



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH



---

# Caritapp: Implementation of a computer system that streamlines the distribution of food

---

## Degree Final Project

Author: Joan Febrer Nafría

Director: Daniel Jiménez González

Specialization: Information Systems

Bachelor Degree in Informatics Engineering

University: Universitat Politècnica de Catalunya

Date: 17 - Oct - 2022

## **Resum**

Aquest projecte està basat en una necessitat real del centre de Càritas Cerdanyola. En aquest document es recull la planificació i implementació de la solució proposada, així com tota la informació necessària per al correcte funcionament de l'aplicatiu.

Caritapp és una aplicació d'escriptori que permet gestionar la repartició d'aliments i els seus receptors, i generar documents d'entrega per al Fons d'Ajuda Europeu per a les Persones Més Desafavorides (FEAD).

## **Resumen**

Este proyecto está basado en una necesidad real del centro de Càritas Cerdanyola. En este documento se recoge la planificación e implementación de la solución propuesta, así como toda la información necesaria para el correcto funcionamiento del aplicativo.

Caritapp es una aplicación de escritorio que permite gestionar el reparto de alimentos y sus receptores, y generar documentos de entrega para el Fondo de Ayuda Europeo para las Personas Más Desfavorecidas (FEAD).

## **Abstract**

This project is based on a real need of the Càritas Cerdanyola center. This document includes the planning and implementation of the proposed solution, as well as all the necessary information to make the application work.

Caritapp is a desktop application that allows managing the food delivery and their receivers, and generating delivery documents for the European Aid to the Most Deprived (FEAD).

# INDEX

---

<b>1. Contextualization</b>	<b>8</b>
1.1. Cáritas	8
1.2. Motivation and formulation of the problem	8
1.3. Project objectives	9
1.4. Contextualization	9
1.5. Stakeholders	9
1.5.1. Developer, designer and beta tester	9
1.5.2. Director of the project	9
1.5.3. Parish priest of the Parish of Sant Martí de Cerdanyola	10
1.5.4. Cáritas Cerdanyola volunteer	10
1.5.5. Cerdanyola residents in state of poverty	10
1.5.6. Social worker	10
<b>2. State of Art</b>	<b>11</b>
2.1. How to receive help	11
2.2. How do they get food	12
2.2.1. European Economic Community	12
2.2.2. Spanish Federation of Food Banks	12
2.2.3. Other entities	12
2.3. How the distribution works	12
<b>3. Scope description</b>	<b>14</b>
3.1. Possible issues	14
3.1.1. Calendar	14
3.1.2. Schedule incompatibility	14
3.2. Methodology	15
3.3. Development Tools	15
3.4. Project tracking	15
<b>4. Planning</b>	<b>16</b>
4.1. Initial planning	16
4.2. Deviation on initial planning	16
<b>5. Stages of the project</b>	<b>17</b>
5.1. Definition of tasks	18
<b>6. Budget</b>	<b>19</b>
6.1. Identification and estimation of costs	20
6.1.1. Human resources	20
6.1.2. Material resources	22
6.1.3. Contingency	23
6.1.4. Unforeseen	23

6.1.5. Final budget	23
6.2. Management Control	24
<b>7. Sustainability report</b>	<b>25</b>
7.1. Economic dimension	25
7.2. Environmental dimension	25
7.3. Social dimension	26
<b>8. Requirements specification</b>	<b>27</b>
8.1. Process for obtaining and validating requirements	27
8.2. Hierarchy of actors	27
8.3. Functional requirements	28
8.3.1. Use case diagrams and description of use cases	28
8.4. Non functional requirements	48
8.4.1. Look and feel requirements	50
8.4.2. Performance requirements	51
8.4.3. Maintainability and support requirements	54
8.4.4. Security requirements	55
<b>9. Design</b>	<b>56</b>
9.1. Patterns used	57
9.2. Relational tables	57
<b>10. Implementation</b>	<b>59</b>
10.1. Technologies and languages used	59
10.2. Development tools	59
10.3. Security and data protection	60
10.4. Views	60
10.4.1. Login screen	61
10.4.2. Generic screen	63
10.4.3. Menu screen	64
10.4.4. Food recipient registration screen	66
10.4.5. Food recipient list screen	69
10.4.6. Food recipient information screen	70
10.4.6.1. Generate PDF	70
10.4.7. Delivery screen	73
10.4.7.1. Signature	74
10.4.8. Create calendar screen	75
10.4.8.1. Select parameters for calendar screen	75
10.4.8.2. Select days for calendar screen	75
10.4.9. Distributed food screen	77
10.4.10. Food recipient tracking screen	78
10.4.11. Account Management	79

10.4.11.1. Volunteer accounts management screen	79
10.4.11.2. Member account management screen	80
10.4.12. Backup management screen	81
<b>11. Results</b>	<b>82</b>
11.1. Assessments	82
11.2. User manual	83
11.3. Use and surveys	83
<b>12. Conclusions</b>	<b>83</b>
12.1. Difficulties encountered	83
12.2. Relationship of the project with the degree	84
12.3. Achievement of specialist technical skills	84
12.4. Future work	85
12.4.1. Maintenance and support	85
12.4.2. Scalability of project	86
12.5. Acknowledgements	87
<b>Bibliography</b>	<b>88</b>

## INDEX OF FIGURES

---

Figure 1: Flowchart How to receive help	11
Figure 2: Flowchart How distribution works	13
Figure 3: Use case diagrams for Cáritas member (I)	28
Figure 4: Use case diagrams for Cáritas member (II)	31
Figure 5: Use case diagrams mixed	34
Figure 6: Use case diagrams for Cáritas administrator (I)	37
Figure 7: Use case diagrams for Cáritas administrator (II)	40
Figure 8: Use case diagrams for Cáritas administrator (III)	45
Figure 9: Use case diagrams for Cáritas administrator (IV)	45
Figure 10: Caritapp MVC	57
Figure 11: UML class diagram	58
Figure 12: Error message	60
Figure 13: Login screen	62
Figure 14: Login screen - login successfully	62
Figure 15: Login screen - recover password	62
Figure 16: Login screen - showing password	62

Figure 17: Login screen - information panel	63
Figure 18: Generic screen	65
Figure 19: Menu screen with admin privileges	65
Figure 20: Menu screen as a member	66
Figure 21: Register new food recipient screen	68
Figure 22: Food recipient list screen	69
Figure 23: Food recipient information screen	70
Figure 24: Food delivery note 2022 in Spanish	71
Figure 25: Food recipient information screen - Select file modal	71
Figure 26: Food recipient information screen - Select day modal	72
Figure 27: View pdf screen	72
Figure 28: Delivery screen	74
Figure 29: Delivery screen - Signature modal	74
Figure 30: Create calendar screen - First screen	75
Figure 31: Create calendar screen - Second screen	76
Figure 32: Distributed food screen	77
Figure 33: Distributed food screen - Delete rows modal	78
Figure 34: Food recipient tracking screen	78
Figure 35: Volunteer accounts management screen	80
Figure 36: Volunteer my space management screen	81
Figure 37: Backup management screen	82

## INDEX OF TABLES

---

Table 1: Planned dates for each stage	17
Table 2: Definition of tasks	19
Table 3: Cost per hour of each role	20
Table 4: Human costs of each task (I)	21
Table 5: Human costs of each task (II)	22
Table 6: Cost of material resources	22
Table 7: Cost of different sources with contingency	23
Table 8: Final budget	23
Table 9: Description of use case 1	29
Table 10: Description of use case 2	29

Table 11: Description of use case 3	30
Table 12: Description of use case 4	30
Table 13: Description of use case 5	31
Table 14: Description of use case 6	32
Table 15: Description of use case 7	32
Table 16: Description of use case 8	33
Table 17: Description of use case 9	33
Table 18: Description of use case 10	34
Table 19: Description of use case 11	35
Table 20: Description of external use case 12	35
Table 21: Description of external use case 13	36
Table 22: Description of external use case 14	36
Table 23: Description of use case 15	38
Table 24: Description of use case 16	38
Table 25: Description of use case 17	39
Table 26: Description of use case 18	39
Table 27: Description of use case 19	41
Table 28: Description of use case 20	41
Table 29: Description of use case 21	42
Table 30: Description of use case 22	42
Table 31: Description of use case 23	42
Table 32: Description of use case 24	43
Table 33: Description of use case 25	43
Table 34: Description of use case 26	44
Table 35: Description of use case 27	46
Table 36: Description of use case 28	46
Table 37: Description of use case 29	47
Table 38: Description of use case 30	47
Table 39: Description of use case 31	47
Table 40: Non functional requirements	49
Table 41: Non functional requirement 1	50
Table 42: Non functional requirement 2	50
Table 43: Non functional requirement 3	51
Table 44: Non functional requirement 4	51

Table 45: Non functional requirement 5	52
Table 46: Non functional requirement 6	52
Table 47: Non functional requirement 7	53
Table 48: Non functional requirement 8	53
Table 49: Non functional requirement 9	54
Table 50: Non functional requirement 10	54
Table 51: Non functional requirement 11	55
Table 52: Non functional requirement 12	55
Table 53: Non functional requirement 13	56
Table 54: Non functional requirement 14	56
Table 55: Login screen (CC) list of errors	63
Table 56: Register new food recipient (RFR) list of errors	68
Table 57: Food recipient information (FRI) list of errors	72
Table 58: Create calendar (CC) list of errors	76
Table 59: Backup management (BM) list of errors	82



# 1. Contextualization

## 1.1. Cáritas

This project is based on the development of an application that allows managing the distribution of food for Cáritas Cerdanyola. Before exposing anything I will explain what Cáritas is and what it does.

Cáritas is defined as the official confederation of charitable and social action entities of the Catholic Church. It was established in 1947 by the Spanish Episcopal Conference. Cáritas Española aims to carry out the charitable and social action of the Church in Spain, through its confederated members. They promote the comprehensive development of people and peoples, especially the poorest and most excluded (Cáritas website, 2022).

Cáritas receives help from private and public entities. During the course of 2020 Cáritas received **273,14 millions of euros** from private entities and **113,57 millions of euros** from public entities (Cáritas website, 2022).

## 1.2. Motivation and formulation of the problem

Every human being has the fundamental right to be free from hunger, according to international human rights standards. The right to adequate food includes the quantitative, qualitative and cultural acceptability aspects. (Food and Agriculture Organization <FAO>, 2004) But even in rich cities there are poor people who do not see this need fulfilled, and Cerdanyola is no exception.

Cáritas is providing these people with food so no one goes hungry. It is not the only work they do, but it is what we are going to focus on for the purpose of this work.

The idea of this project came one day when I was talking to the priest of the Parish of Sant Martí de Cerdanyola and I was told about Cáritas and how they distribute food. Everything was done by hand, there was nothing computerized. That caused them delays and time waste. Every time someone arrived to ask for food, they had to check from a list printed in paper if that person had the right to take the weekly loot, and if he or she already got it that week.

### 1.3. Project objectives

The goal of this project is to build an information system capable of having an accessible user control, in order to give Cáritas greater control over food outflows, by the means of:

1. Making the volunteers able to know if the people who come to ask for food are within the database and if they are allowed to collect the pack or not.
2. Automatically notifying all those who have not come to collect food throughout the week, and reminding them that, if they do not attend the center in a certain period of time, the help will be withdrawn.

### 1.4. Contextualization

Our focus will be in the creation of software easy to use and that covers the specific needs required by this project in particular, creating an information system that brings more agility on the day-to-day basics.

### 1.5. Stakeholders

Listed below are the different stakeholders involved, whether they are interested people, beneficiaries, affected people or participants in development.

#### 1.5.1. Developer, designer and beta tester

The agent in charge of this part will be me; the functions that I will carry out will be to design the information system, implement it and test its results, with the help of my director and my GEP tutor.

#### 1.5.2. Director of the project

The director of the project is Daniel Jiménez González. His role will be to supervise, help and guide throughout the project so that its completion is as expected.

### 1.5.3. Parish priest of the Parish of Sant Martí de Cerdanyola

Juan Carlos Montserrat, the parish priest, is the bridge between Càritas Cerdanyola and me. He is the one who determines the functionalities that the system should have. His function will be to review the progress that is made from the sprints.

### 1.5.4. Càritas Cerdanyola volunteer

Sister Eva will be the direct contact with Càritas Cerdanyola; her role will be, together with the priest Juan Carlos Montserrat, reviewer of the progress. She will provide access to the necessary information for me to fully understand the processes followed by Càritas Cerdanyola in relation to the distribution of food.

