Immagine che contiene logo

Descrizione generata automaticamente

Iqueue Project

RASD

Software Engineering for Automation (2022-2023)

Giacomelli Gianluca, 10615105 Professors: Rossi Matteo Giovanni

Gottardini Andrea, Codice persona Lestingi Livia

Veronese Niccolò Enrico, 10620278

Summary

[1 Introduction 3](#_Toc131174259)

[1.1 Purpose 3](#_Toc131174260)

[1.2 Scope 3](#_Toc131174261)

[1.3 Definitions, Acronyms, Abbreviations 3](#_Toc131174262)

[1.4 Reference Documents 3](#_Toc131174263)

[1.5 Document Structure 3](#_Toc131174264)

[2 Overall description 3](#_Toc131174265)

[2.1 Product perspective 3](#_Toc131174266)

[2.2 Product functions 3](#_Toc131174267)

[2.3 User characteristics 3](#_Toc131174268)

[2.4 Assumptions, dependencies and constraints 3](#_Toc131174269)

[3 Specific requirements 3](#_Toc131174270)

[3.1 External Interface Requirements 3](#_Toc131174271)

[3.2 Functional Requirements 3](#_Toc131174272)

[3.3 Performance Requirements 3](#_Toc131174273)

[3.4 Design Constraints 3](#_Toc131174274)

[3.5 Software Systems Attributes 3](#_Toc131174275)

# **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. A RASD should work as baseline for the following tasks in software development, in particular in project planning, software evaluation and change control. Such document has a wide audience, and hence it has to be written as clear as possible.

## **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 owner can keep track in an easy way 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 which 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 | **Customers** can view the current queue status for the shop and estimated wait times. |
| G2 | **Customers** can book time slots in advance to visit the shop, reducing wait times. |
| G3 | **Customers** can receive notifications and alerts about their queue status and estimated wait times. |
| G4 | **Customers** can get discounts or other benefits by using the Iqueue app to visit shops. |
| G5 | **Customers** can provide feedback and ratings for the shops, helping other users make informed decisions. |
| G6 | **Customers** can easily search and discover new shops based on their preferences and location. |
| G7 | **Shop owners** can efficiently manage and organize the incoming flow of customers, improving their overall customer experience. |
| G8 | **Shop owners** can better forecast and plan their inventory and staffing needs based on the expected footfall of customers. |
| G9 | **Shop owners** can access analytics and insights on their business performance and customer behaviour, helping them make informed decisions. |
| G10 | **Shop owners** can improve their brand awareness and visibility by being featured on the app and leveraging the app’s marketing capabilities. |
| G11 | **Shop owners** can offer personalized promotions or deals to customers through the app. |
| G12 | **Shop owners** can build customer loyalty by offering a seamless and convenient experience through the app. |

## **Scope**

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

Table 2: List of World phenomena

|  |  |
| --- | --- |
| WP1 | A **customer** enters a shop |
| WP2 | A **customer** joins a queue at a shop |
| WP3 | A **shop experiences** high demand and long queues during peak hours |
| WP4 | A **shop experiences** low foot traffic and sales during off-peak hours |
| WP5 | A **shop owner** adjusts pricing or sales strategies to attract more customers |
| WP6 | A **shop owner** restocks inventory based on sales data and demand forecasts |
| WP7 | A **customer** provides feedback to a shop owner about their experience |
| WP8 | A **shop owner** updates their store layout or design to improve customer flow and experience |
| WP9 | A **shop owner** launches a marketing campaign to increase brand awareness and attract new customers |
| WP10 | A **shop owner** hires or trains new staff members to improve customer service and efficiency. |

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

Table 3: List of shared phenomena

|  |  |
| --- | --- |
| SP1 | A customer registers an account on the app |
| SP2 | A customer views nearby shops on the app based on their current location |
| SP3 | A customer selects a shop on the app to visit |
| SP4 | The app shows the estimated waiting time for the selected shop |
| SP5 | A customer books a time slot in advance on the app to visit the shop |
| SP6 | The app sends a notification to the customer when their turn in the queue is approaching |
| SP7 | A customer enters the shop and checks-in on the app |
| SP8 | The app updates the estimated waiting time based on the customer's check-in |
| SP9 | A customer cancels their booking on the app |
| SP10 | A customer provides feedback and ratings for the shop on the app |
| SP11 | The app shows recommended shops to the customer based on their previous visits and ratings |
| SP12 | A customer redeems rewards and offers on the app when making a purchase |
| SP13 | The app allows customers to track their rewards and loyalty points |
| SP14 | A customer views the shop's menu or product catalogue on the app |
| SP15 | The app allows customers to place orders for pickup or delivery from the shop |
| SP16 | 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 or directions to guide the customer to the shop. |
| 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. |

