating/ticket system works correctly so it gives precise information about the number of people without Iqueue who are waiting.
D6	All the data the shop owner insert is correct.