### 1.5.5. Cerdanyola residents in state of poverty

Last but not least, the real users of the application. These users are the residents of Cerdanyola in a state of poverty who need to receive food aid and have contacted Càritas.

### 1.5.6. Social worker

The social worker is responsible for evaluating the health status of a person in terms of: the social conflicts they may be going through, their causes, how they originate at an individual and social level, among other things. They assess whether the resident is in an unfavorable condition and has access to assistance.

## 2. State of Art

### 2.1. How to receive help

People who want to receive this help must be assigned to a social worker, who will evaluate their case. If a person is considered a candidate to receive this help, Cáritas is informed and a file is passed with all their data. This is how Cáritas knows who can collect food.

We can see in Figure 1 a flowchart of how the flow of asking Cáritas for help works:

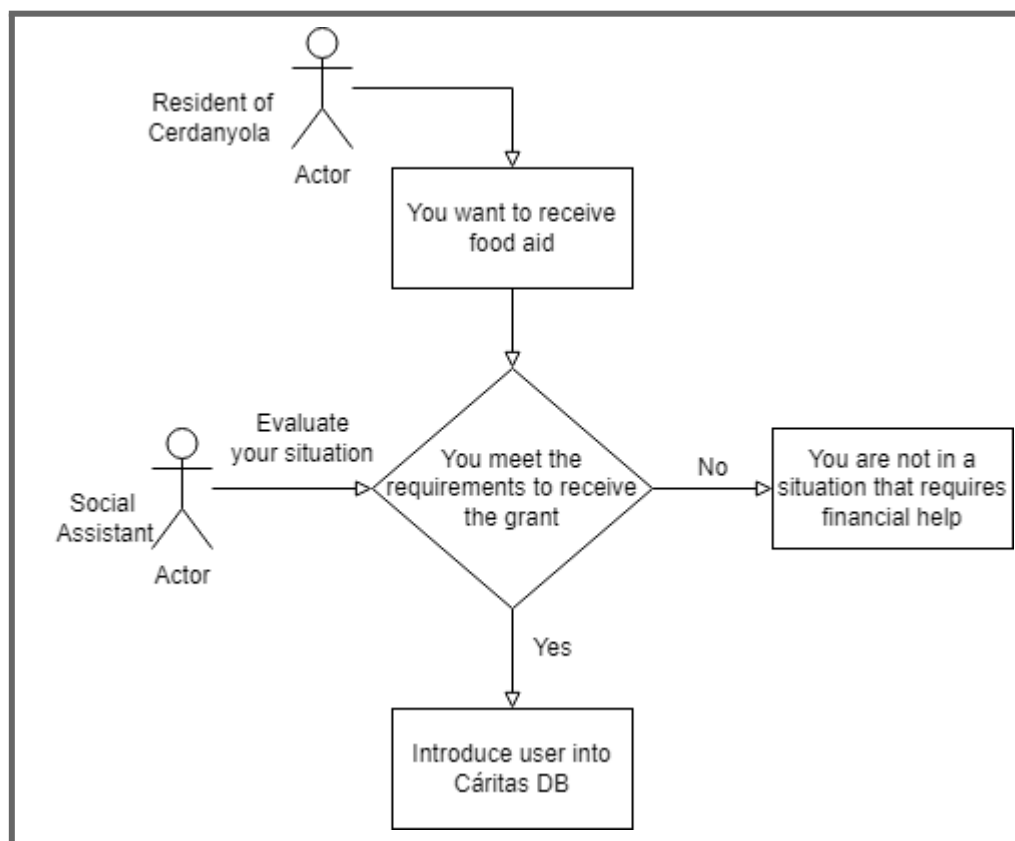


Figure 1: Flowchart How to receive help

## 2.2. How do they get food

To get the food it distributes, Cáritas receives help from other entities. Next, we will detail what these entities are.

### 2.2.1. European Economic Community

The European Economic Community (EEC) is a regional organization that aims to bring about economic integration among its member states. (EEC website, 2022) Cáritas needs to report to EEC how they distribute the food that is delivered by them.

### 2.2.2. Spanish Federation of Food Banks

The Spanish Federation of Food Banks (FESBAL) is an apolitical and non-denominational entity, founded in 1995, which promotes the work, image and figure of the associated Food Banks in their fight against hunger, poverty and waste of food through its use and distribution to the most needy people, also contributing to improve the environment (FESBAL website, 2022). There is no need to report the distribution of this food.

### 2.2.3. Other entities

This group includes supermarkets, local stores and private entities. As happened with FESBAL, it is not necessary to report the distribution.

## 2.3. How the distribution works

Each user has a file with the data provided to Cáritas by the social worker; in this file, users who have access to help are divided into 2 groups, small families (B) and large families (A). Depending on the family size, users will have access to more or less food.

Every Tuesday the batches are prepared to be distributed the next day. Wednesday is delivery day, people arrive at the distribution place and ask for their pack. Cáritas volunteers need to search from a list if that person is in it and if she/he already got the pack that week.

At the end of the day they have to manually look at the people who have not come to pick up their lot. For those who have not come, an email/SMS/Whatsapp is sent reminding them that they have access to picking up food, and they are warned that if they do not pick the aid up within a certain period of time, it will be withdrawn.

We can see in Figure 2 a flow chart of how food distribution works.

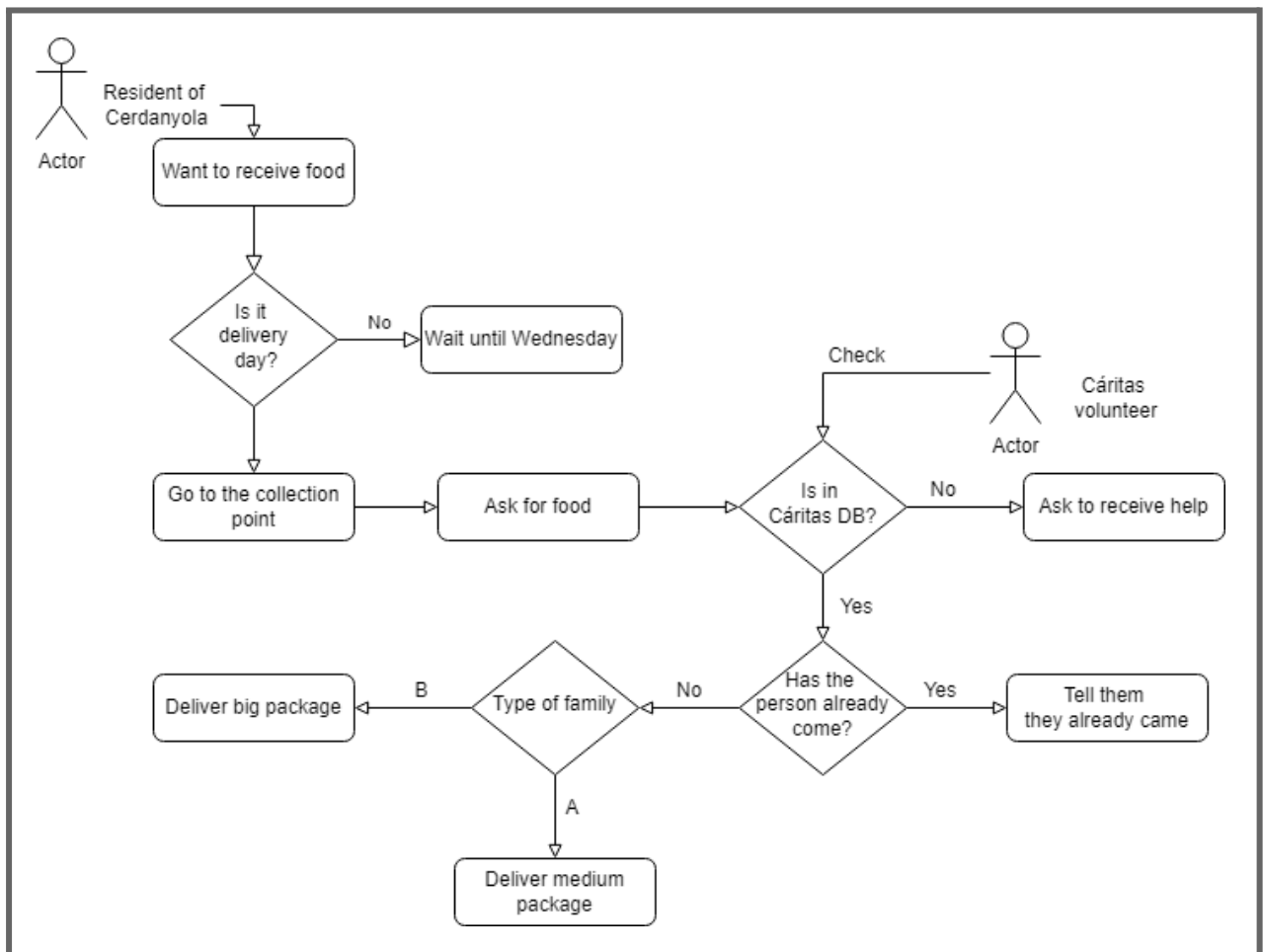


Figure 2: Flowchart How distribution works

## 2.4 Existing solutions

To examine existing solutions, although they are not exactly the same, we will look at the operation of the restaurant where the waiter has direct communication via tablet or phone with the kitchen.

An example would be the Viena restaurant. The customer arrives at the box and asks for her/his menu, this order is passed to the kitchen, the kitchen warns when it

has it ready and leaves it on its corresponding grill. When the worker distributes the food in the trays, he/she records that it has been delivered.

### 3. Scope description

According to the first version of the project, defined through the first meeting with the members of Cáritas and the priest of Cerdanyola, the objective of the project is to create a simple, easy-to-use and user-friendly tool. It has to cover the basic aspects required for the cast. It is meant to be a database that encompasses all users.

The creation of a Customer Relationship Management (CRM) allows us to have this database. Whenever we enter a DNI and the user's information, we can know if a batch of type A or B is to be delivered, if it has been already collected or if the user is not within the aid program. It also allows users to register or, on the contrary, to unsubscribe.

The system generates a report of the amount of food that has been distributed, specifically that provided by the EEC. Finally we have also included a management system that allows the automatic sending of messages to all the people who have not come to pick up their lot.

#### 3.1. Possible issues

##### 3.1.1. Calendar

The duration of the project is short, the objective in case of lack of time is to create a simple and functional system, with the possibility of being expanded in the future. In case the calendar is not a problem, the idea would be to add new features to the baseline system.

##### 3.1.2. Schedule incompatibility

Due to the fact that I am working full time on other projects, and that Cáritas volunteers also have other occupations, it could be a problem to make our schedules coincide to do the corresponding sprints.

## 3.2. Methodology

We will use agile methodology, specifically Scrum. By definition, agile methodologies are those that allow the way of working to be adapted to the conditions of the project, achieving flexibility and immediacy in the response to adapt the project and its development to the specific circumstances of the environment. (Hurtado, Javier Sáez, 2021)

Scrum is an agile development methodology used in the development of Software based on iterative and incremental processes. Scrum is an adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project (Digite website, 2019).

## 3.3. Development Tools

I will use Git to have better control over the various versions of the project. Java will be used for the language, using Eclipse as the IDE. There will be a desktop application.

## 3.4. Project tracking

The idea is to have regular sprints (2/3 weeks) with members of Cáritas, the director and me. That will give us a total of 6 sprints.



## 4. Planning

### 4.1. Initial planning

The approximate duration of the project is 4 months (119 days), starting on February 21st with the management of the project and ending on June 20th, as the reading is on June 30th.

From February 21st to March 15th, the project is managed, analyzing the context, defining the scope and time planning, doing the risk and cost analysis, and doing the sustainability report. Following the requirements of GEP, this strip will consist of 76 hours of work.

Development will begin on March 16th, following the Scrum methodology, and end on June 13th. The estimated time of development will consist of 315 hours of work.

Finally, between June 13th and 20th, the memory of the project will be done. This part consists of 60 hours of work including the documentation made in each sprint plus the final review of the documentation to finalize the small details. It is important to know that the memory report has to be delivered to the tribunal one week before the presentation. Thus, in total the project will consist of 451 hours of work.

### 4.2. Deviation on initial planning

An extension of 4 months has been necessary for the completion of the project. This extension has given the necessary time to achieve the objectives mentioned in the initial planning. In this extension, work has been done with an extension of 95 hours, making the project last 546 hours.