## **Definitions, Acronyms, Abbreviations**

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

## **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”;

## **Document Structure**

This document complies with the SRS11 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 in 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 to the user needs; it also provides the main assumptions under which Iqueue will work properly;
3. the third part contains the complete requirements of the system, from both the functional and the non-functional points of view;

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 a section to another, because the purpose of it is to be a reference document.

# **Overall description**

## **Product perspective**

## **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 chapter 3.

## **Data collection**

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

1. The customers who will book their presence in the shop trough the application. At each customer will correspond a QR code and thus a value in the waiting list. In this way the application will be able to detect the number of people in queue for the shop.
2. The shop owners who 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 ticket system: every time a new customer enters the shop without having the application, he must take a ticket (or scan a QR code) which will be read by the application. In this way Iqueue will be able to keep track also of the people in the waiting list without the application.

## **Data analysis**

The raw data collected by DREAM 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 queue in front of the corresponding shop with the application.
2. Associate the value coming from the ticket system (or QR code scan) to a number of people in 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, ..)

## **QR code reading and generating**

The application must be able to read the QR code for the customer with Iqueue so that he can access to special offers from the shop. In addition to that, Iqueue generates a QR code (or it uses a ticket system) so that it can track the people without the app in the shop. This QR code must be created so that it contains the information of the previous and the next number of people without the app in queue.

## **Special offer inseriment**

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

## **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 are the shops with the lowest number of people in queue, so that they can save time. In addition, having the Iqueue allow 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 costumers so that to advertise their activity and to create a lock in effect in the clients.

## **Assumptions, dependencies and constraints**

Table 4: List of domain assumptions

|  |  |
| --- | --- |
| D1 | The customers with the application own at least one mobile phone with 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 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. |
| D7 |  |
| D7 |  |

# Specific requirements

## **External Interface Requirements**

## User interfaces

Iqueue is provided to the users, namely customers and shop owners, as a application, accessible through a through a store of applications (Play Store or App Store). Therefore, Iqueue is not given with a CLI, *Command Line Interface* but only with a GUI *Graphical User Interface*.

## Hardware interfaces

Because Iqueue is to be implemented as an app, every user can access it through the device he prefers, 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.

## Software interfaces

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

1. every user’s device must have an application store (Play Store or App Store) installed on it through which the user can access the app; no other software requirements are requested to these kind of devices;
2. Iqueue requires also an interface to the GPS system such as Google maps or Maps so that it can link the queue for a specific shop with its position.

## **Functional Requirements**

Iqueue allows its users to perform many tasks, and is able to interact with other different systems. In order to provide the main requirements of the system and a summary of the possible situations in which Iqueue is involved and used, this paragraph first lists all the requirements of the system, then it lists some concrete scenarios and finally it abstracts from details and specificities showing the corresponding use cases.

## Requirements

Table 5: List of requirements

|  |  |
| --- | --- |
| R1 | Iqueue must allow the costumers to see the number of people in 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 offer for their shop |
| R4 | Iqueue must read the QR code of the clients with the app |
| R5 | The app shall allows customers to track their rewards and loyalty points |
| R6 | Iqueue must satisfy a customer who books a time slot in advance on the app to visit the shop |
| R7 | The app provides a map or directions to guide the customer to the shop. |
| R8 | A customer earns badges or achievements on the app for frequent visits or purchases |
| R9 | The app must show the shop's hours of operation, contact information, and other details |
| R10 | The app shall allows customers to create and save shopping lists or wish lists. |
| R11 | The app shall shows the customer's purchase history and receipts |
| R12 | The app shall allows customers to easily switch between different languages or currencies based on their preferences. |
| R13 |  |
| R14 |  |
| R15 |  |
| R16 |  |

## **Performance Requirements**

## **Design Constraints**

## **Software Systems Attributes**