Therefore, all planning points from now on will proceed on the basis that the project has lasted 8 months. That is to say, they will show “what has happened” and not “what would have happened according to the initial estimation”.

## 5. Stages of the project

The project is divided into 8 stages. The first stage is project management, also called the project start-up stage. Below we will have the 6 stages of development, which are called sprints. Finally a period for the documentation and completion of the project memory report.

For each of the sprints we will have assigned tasks and for each of these tasks we will have a document on the sprint meeting. Once the stage is over, we will do a retrospective, assessing whether or not the objectives have been achieved and proposing possible improvements for the future stages. Finally, the last stage will be the writing of the memory.

We can see the planning of the start and end dates of each stage in Table 1.

Stage	Initial date	Final date
Project management	21-02-2022	21-03-2022
Sprint 1	22-03-2022	06-04-2022
Sprint 2	07-04-2022	27-04-2022
Sprint 3	29-04-2022	18-05-2022
Sprint 4	19-05-2022	20-06-2022
Sprint 5	21-06-2022	30-07-2022
Sprint 6	01-09-2022	15-09-2022
Final documentation	16-09-2022	10-10-2022

Table 1: Planned dates for each stage

Tasks will be divided in tasks of project management (PM), development (DV), documentation (DC), validity and deployment (VD).

## 5.1. Definition of tasks

Abbreviation	Details	Duration (h)	Dependencies
<b>DESIGN</b>			
PM1	Scope and contextualization	24.5	-
PM2	Planning	18.25	PM1
PM3	Budget and sustainability	18.25	PM2
PM4	Integration of the final document	15	PM3
<b>IMPLEMENTATION</b>			
<b>Sprint 1</b>			
DV1	Beginning of development	20	-
DV2	DB creation	20	-
DC1	Memory documentation of sprint 1	10	DV1,DV2
<b>Sprint 2</b>			
DV3	Simple program for managing DB	30	DV2
DV4	Visual consultation of the user data	30	DV3
PM5	Evaluation of sprint 1	5	Sprint 1
DC2	Memory documentation of sprint 2	10	Sprint 2
<b>Sprint 3</b>			
DV5	Attendance control	10	DV3
DV6	Mail/SMS/Whatsapp if no attendance	30	DV5
DV7	Creation of calendar	10	-
DV8	Creation of monitoring table	10	DV5
PM6	Evaluation of sprint 2	5	Sprint 2
DC3	Memory documentation of sprint 3	10	Sprint 3
<b>Sprint 4</b>			
DV9	Report for CEE	40	-
DV10	Definition of type of users	10	-
DV11	Creation of observation field	10	DV5

<b>Abbreviation</b>	<b>Details</b>	<b>Duration(h)</b>	<b>Dependencies</b>
DV12	<b>Backup/Restore DB</b>	15	DV2
DV13	<b>Capture signature</b>	25	-
PM7	<b>Evaluation of sprint 3</b>	5	Sprint 3
DC4	<b>Memory documentation of sprint 4</b>	10	Sprint 4
<b>Sprint 5</b>			
DV14	<b>Creation of Login Screen</b>	15	-
DV15	<b>Volunteer type of users</b>	5	-
DV16	<b>Intuitive Menu Interface</b>	10	-
PM8	<b>Evaluation of sprint 4</b>	5	Sprint 4
DC5	<b>Memory documentation of sprint 5</b>	10	Sprint 5
<b>Sprint 6</b>			
DV17	<b>Creation of volunteer management</b>	15	DV15
DV18	<b>Generate document PDF in batch</b>	10	DV9
DV19	<b>Volunteer account details, and management</b>	10	DV15
PM9	<b>Evaluation of sprint 5</b>	5	Sprint 5
DC6	<b>Memory documentation of sprint 6</b>	10	Sprint 6
<b>RESULTS</b>			
VD1	<b>Project validation</b>	15	-
VD2	<b>Deployment tutorial</b>	15	VD1
DC7	<b>Finalization of memory</b>	20	-

Table 2: Definition of tasks

The tasks were developed in the order they appear in Table 2. Since all the work was done by me, there was not much that could be parallelized. And so I've been using the method of starting a task and ending it.

## 6. Budget

### 6.1. Identification and estimation of costs

A number of human and material resources are needed in order for this project to be considered a development success.

Material resources are the software and hardware with which we will develop the project, human resources are the cost of all staff involved in it. There are also indirect resources, such as the cost of electricity used to develop the project, the cost of travel, and also all those arising from contingencies.

#### 6.1.1. Human resources

First of all, in Table 3 we can see the price per hour (in €) of each of the roles that will be part of the project (Siordia, Omar Rodriguez, 2020). The cost of social security must be added to this price.

Role	Salary/hour gross	Salary/hour net
Project manager	32	42
Developer	18	23
UI/UX Designer	25	32
QA Tester	18	23

Table 3: Cost per hour of each role

Below we can see, in Table 4, the price related to the dedication of each of the roles with the tasks described in section 5. I have decided to make a division by task so that I have a better view of the cost of each of these.

Task	Project Manager	Developer	UI/UX Designer	QA Tester	Total
PM1	1,029€				1,029€
PM2	766€				766€
PM3	766€				766€
PM4	630€				630€
DV1		460€			460€
DV2		460€			460€
DC1	420€				420€
DV3		690€			690€
DV4		230€	640€		870€
PM5	210€				210€
DC2	420€				420€
DV5		345€		345€	690€
DV6		460€		230€	690€
DV7		184€		46€	230€
DV8		184€		46€	230€
PM6	210€				210€
DC3	420€				420€
DV9		690€	160€	115€	965€
DV10		184€		46€	230€
DV11		184€		46€	230€
DV12		345€			345€
DV13		414€	64€	115€	593€
PM7	210€				210€
DC4	420€				420€

Table 4: Human costs of each task (I)

Task	Project Manager	Developer	UI/UX Designer	QA Tester	Total
DV14		180€	125€		305€
DV15		90€			90€
DV16		90€	125€		215€
PM8	210€				210€
DC5	420€				420€
DV17		180€	125€		305€
DV18		90€		90€	180€
DV19		180€			180€
PM9	210€				210€
DC6	420€				420€
VD1		115€	160€	115€	390€
VD2		115€	160€	115€	390€
DC7	840€				840€
<b>TOTAL</b>	<b>6,572€</b>	<b>5,870€</b>	<b>1,559€</b>	<b>1,309€</b>	<b>15,310€</b>

Table 5: Human costs of each task (II)

### 6.1.2. Material resources

My desktop computer will be used during the course of the project. All software used is free of charge and is therefore not included in the material costs.

Another material resource is the cost of electricity to carry out the project. The hourly power consumption of my computer is 57 W (OuterVision website, 2022).

$$\frac{\text{Cost(Euros)}}{\text{Useful life(year)} * 220(\text{working days/year}) * \text{dedication/day(hours)}} * \text{project duration(hours)}$$

Source	Price	Total(457h)
Electricity	0.4€/kWh	11€

Element	Cost (€)	Useful life (years)	Dedication/day	Amortization (€)
Computer	990	5	4	86

Table 6: Cost of material resources

### 6.1.3. Contingency

For contingencies, a specific percentage of 10% is set. It is a measure to prevent any uncertain moment in the future. As we can see in Table 7, 10% is added to each type of cost.

Type of cost	Cost (€)	Contingency (%)	Final cost (€)
Human cost	15,310	10%	16,841
Material cost	97	10%	107

Table 7: Cost of different sources with contingency

### 6.1.4. Unforeseen

Unforeseen events are situations that affect time planning, causing deviations from the agreed deadlines. There may be contingencies with material and human resources. In the case of material resources, there may be a possibility that the computer you are working with may be broken, or that a component may need to be repaired. In our case, during the course of the project, there were no unforeseen events.

### 6.1.5. Final budget

We can see the final budget result in Table 8:

Type of cost	Final cost
Human cost	16,841€
Material cost	107€
Total	16,948€

Table 8: Final budget

Note: As this is a social project, there will not be any charge for this project.



## 6.2. Management Control

To keep track of deviations in the budget more accurately, when closing a task, the budget will be updated based on the hours used to solve that task, especially to control any unforeseen events that may have occurred. We will use a series of formulas to calculate the deviations.

- **Deviation of hours consumed per task**

$$(Estimated\ hours - Actual\ hours) * Estimated\ cost$$

- **Deviation of costs according to the hours consumed per task**

$$(Estimated\ hours - Actual\ hours) * Actual\ cost$$

- **Deviation of human resource costs per task**

$$(Estimated\ cost - Actual\ cost) * Actual\ hours$$

- **Total deviation from material costs**

$$Estimated\ material\ cost - Actual\ material\ cost$$

- **Total deviation from indirect costs**

$$Estimated\ indirect\ cost - Actual\ indirect\ cost$$

- **Total deviation from the unforeseen**

$$Estimated\ unforeseen\ cost - Actual\ unforeseen\ cost$$

- **Total deviation of staff costs**

$$Estimated\ personal\ cost - Actual\ personal\ cost$$

- **Total deviation of hours**

$$Estimated\ hours - Actual\ hours$$

- **Total cost deviation**

$$Estimated\ total\ cost - Actual\ total\ cost$$

## 7. Sustainability report

When we talk about sustainability, we always think about the environment. However, the social and economic resources are also part of this sustainability.

What we can draw from sustainability is that, in the end, the resources we have on Earth are limited and we need to make good use of them so that they will continue to exist in the future.

Below we will detail each of the dimensions of sustainability by answering the questions of the Sustainability matrix.

### 7.1. Economic dimension

When we speak of economic sustainability, we refer to the ability to generate wealth in the form of adequate amounts, equitable in different social spheres that is a capable and solvent population of its economic problems, as well as strengthening production and consumption in monetary production sectors. In a few words, it is a balance between human beings and nature to satisfy needs and not sacrifice future generations. (Responsabilidad social website, 2022)

**Have you estimated the cost of carrying out the project (human and material resources)?**

Material and human costs have been taken into account to carry out this project.

**How is the problem you want to address currently solved (state of the art)?**

At the moment all the processes are done without any computerization involved.

**How will your solution economically improve the existing ones?**

It will not really have a direct "economic" impact, but will be reflected in the time savings they will have in using the application.

### 7.2. Environmental dimension

When we speak of environmental sustainability, we refer to the ability to maintain biological aspects in their productivity and diversity over time and, in this way, take care of the preservation of natural resources by promoting a conscious responsibility

regarding the ecological. At the same time, it also means to grow in human development while taking care of the environment where you live. Currently many companies have begun to promote these changes (Responsabilidad social website, 2022).

I have made a small estimate of the impact that the project will have on the environment. It is true that due to the use of the application there will be a consumption of energy resources, but at the same time it can be used to reduce the number of sheets used by the lists and therefore it is a pull.

### 7.3. Social dimension

When we speak of social sustainability, we refer to adopting values that generate behaviors such as the value of nature; maintaining harmonious and satisfactory levels of education, training and awareness; offering support to the population of a country to improve themselves and keep a good standard of living; and promoting the involvement of these same people to create something new in the society they are part of today (Responsabilidad social website, 2022).

#### **What do you think the realization of this project will bring you on a personal level?**

It is a project with a huge social aspect: the fact of being able to contribute, even in a small way, with an entity such as Cáritas, is priceless. Plus it is a new challenge and therefore a good source of learning.

#### **How is the problem you want to address currently solved (state of the art)?**

Every Tuesday the batches of food are prepared to be distributed the next day. Wednesday is delivery day, people arrive at the distribution place and ask for his/her pack. Cáritas volunteers need to search from a list if that person is in the list and if she/he already got the pack of food that week.

At the end of the day they have to manually look at the people who have not come to pick up their lot. For those who have not come, an email is sent reminding them that they have access to pick up food and they are warned that, if they do not do it within a certain period of time, the aid will be withdrawn. So the management control takes a lot of time.

## How will your solution improve socially (quality of life) over existing ones? Is there a real need for the project?

As I described in the previous question, controlling all this takes a long time and therefore also leads to a waste of time. With the solution we will try to improve the management time and therefore the quality of life of the volunteers.

## 8. Requirements specification

The functional and non-functional requirements of the system are specified below.

### 8.1. Process for obtaining and validating requirements

To obtain and validate these requirements, I have had several meetings with some members of Càritas Cerdanyola. Once the meetings have taken place, the following conclusions are drawn:

- You need a system that allows you to have food recipients in a database.
- It is requested that the application allows to mark the days that a person has come to collect food, in order to keep a record and notify in case of absence.
- It is requested the generation of a document with the times that a user has come and her/his signature, to manage the follow-up to be done by the FEAD.

### 8.2. Hierarchy of actors

Our system will be used by three actors.

- **Càritas volunteer:** Actor representing a member interacting with the system. This volunteer can belong to two categories: Càritas administrator or Càritas member.
- **Food recipient:** Actor who is registered in the system database and goes to the center to collect food.
- **Social worker:** This actor allows you to register or unsubscribe a food recipient.

### 8.3. Functional requirements

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So it is important to make them clear, both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions.

There are some test cases that are external to the application, but are described to try and make it easier to understand. We will refer to them as External Use Cases. In this section we will see the use case diagrams and their description.

#### 8.3.1. Use case diagrams and description of use cases

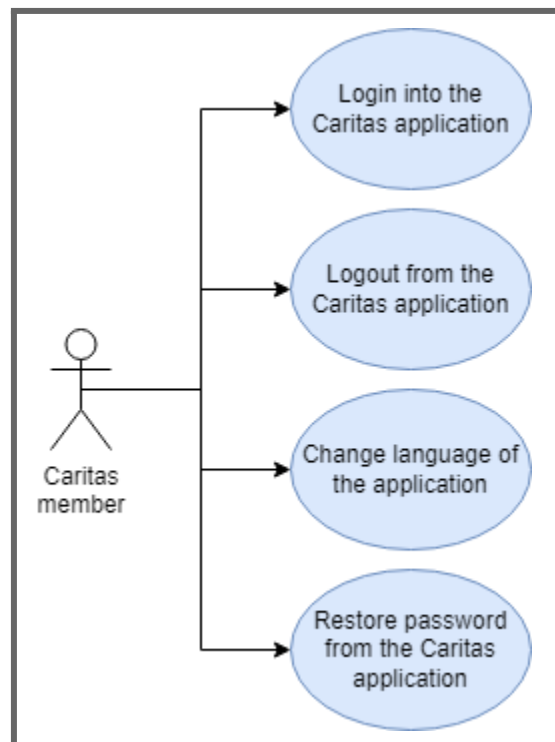


Figure 3: Use case diagrams for Cáritas member (I)

<b>Use case 1</b>	<b>Login into the Cáritas application</b>
Principal Actor	Cáritas administrator/member
Precondition	The user is not logged in
Trigger	The user wants to log in to the system
Main stage of success	<ol style="list-style-type: none"> <li>1. The system displays a form for the user to fill out with the information needed to login.</li> <li>2. The user enters their username and password.</li> <li>3. The user confirms the data entered.</li> <li>4. The system validates the data.</li> <li>5. The system logs in the user and calls the use case Consult food recipient list (member) or Register new food recipient (administrator).</li> </ol>
Extensions of the use case	<p>2a/3a. The user does not want to log in to the system.</p> <p>5a. Not all required information was entered, or some information entered is incorrect.</p> <ul style="list-style-type: none"> <li>- 5a1. The user is incorrect. The system displays a message stating that the user is incorrect and returns to step 2.</li> <li>- 5a2. The user / password combination is incorrect. The system displays a message stating that the user / password combination is incorrect and returns to step 2.</li> <li>- 5a3. Some of the fields are empty. The system displays a message stating that one of the fields is empty and returns to step 2.</li> </ul>

Table 9: Description of use case 1

<b>Use case 2</b>	<b>Logout from Cáritas application</b>
Principal Actor	Cáritas administrator/member
Precondition	The user is logged in
Trigger	The user wants to log out of the system
Main stage of success	<ol style="list-style-type: none"> <li>1. The system logs out of the user and calls the use case Login into the Cáritas application.</li> </ol>

Table 10: Description of use case 2

<b>Use case 3</b>	<b>Change language of application</b>
Principal Actor	Cáritas administrator/member
Precondition	The user is viewing the login screen
Trigger	The user wants to change the language of the application
Main stage of success	1. The user changes the language of the application to Catalan, Spanish or English.

Table 11: Description of use case 3

<b>Use case 4</b>	<b>Restore password from Cáritas application</b>
Principal Actor	Cáritas administrator/member
Precondition	The user is viewing the login screen and is not logged in
Trigger	The user wants to log in but does not remember their credentials
Main stage of success	<ol style="list-style-type: none"> <li>1. The system displays a form so that the user can fill it in with the necessary information to be able to request the reset of their password.</li> <li>2. The user enters their email.</li> <li>3. The user confirms the data entered.</li> <li>4. The system validates the data.</li> <li>5. The system sends a message to the user's email with their username and a new generated password.</li> </ol>
Extensions of the use case	<p>2a/3a. The user does not want to recover their password.</p> <p>5a. Not all required information was entered, or some information entered is incorrect.</p> <ul style="list-style-type: none"> <li>- 5a1. The user has entered an email that does not exist in the system, the system displays a message informing that the email does not belong to any member and returns to the use case Login into Cáritas application.</li> <li>- 5a2. The user has entered an invalid email in the system, the system displays a message informing that the email is invalid and returns to the use case Login into Cáritas application.</li> </ul>

Table 12: Description of use case 4

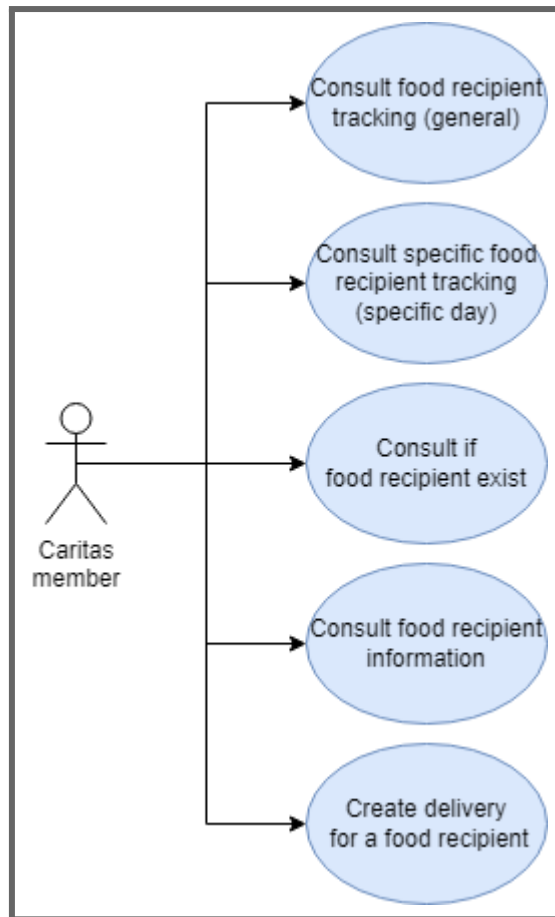


Figure 4: Use case diagrams for Cáritas member (II)

Use case 5	Consult food recipient tracking (general)
Principal Actor	Cáritas administrator/member
Precondition	The user is viewing the food recipient tracking tab
Trigger	The user wants to consult the tracking of all food recipients of a certain family type in a certain year
Main stage of success	<ol style="list-style-type: none"> <li>1. The user searches for the food recipients and specific dates he wants to consult.</li> <li>2. The user may use the filter to search for food recipients.</li> <li>3. The user confirms the data introduced.</li> <li>4. The system shows the food recipients that match with the search.</li> </ol>

Table 13: Description of use case 5



<b>Use case 6</b>	<b>Consult specific food recipient tracking (specific day)</b>
Principal Actor	Cáritas administrator/member
Precondition	The user is viewing the food recipient tracking tab
Trigger	The user wants to consult the information for a specific day
Main stage of success	<ol style="list-style-type: none"> <li>1. The user searches for the food recipient and specific date she/he wants to consult.</li> <li>2. The user confirms the data.</li> <li>3. The user double clicks the cell.</li> <li>4. The system shows in a new tab the information related to that delivery.</li> </ol>
Extensions of the use case	3a. The user has not received food that day. The system displays a message informing that there is no data related to that day for that food recipient.

Table 14: Description of use case 6

<b>Use case 7</b>	<b>Consult if food recipient exists</b>
Principal Actor	Cáritas member
Precondition	The user is viewing the food recipient list tab.
Trigger	The user wants to search for the existence of a food recipient
Main stage of success	<ol style="list-style-type: none"> <li>1. The system shows a table of all the food recipients in the database.</li> <li>2. The user can use the search engine to speed up the search or to filter. This filtering can be done by all the fields in the table (name, last name, ID ...).</li> <li>3. The system returns users who match the search.</li> </ol>
Extensions of the use case	3a. No food recipient found No food recipient matches with the search. The system shows no results.

Table 15: Description of use case 7

<b>Use case 8</b>	<b>Consult food recipient information</b>
Principal Actor	Cáritas administrator/member
Precondition	The user is viewing the food recipient list tab.
Trigger	The user wants to see the information about a food recipient
Main stage of success	<ol style="list-style-type: none"> <li>1. The system shows in a table of all the food recipients in the database.</li> <li>2. The user can use the search engine to speed up the search or to filter. This filtering can be done by all the fields in the table (name, last name, ID ...).</li> <li>3. The system returns users who match the search.</li> <li>4. The user double-clicks on the row of the user they want to get information about.</li> </ol>

Table 16: Description of use case 8

<b>Use case 9</b>	<b>Create a delivery for a food recipient</b>
Principal Actor	Cáritas administrator/member
Precondition	The user is viewing consult user tab
Trigger	The user wants to create a delivery for a food recipient
Main stage of success	<ol style="list-style-type: none"> <li>1. The user presses the delivery button.</li> <li>2. The system displays a new tab with delivery information, a comment field, and a signature field.</li> <li>3. The user can add comments about the delivery by pressing the comment button.</li> <li>4. The user presses the delivery button.</li> <li>5. The food recipient must make a signature.</li> <li>6. The delivery is made.</li> <li>7. The user delivers the corresponding batch of food to the food recipient.</li> </ol>
Extensions of the use case	<ol style="list-style-type: none"> <li>4a. A delivery has already been made. In this case the system displays a message informing that this food recipient has already received the batch.</li> <li>5a. The food recipient does not want to sign. The system returns to step 2.</li> </ol>

Table 17: Description of use case 9

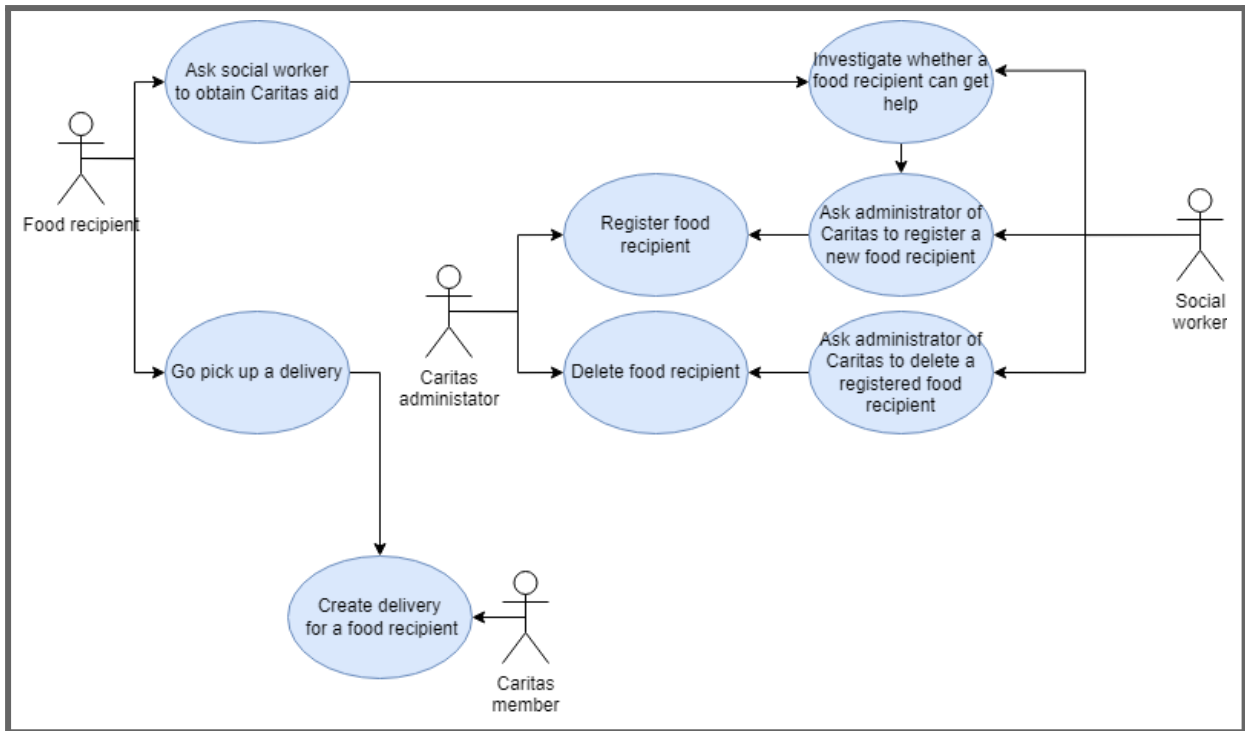


Figure 5: Use case diagrams mixed

Use case 10	Ask social worker to obtain Cáritas aid
Principal Actor	Food recipient
Precondition	There is someone who wants to receive help
Trigger	The person asks if he can receive help from Cáritas
Main stage of success	<ol style="list-style-type: none"> <li>1. A person goes to Cáritas to ask for help.</li> <li>2. The Cáritas volunteer asks if that person is entitled to receive the Cáritas help.</li> <li>3. If not, the volunteer informs the person that they must ask their social worker for help.</li> </ol>
Extensions of the use case	2a User is registered. Then he receives the food on the delivery day. This goes to the use case 11.

Table 18: Description of use case 10

<b>Use case 11</b>	<b>Go pick up a delivery</b>
Principal Actor	Food recipient
Precondition	It is the day of the delivery and the food recipient is in the center of Cáritas.
Trigger	The food recipient wants to collect the food that belongs to him
Main stage of success	<ol style="list-style-type: none"> <li>1. The food recipient orders his batch of food.</li> <li>2. A Cáritas member introduces the information of the food recipient.</li> <li>3. The system validates if a food recipient exists or not.</li> <li>4. The Cáritas member starts a use case to create a delivery for the food recipient.</li> </ol>
Extensions of the use case	3a The food recipient doesn't exist in the database. The Cáritas member offers the possibility to fill in some data to contact a social worker. Use of case ends.

Table 19: Description of use case 11

<b>External use case 12</b>	<b>Investigate whether a food recipient can get help</b>
Principal Actor	Social worker
Precondition	There is a social worker
Trigger	One person asks if he can receive help from Cáritas
Main stage of success	<ol style="list-style-type: none"> <li>1. The social worker looks into the economic situation of the person's family.</li> <li>2. If that person can receive the help, he/she goes to use case 13.</li> </ol>
Extensions of the use case	2a. The user can't receive help, because his economic situation is good.

Table 20: Description of external use case 12

<b>Use case 13</b>	<b>Ask administrator of Cáritas to register a food recipient</b>
Principal Actor	Social worker
Precondition	There is a valid food recipient (Investigate whether a food recipient can get help. Use case of Table 20).
Trigger	We want to register a new food recipient into the Cáritas DB
Main stage of success	<ol style="list-style-type: none"> <li>1. The social worker asks the Cáritas administrator to register a new food recipient.</li> <li>2. Go to use case 15.</li> </ol>

Table 21: Description of external use case 13

<b>External use case 14</b>	<b>Ask administrator of Cáritas to delete food recipient</b>
Principal Actor	Social worker
Precondition	There is a food recipient registered into Cáritas
Trigger	We want to remove the help for a food recipient
Main stage of success	<ol style="list-style-type: none"> <li>1. The social worker asks the Cáritas administrator to remove the help for the food recipient.</li> <li>2. Go to use case 17.</li> </ol>

Table 22: Description of external use case 14

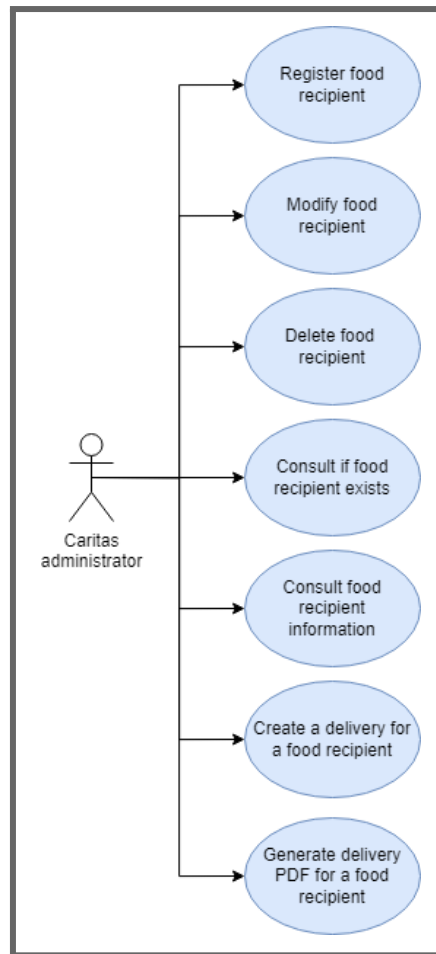


Figure 6: Use case diagrams for Cáritas administrator (I)

Use case 15	Register food recipient
Principal Actor	Cáritas administrator
Precondition	The social worker has validated that a person can receive help for their situation (Ask administrator of Cáritas to register a food recipient. Use case of Table 21).
Trigger	The user want to register a food recipient
Main stage of success	<ol style="list-style-type: none"> <li>1. The user enters the required data (name, last name, ID, phone number...).</li> <li>2. The user confirms the data entered.</li> <li>3. The system validates the data.</li> <li>4. The food recipient is registered in the DB.</li> </ol>
Extensions of the use case	<ol style="list-style-type: none"> <li>3a. The food recipient exists. The system displays a message informing that a food recipient with the same ID exists.</li> <li>3b. Wrong data.               <ul style="list-style-type: none"> <li>- 3b1. Non-valid email. The system displays a message informing that the email is not valid.</li> <li>- 3b2. Non-valid phone number. The system</li> </ul> </li> </ol>

	<p>displays a message informing that the phone number is not valid.</p> <ul style="list-style-type: none"> <li>- 3b3. Non-valid members. The system displays a message informing that kids between 0-2 years can't be equal or more than the total number of members.</li> <li>- 3b4. The ID is incorrect. The system displays a message informing the ID is not correct.</li> </ul>
--	--

Table 23: Description of use case 15

<b>Use case 16</b>	<b>Modify food recipient</b>
Principal Actor	Cáritas administrator
Precondition	The food recipient is already registered
Trigger	Some data of the food recipient is wrong or has changed
Main stage of success	<ol style="list-style-type: none"> <li>1. The user modifies the data.</li> <li>2. The user confirms the data modified.</li> <li>3. The system asks to validate the modifications.</li> <li>4. The user accepts.</li> <li>5. The system validates the data.</li> <li>6. The food recipient data is modified.</li> </ol>
Extensions of the use case	<p>3a. User doesn't want to modify food recipient data. Return to step 1.</p> <p>5a. If the ID is modified and the new ID exists, the system displays a message informing that a food recipient with the same ID exists.</p> <p>5b. Wrong data</p> <ul style="list-style-type: none"> <li>- 5b1. Non-valid email. The system displays a message informing that the email is not valid.</li> <li>- 5b2. Not-valid phone number. The system displays a message informing that the phone number is not valid.</li> <li>- 5b3. Non-valid members. The system displays a message informing that kids between 0-2 years can't be equal or more than the total number of members.</li> </ul>

Table 24: Description of use case 16

<b>Use case 17</b>	<b>Delete food recipient</b>
Principal Actor	Cáritas administrator
Precondition	The food recipient is already registered (Ask administrator of Cáritas to delete food recipient. Use case of Table 22).
Trigger	The social worker or the food recipient ask for their data deletion
Main stage of success	<ol style="list-style-type: none"> <li>1. The user presses the delete button.</li> <li>2. The system asks to validate the deletion.</li> <li>3. The user accepts</li> <li>4. The food recipient is deleted from the DB.</li> </ol>
Extensions of the use case	3a. User doesn't want to delete food recipient

Table 25: Description of use case 17

<b>Use case 18</b>	<b>Generate delivery PDF for a food recipient</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing consult food recipient information
Trigger	The user wants to create a delivery document for a food recipient
Main stage of success	<ol style="list-style-type: none"> <li>1. The user presses the generate PDF button.</li> <li>2. The system shows a file chooser.</li> <li>3. The user selects the pdf she/he wants to fill out.</li> <li>4. The user confirms the file selected.</li> <li>5. The system shows a list of the possible delivery dates where the food recipient received food this year.</li> <li>6. The user selects the dates she/he wants to import into the pdf.</li> <li>7. The user confirms the dates selected.</li> <li>8. The system shows a new screen with the pdf selected filled with the data of the deliveries for the dates selected</li> </ol>
Extensions of the use case	4a. The pdf to fill does not exist. The user cancels the use case.

Table 26: Description of use case 18



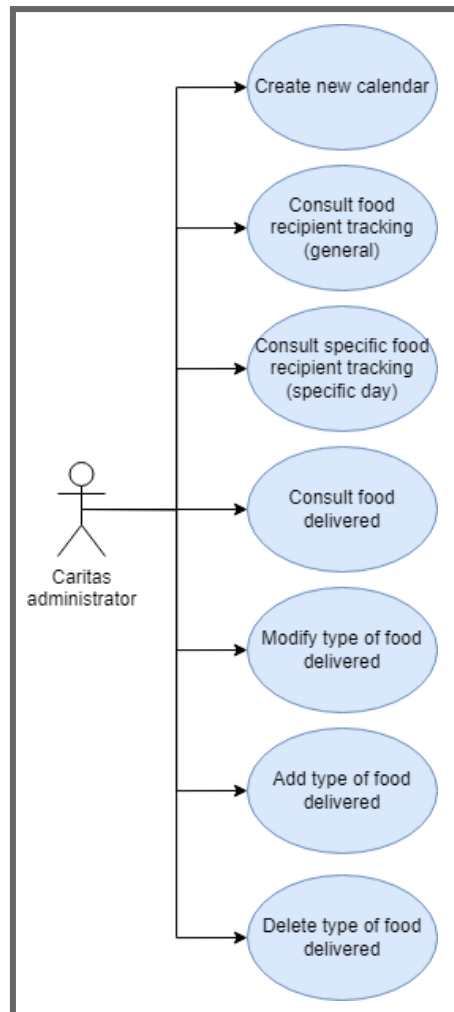


Figure 7: Use case diagrams for Cáritas administrator (II)

Use case 19	Create new calendar
Principal Actor	Cáritas administrator
Precondition	The user is viewing create calendar tab
Trigger	The user wants to create a new calendar for a specific family type and year
Main stage of success	<ol style="list-style-type: none"> <li>1. The user introduces the data (1st day of January, day of delivery, periodicity, year and family type. They can also add a month where no deliveries are made).</li> <li>2. The user confirms the data introduced.</li> <li>3. The system validates the data.</li> <li>4. The system auto generates a calendar and displays a list with every selected day (Monday, Tuesday...) within every periodicity (every week, every two weeks...) from the specified dates.</li> <li>5. The user can delete, modify or add new dates.</li> <li>6. The user confirms the data.</li> <li>7. The system validates the data.</li> </ol>

	8. The calendar is created.
Extensions of the use case	<p>3a. Calendar for that year + family type already exists</p> <p>6a. Wrong data.</p> <ul style="list-style-type: none"> <li>- 6a1. Day does not exist. The system displays a message informing that some data introduced contains a day that does not exist (i.e February 30).</li> <li>- 6a2. Wrong month. The system displays a message informing that some months are not well written (they need to be in the selected language).</li> <li>- 6a3. Wrong format. The system displays a message informing that the data must be month (the name of an existing month in the selected language) day (integer).</li> </ul>

Table 27: Description of use case 19

<b>Use case 20</b>	<b>Consult delivered food</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing the delivered food tab
Trigger	The user wants to consult the delivered food tab
Main stage of success	<ol style="list-style-type: none"> <li>1. The user may select from the dropdown the year and family type she/he wants to see.</li> <li>2. The system shows the delivered food for the year and family type selected.</li> </ol>

Table 28: Description of use case 20

<b>Use case 21</b>	<b>Modify type of delivered food</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing the delivered food tab with the year and family type she/he wants to modify
Trigger	The user wants to modify the name of delivered food
Main stage of success	<ol style="list-style-type: none"> <li>1. The user may change the name of the delivered food.</li> <li>2. The user confirms the data introduced.</li> <li>3. The user presses the save button.</li> <li>4. The system changes the name of that specific food.</li> </ol>
Extensions of the use case	1a. That food name already exists. The system displays a message informing that there is already a name like that.

Table 29: Description of use case 21

<b>Use case 22</b>	<b>Add type of delivered food</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing the delivered food tab
Trigger	The user wants to add a new type of delivered food.
Main stage of success	<ol style="list-style-type: none"> <li>1. The user presses the add row button.</li> <li>2. The system adds a new line for that new food type.</li> <li>3. If the user wants to modify it, she/he has to do use case 21 and/or 24.</li> </ol>

Table 30: Description of use case 22

<b>Use case 23</b>	<b>Delete type of delivered food</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing the delivered food tab
Trigger	The user wants to add a new type of delivered food
Main stage of success	<ol style="list-style-type: none"> <li>1. The user presses the delete row/s button.</li> <li>2. The system shows a list of all delivered food values.</li> <li>3. The user selects the food types she/he wants to delete.</li> <li>4. The user confirms the data selected.</li> <li>5. The system deletes those food types.</li> </ol>

Table 31: Description of use case 23

<b>Use case 24</b>	<b>Modify units of delivered food for a type of food and a specific day</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing the delivered food tab
Trigger	The user wants to modify the quantity of delivered food (the quantity is always counted in packages).
Main stage of success	<ol style="list-style-type: none"> <li>1. The user may select from the dropdown the year and family type she/he wants to modify.</li> <li>2. The system shows the delivered food for the year and family type selected.</li> <li>3. The user may change the quantity of delivered food for a specific day.</li> <li>4. The user confirms the data introduced.</li> <li>5. The user presses the save button.</li> <li>6. The system changes the quantity of those specific dates.</li> </ol>
Extensions of the use case	3a. The quantity is not numeric. The system displays a message informing that the data introduced is not valid and must be numeric.

Table 32: Description of use case 24

<b>Use case 25</b>	<b>Add units of delivered food for a type of food and specific day</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing the delivered food tab
Trigger	The user wants to add a quantity of delivered food for a day without value (the quantity is always counted in packages).
Main stage of success	<ol style="list-style-type: none"> <li>1. The user may select from the dropdown the year and family type she/he wants to modify.</li> <li>2. The system shows the delivered food for the year and family type selected.</li> <li>3. The user may add the quantity of delivered food for a specific day.</li> <li>4. The user confirms the data introduced.</li> <li>5. The user presses the save button.</li> <li>6. The system adds the quantity of those specific dates.</li> </ol>
Extensions of the use case	3a. The quantity is not numeric. The system displays a message informing that the data introduced is not valid and must be numeric.

Table 33: Description of use case 25

Use case 26	Register volunteer account
Principal Actor	Cáritas administrator
Precondition	The user is viewing volunteer account management tab
Trigger	The user wants to register a new volunteer account
Main stage of success	<ol style="list-style-type: none"> <li>1. The user introduces the data (username, password, email and if the volunteer account will be administrator)</li> <li>2. The user confirms the data.</li> <li>3. The system validates the data.</li> <li>4. The volunteer account is created.</li> </ol>
Extensions of the use case	<p>3a. Wrong data</p> <ul style="list-style-type: none"> <li>- 3a1. Username is empty. The system displays a message informing that the username field can not be empty.</li> <li>- 3a2. Password is empty. The system displays a message informing that the password field can not be empty.</li> <li>- 3a3. Email is empty. The system displays a message informing that the email field can not be empty.</li> <li>- 3a4. Volunteer exists. The system displays a message informing that a volunteer with the same username exists.</li> </ul>

Table 34: Description of use case 26

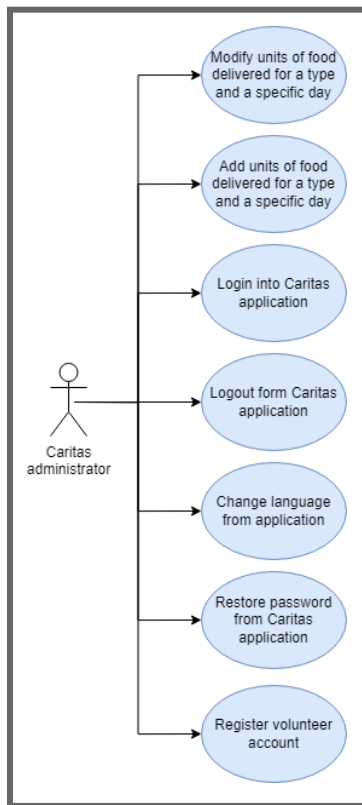


Figure 8: Use case diagrams for Cáritas administrator (III)

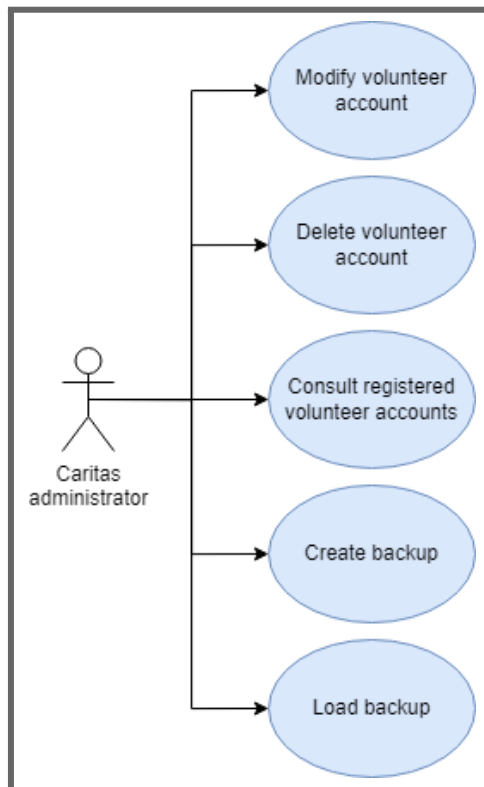


Figure 9: Use case diagrams for Cáritas administrator (IV)

<b>Use case 27</b>	<b>Modify volunteer account</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing volunteer account management tab and the member is registered
Trigger	The user wants to modify a registered volunteer
Main stage of success	<ol style="list-style-type: none"> <li>1. The user introduces the data (username, password, email and if the volunteer account will be administrator).</li> <li>2. The user confirms the data.</li> <li>3. The system validates the data.</li> <li>4. The member data is modified.</li> </ol>
Extensions of the use case	<p>3a. Wrong data</p> <ul style="list-style-type: none"> <li>- 3a1. Username is empty. The system displays a message informing that the username field can not be empty.</li> <li>- 3a2. Password is empty. The system displays a message informing that the password field can not be empty.</li> <li>- 3a3. Email is empty. The system displays a message informing that the email field can not be empty.</li> <li>- 3a4. Volunteer does not exist. The system displays a message informing that the volunteer does not exist.</li> </ul>

Table 35: Description of use case 27

<b>Use case 28</b>	<b>Delete volunteer account</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing volunteer account management tab and the volunteer is registered
Trigger	The user wants to delete a registered volunteer
Main stage of success	<ol style="list-style-type: none"> <li>1. The user introduces the username.</li> <li>2. The user confirms the data.</li> <li>3. The system validates the data.</li> <li>4. The member is deleted.</li> </ol>
Extensions of the use case	<p>3a. Wrong data</p> <ul style="list-style-type: none"> <li>- 3a1. Username is empty. The system displays a message informing that the username field can not be empty.</li> <li>- 3a2. The volunteer does not exist. The system displays a message informing that the volunteer does not exist.</li> </ul>

Table 36: Description of use case 28

<b>Use case 29</b>	<b>Consult registered volunteer accounts</b>
Principal Actor	Cáritas administrator
Precondition	The user is viewing volunteer account management tab and the volunteer is registered
Trigger	The user wants to consult all registered volunteers
Main stage of success	<ol style="list-style-type: none"> <li>1. The user presses the button "consult all volunteers".</li> <li>2. The system shows a list of all registered volunteers.</li> <li>3. The user presses a member.</li> <li>4. The system writes the data associated to that volunteer in the correspondent fields.</li> </ol>
Extensions of the use case	3a User does not want to consult registered volunteers.

Table 37: Description of use case 29

<b>Use case 30</b>	<b>Create backup</b>
Principal Actor	Cáritas administrator
Precondition	The user is in the backup tab
Trigger	The user wants to create a backup
Main stage of success	<ol style="list-style-type: none"> <li>1. The user presses the button "create backup".</li> <li>2. The system creates a backup in the folder backup.</li> </ol>

Table 38: Description of use case 30

<b>Use case 31</b>	<b>Restore backup</b>
Principal Actor	Cáritas administrator
Precondition	The user is in the backup tab and has previously created a backup
Trigger	The user wants to restore an old backup
Main stage of success	<ol style="list-style-type: none"> <li>1. The user presses the button restore backup.</li> <li>2. The user introduces the backup that she/he wants to restore.</li> <li>3. The system validates the file.</li> <li>4. The backup is loaded.</li> </ol>
Extensions of the use case	3a. Wrong file. The file is not an .mysql file or is corrupted.

Table 39: Description of use case 31



## 8.4. Non functional requirements

The non-functional requirements are also called the quality requirements; to analyze them we will use the template of Volere (O'Reilly website, 2022). The Volere template is divided into 16 sections. The first three correspond to the "Project Drivers", from 4th to 6th they correspond to the "Project Constraints", from 7th to 9th to the functional requirements, and from 10th to 16th to the non-functional requirements.

Non-functional requirements are the properties that the functions must have, such as performance and usability. These requirements are as important as the functional requirements for the product's success.

In table 40 we can see a list of all the non-functional requirements.

<p><b>10. Look and feel requirements</b></p> <table border="1"> <tr> <td>a. Appearance</td> </tr> <tr> <td>b. Style</td> </tr> </table>	a. Appearance	b. Style	<p><b>11. Usability and Humanity requirements</b></p> <table border="1"> <tr> <td>a. Ease of use</td> </tr> <tr> <td>b. Personalization and Internationalization</td> </tr> <tr> <td>c. Learning</td> </tr> <tr> <td>d. Understandability and Politeness</td> </tr> <tr> <td>e. Accessibility</td> </tr> <tr> <td>f. Convenience</td> </tr> </table>	a. Ease of use	b. Personalization and Internationalization	c. Learning	d. Understandability and Politeness	e. Accessibility	f. Convenience						
a. Appearance															
b. Style															
a. Ease of use															
b. Personalization and Internationalization															
c. Learning															
d. Understandability and Politeness															
e. Accessibility															
f. Convenience															
<p><b>12. Performance requirements</b></p> <table border="1"> <tr> <td>a. Speed and latency</td> </tr> <tr> <td>b. Safety-Critical</td> </tr> <tr> <td>c. Precision and accuracy</td> </tr> <tr> <td>d. Reliability and availability</td> </tr> <tr> <td>e. Robustness and fault-tolerance</td> </tr> <tr> <td>f. Capacity</td> </tr> <tr> <td>g. Scalability</td> </tr> <tr> <td>h. Longevity</td> </tr> </table>	a. Speed and latency	b. Safety-Critical	c. Precision and accuracy	d. Reliability and availability	e. Robustness and fault-tolerance	f. Capacity	g. Scalability	h. Longevity	<p><b>13. Operational and environmental requirements</b></p> <table border="1"> <tr> <td>a. Expected physical</td> </tr> <tr> <td>b. Wider environment</td> </tr> <tr> <td>c. Requirements for interfacing with adjacent systems</td> </tr> <tr> <td>d. Productization</td> </tr> <tr> <td>e. Release</td> </tr> <tr> <td>f. Backwards compatibility</td> </tr> </table>	a. Expected physical	b. Wider environment	c. Requirements for interfacing with adjacent systems	d. Productization	e. Release	f. Backwards compatibility
a. Speed and latency															
b. Safety-Critical															
c. Precision and accuracy															
d. Reliability and availability															
e. Robustness and fault-tolerance															
f. Capacity															
g. Scalability															
h. Longevity															
a. Expected physical															
b. Wider environment															
c. Requirements for interfacing with adjacent systems															
d. Productization															
e. Release															
f. Backwards compatibility															
<p><b>14. Maintainability and support requirements</b></p> <table border="1"> <tr> <td>a. Maintenance</td> </tr> <tr> <td>b. Supportability</td> </tr> <tr> <td>c. Adaptability</td> </tr> </table>	a. Maintenance	b. Supportability	c. Adaptability	<p><b>15. Security requirements</b></p> <table border="1"> <tr> <td>a. Access</td> </tr> <tr> <td>b. Integrity</td> </tr> <tr> <td>c. Privacy</td> </tr> <tr> <td>d. Audit</td> </tr> <tr> <td>e. Immunity</td> </tr> </table>	a. Access	b. Integrity	c. Privacy	d. Audit	e. Immunity						
a. Maintenance															
b. Supportability															
c. Adaptability															
a. Access															
b. Integrity															
c. Privacy															
d. Audit															
e. Immunity															
<p><b>16. Cultural requirements</b></p> <table border="1"> <tr> <td>a. Cultural market</td> </tr> <tr> <td>b. Cultural diversity and inclusion</td> </tr> </table>	a. Cultural market	b. Cultural diversity and inclusion	<p><b>17. Compliance requirements</b></p> <table border="1"> <tr> <td>a. Legal compliance</td> </tr> <tr> <td>b. Standards compliance</td> </tr> </table>	a. Legal compliance	b. Standards compliance										
a. Cultural market															
b. Cultural diversity and inclusion															
a. Legal compliance															
b. Standards compliance															

Table 40: Non functional requirements

### 8.4.1. Look and feel requirements

In this section we will examine the look and feel requirements that Caritapp meets. The look and feel requirements describe the intended spirit, the mood, or the style of the product's appearance.

<b>Identifier</b>	<b>1</b>
<b>Type</b>	<b>10a. Appearance</b>
Description	The interface design must be responsive and attractive.
Justification	Good design invites user to use the system.
Fit Criterion	At least 75% of users say they like the interface design.
Priority	High
User satisfaction	5

Table 41: Non functional requirement 1

### Usability and humanity requirements

The usability and humanity requirements make the product conform to the user's abilities and expectations of the usage experience.

<b>Identifier</b>	<b>2</b>
<b>Type</b>	<b>11a. Ease of Use</b>
Description	How easy it is for the intended users of product to operate it.
Justification	The user must know in each moment what she/he is doing, and in case of error, know the exact error and how to solve it.
Fit Criterion	At least 75% of users claim that they have had no more than two unsolved doubts and have made no or more than two errors while using the system for a couple of weeks.
Priority	High
User satisfaction	4

Table 42: Non functional requirement 2

<b>Identifier</b>	<b>3</b>
<b>Type</b>	<b>11b. Internationalization</b>
Description	It is the process of designing a software application so that it can be adapted to various languages and regions without engineering changes.
Justification	The system is designed for any Cáritas in Spain.
Fit Criterion	At least 95% of users claim to have found their preferred language.
Priority	High
User satisfaction	5

Table 43: Non functional requirement 3

<b>Identifier</b>	<b>4</b>
<b>Type</b>	<b>11c. Learning</b>
Description	How easy it should be to learn to use the product (learning curve).
Justification	The system must be easy to understand, making the user invert the shortest time learning how to use it.
Fit Criterion	At least 65% of users state that they have not consulted the help menu or asked questions to third parties while using the system for a couple of weeks.
Priority	High
User satisfaction	4

Table 44: Non functional requirement 4

### 8.4.2. Performance requirements

In this section we will look into the performance requirements that Caritapp meets. Performance requirements define how well the software system accomplishes certain functions under specific conditions.

<b>Identifier</b>	<b>5</b>
<b>Type</b>	<b>12a. Speed and Latency</b>
Description	How fast the system is.
Justification	If the system is too slow, the user may think the system is not working or even stop using the application, because of the long waits.
Fit Criterion	The response time of the system is in no case greater than eight seconds and, under normal conditions (no problem with the web server or with the user's Internet connection), it takes between one and four seconds.
Priority	Medium-High
User satisfaction	4

Table 45: Non functional requirement 5

<b>Identifier</b>	<b>6</b>
<b>Type</b>	<b>12d. Reliability and Availability</b>
Description	The system may be accessible 24/7.
Justification	The user may use it at any time.
Fit Criterion	During the testing phase and at different times of the day, it is checked that the system works correctly.
Priority	High
User satisfaction	4

Table 46: Non functional requirement

<b>Identifier</b>	<b>7</b>
<b>Type</b>	<b>12e. Robustness</b>
Description	The capability to cope with unknown errors during execution and provide the system services all the same.
Justification	All the unknown errors must be identified.
Fit Criterion	The system returns messages of all possible errors that may arise in the system.
Priority	High
Validation	To validate the robustness of the application, several tests have been done to make the system fail, gradually adding error messages, in case something is not right, and fixing all the errors that were appearing. In any case, there will be a small testing phase at the launch of the application, in which all the defects found will be fixed.
User satisfaction	4

Table 47: Non functional requirement 7

<b>Identifier</b>	<b>8</b>
<b>Type</b>	<b>12e. Fault-tolerance</b>
Description	The capability to tolerate a certain set of errors defined during the development of the system.
Justification	It lets the user the chance to do something wrong without breaking/collapsing the system.
Fit Criterion	All errors that the user can make are controlled.
Priority	High
Validation	The user is left with the ability to make mistakes and if he does, the system will warn him that what he has done is wrong and will give him a hint to solve the error.
User satisfaction	4

Table 48: Non functional requirement 8

<b>Identifier</b>	<b>9</b>
<b>Type</b>	<b>12f. Capacity</b>
Description	How many users can be simultaneously using the system.
Justification	The user may access the application at any time, without having to wait a long time.
Fit Criterion	At the initial milestone, a maximum of 5 users will use the system simultaneously.
Priority	High.
Validation	The default limit is 151 parallel connections for MySQL Database.
User satisfaction	5

Table 49: Non functional requirement 9

<b>Identifier</b>	<b>10</b>
<b>Type</b>	<b>12.g Scalability</b>
Description	If the system is scalable, more users will be able to use the system. In our case, also more Cáritas centers may use this system.
Justification	Any center may use this system.
Fit Criterion	The system facilitates the number of active users to be more numerous.
Priority	Medium
Validation	If we have a dedicated server we can increase the number of parallel connections to around 10000.
User satisfaction	5

Table 50: Non functional requirement 10

### 8.4.3. Maintainability and support requirements

In this section we will study the maintainability and support requirements that Caritapp meets. A maintainable system must be capable of being maintained cost-effectively over its expected lifetime.

<b>Identifier</b>	<b>11</b>
<b>Type</b>	<b>14a. Maintenance</b>
Description	It is the ease with which faults in a software system can be found and properly fixed.
Justification	All faults must be found and fixed.
Fit Criterion	During the first months of life, comprehensive maintenance will be carried out.
Priority	High
Validation	During the first months of the product launch there will be a maintenance phase, generally focused on correcting all defects reported by volunteers.
User satisfaction	5

Table 51: Non functional requirement 11

#### 8.4.4. Security requirements

In this section we will check the security requirements that Caritapp meets. A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

<b>Identifier</b>	<b>12</b>
<b>Type</b>	<b>15a. Access</b>
Description	How well the system is safeguarded against deliberate and intrusive faults from internal and external sources.
Justification	Confidentiality of data must be ensured.
Fit Criterion	During the testing phase, the identification processes and permissions of each type of user are checked, some of which are exclusive and others are common to other users.
Priority	High
User satisfaction	5

Table 52: Non functional requirement 12



<b>Identifier</b>	<b>13</b>
<b>Type</b>	<b>15b. Integrity</b>
Description	How well the data are maintained by the software system in terms of accuracy, authenticity, and without corruption.
Justification	Integrity of data must be ensured.
Fit Criterion	<ul style="list-style-type: none"> <li>- The administrator (or preferably the system automatically) must periodically make backup copies of the database.</li> <li>- The system does not accept incorrect data and notifies the user how to enter it in order for it to be valid.</li> </ul>
Priority	High
User satisfaction	5

Table 53: Non functional requirement 13

<b>Identifier</b>	<b>14</b>
<b>Type</b>	<b>15c. Privacy/Confidentiality</b>
Description	Confidentiality is the degree to which the software system protects sensitive data and allows only authorized access to the data
Justification	The food recipients data must be confidential, as it is sensitive data
Fit Criterion	The system guarantees the confidentiality of the data.
Priority	High
User satisfaction	4

Table 54: Non functional requirement 14

## 9. Design

In this section we will talk about the design of the application, specifically which design pattern is used and a brief explanation of the relational tables we have.

### 9.1. Patterns used

The Model-View-Controller (MVC) pattern is based on the division of one application into three independent and interconnected parts (model, view and controller, as the name implies) in order to separate the rendering internal of the external data. This decoupling makes it possible to modify one part without affecting the others, and makes modularity, reusability, maintenance and scalability easier.

- **Model:** The model manages access to the database.
- **View:** It encompasses the information that is sent to the client and the interaction mechanisms with them.
- **Controller:** It acts as an intermediary between the model and the view, managing the flow of information between them and the transformations to adapt the data to the needs of each one.

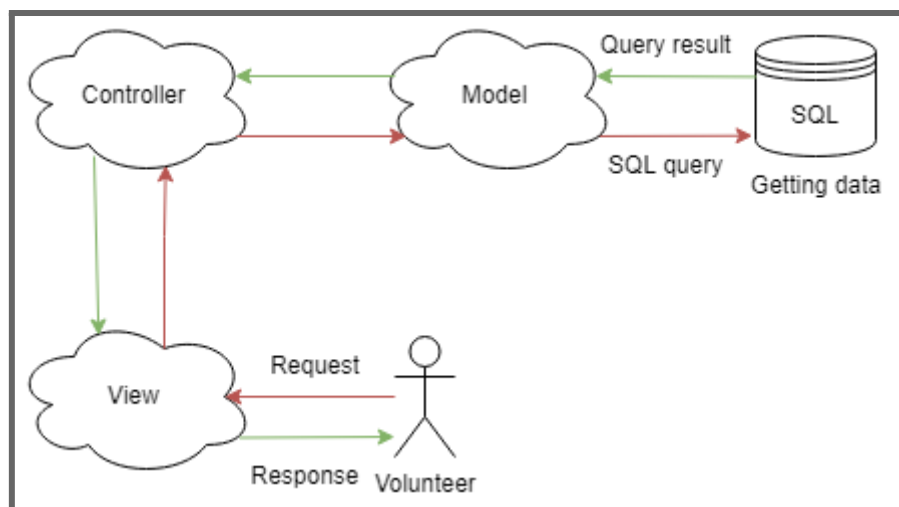


Figure 10: Caritapp MVC

### 9.2. Relational tables

The Unified Modeling Language (UML) Class diagram is a graphical notation used to construct and visualize object oriented systems. In the UML a class diagram is a type of static structure diagram that describes the structure of a system by showing:

classes, the attributes of the classes, operations (or methods) and the relationship among objects. In Figure 11 we can see the relationship between the tables.

To create a food distribution we need a variety of food and a date. There can not be more than two food distributions for the same date. Once the food distribution is set, we can generate a delivery involving a volunteer and a food recipient (who will receive the food). For a food distribution we can make several deliveries, but never to the same food recipient. A calendar is a set of food distributions. All food distribution must be of the same familyType of the calendar. For each year/familyType only one calendar can exist.

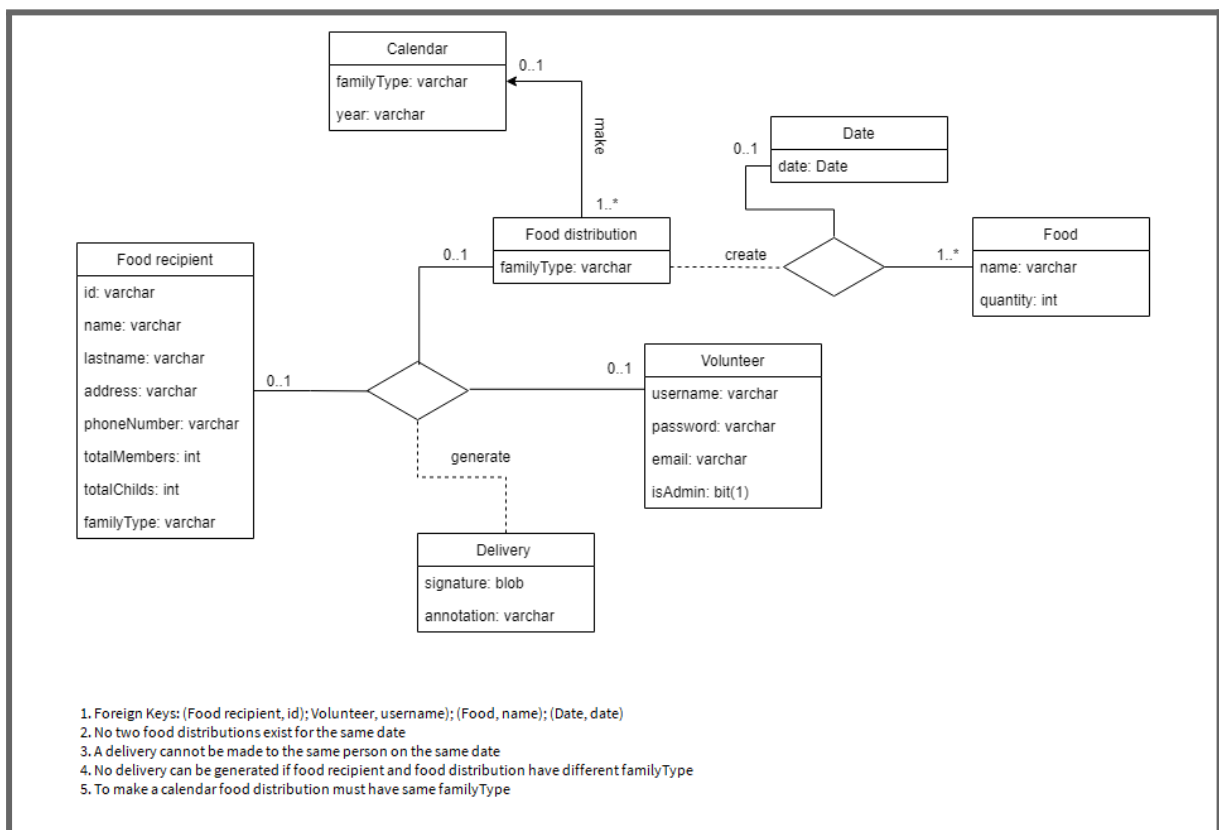


Figure 11: UML class diagram

# 10. Implementation

## 10.1. Technologies and languages used

The language used in this project is Java. Java is a language that is platform independent, which means that compiled code can run on anything that has a JRE (Java Runtime Environment). In particular, the application has been developed with Java SE-16. SE stands for Standard Edition. Java SE's API provides the core functionality of the Java programming language. It defines everything from the basic types and objects of the Java programming language to high-level classes that are used for networking, security, database access, graphical user interface (GUI) development, and XML parsing (Hartman, James, 2020).

Java also provides you with OOP (Object-oriented programming), a computer programming model that organizes software design around data, or objects, rather than functions and logic:

- Encapsulation
- Inheritance
- Polymorphism

Database and database management will be developed over MySQL. MySQL is an open-source relational database management system (RDBMS), which means a digital database system to maintain relational databases. So we will manage data using a structure and language consistent with first-order predicate logic, a language that uses intuitive syntax.

## 10.2. Development tools

- Eclipse: it is the IDE that has been chosen to develop the project. Eclipse is an intuitive IDE, this will save us some time learning the IDE. At the same time it is an IDE that allows us to carry out our goal. In this case we will use the package Eclipse IDE for Enterprise Java and Web Developers. This package contains tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and more.

- MySQL Workbench 8.0: in order to view the tables in the database in a convenient and intuitive way.

It is important to know that, in order for this application to work, a SE 16 is needed (Oracle website, 2022) or a higher version of Java installed, and also MySQL version 8.0 (MySQL website, 2022).

### 10.3. Security and data protection

Nowadays security is an important issue. Even more with the treatment of sensitive data, as is the case of Cáritas. MySQL uses security based on Access Control Lists (ACLs) for all connections, queries, and other operations that users can attempt to perform. An ACL is a list of rules that specifies which users or systems are granted or denied access to a particular object or system resource. Access control lists are also installed in routers or switches, where they act as filters, managing which traffic can access the network (Ben Lutkevich, 2022). Therefore, the database can only be accessed from a local network of the Cáritas server.

There is also support for SSL-encrypted (Secure Sockets Layer) connections between MySQL clients and servers (MySQL website, 2022). To provide a high degree of privacy, SSL encrypts data that is transmitted over the web. Therefore, anyone who tries to intercept this data will find a confusing mix of characters, which will be very difficult to decipher (Cloudfare website, 2022).

### 10.4. Views

In this section we will see all the views of the application. In all those sections where it is possible for the system to return a series of error messages, there will be a table with the identifier of the error and its message. All error messages have the same structure as the error in Figure 12.

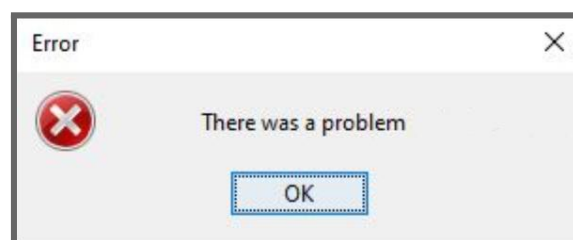


Figure 12: Error message display

### 10.4.1. Login screen

The first screen we see when executing the application is the login screen. A smooth login experience gets users onto your site without friction, but a poor design can be off-putting and leave a bad impression before they have even signed in.

In this screen we can see different sections:

- **Language selector** in the upper right corner. This selector is used to change the language of the system between Catalan, Spanish and English.
- **User and password field**: these fields are required to logging in. You must introduce a valid and existent user/password combination in order to log in. By default an admin user (user: admin, password:admin) and member user (user: member, password: member) are created. New user registration must be done by an admin user inside the application.
- **Show password checkbox**: it shows you the password if enabled, otherwise the password is kept as secret with asterisk characters.
- **Login button**: if data is correct, it shows a successful login message (Figure 13) and logs in into the system. Otherwise it shows an error message, incorrect user message (Table 55 - Message error LS-1), or password not correct message (Table 55 - Message error LS-2).
- **Clean button**: it cleans the data introduced in the user and password fields.
- **Forgot password button**: in case you forgot your password you can ask for a new one. A popup is shown asking for your email address and username. When introduced, the system verifies if the email is registered, and if so, an email is sent with the new password for the user. If the email is not registered the system shows an error message (Table 55 - Message error LS-3).
- **Information button**: it shows an info panel with the data described above. This panel details a short description of each of the buttons. (Figure 17).



Figure 13: Login screen

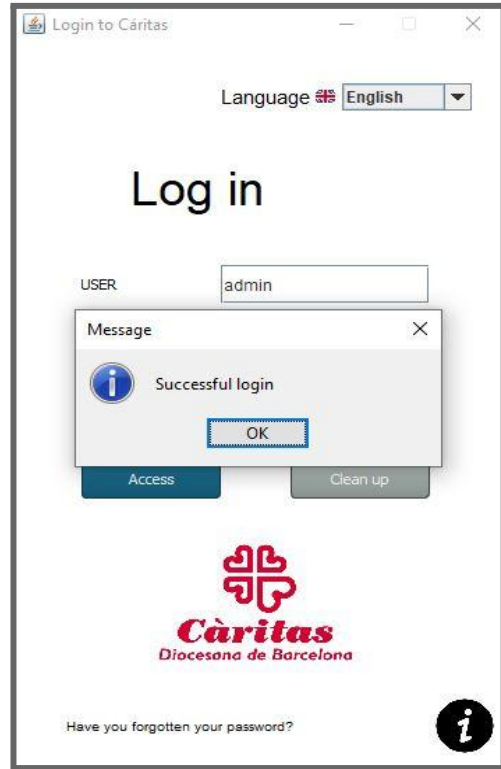


Figure 14: Login screen - login successfully

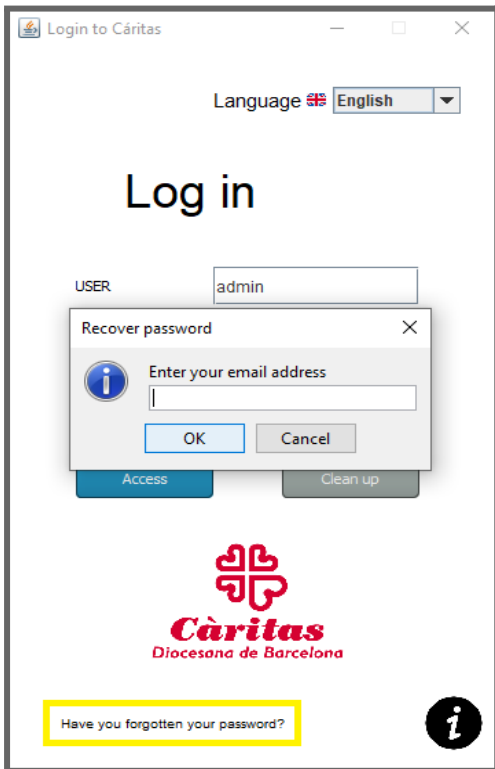


Figure 15: Login screen - recover password



Figure 16: Login screen - showing password

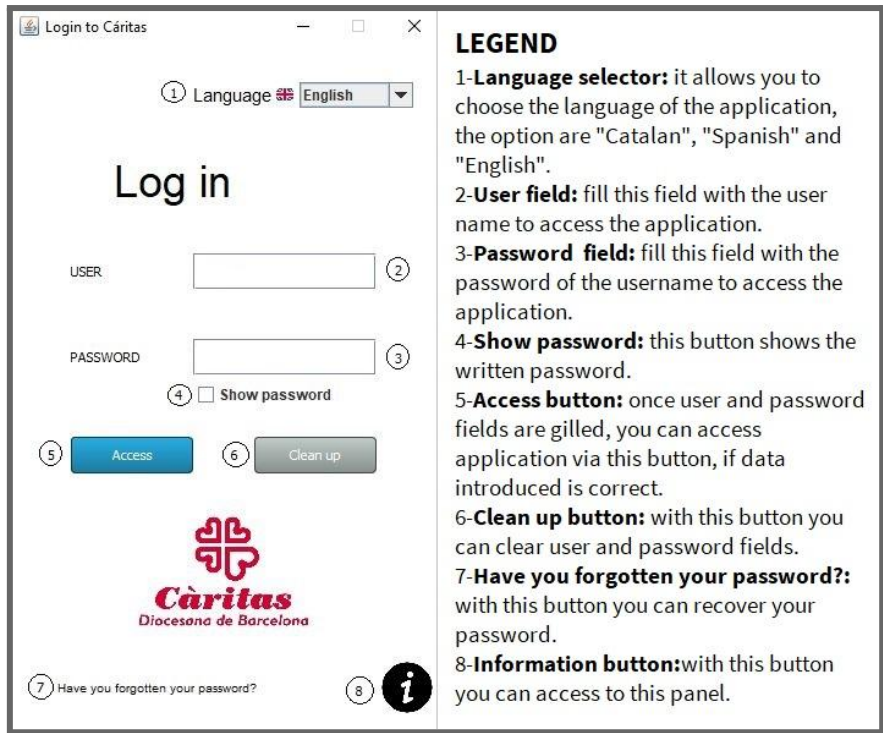


Figure 17: Login screen: information panel

Error ID	Error description
LS-1	Entered user does not exist.
LS-2	The entered password is not correct.
LS-3	There is no user registered with that email.

Table 55: Login screen (LS) list of errors

### 10.4.2. Generic screen

Figure 18 shows an example of the generic screen template. This is shown at the top part of each screen of the application. We will call this the generic screen, although you could consider it the header or the borders. In this generic screen we have different sections.

The first, at the left side, shows who is logged in. In this section we have a "Hello username".

The second, at the right side contains the following buttons:

- **Home:** this button will take us to the menu page, the initial screen.
- **Back:** this button will return us to the previous page we were on.



- **Logout:** this button will allow us to log out.

In this section also appears the Cáritas logo.



Figure 18: Generic screen

### 10.4.3. Menu screen

Figure 19 shows an example of the menu screen as an administration, Figure 20 shows an example of the menu screen as a member. This screen is the main screen, from which you can access all the functions of the system. There are different buttons visible, that are activated based on the privileges of the person who has logged in.

- **Register new food recipient:** This button takes you to the food recipient creation screen.
- **Food recipient list:** This button takes you to the search screen, where you can find all registered food recipients.
- **Make a delivery:** This button asks you for some information, such as ID or name/surname of the food recipient you want to create the delivery. If only one food recipient is found, the delivery screen for this person is shown. Otherwise, if more than one possible food recipient matches the search, a list of possible food recipients is shown.
- **Food delivered:** This button takes you to the delivered food screen in which you can specify the number and type of food that will be distributed each day of delivery.
- **Create calendar:** This button takes you to the create calendar screen in which you can create delivery calendars for each year and each family type (A or B).
- **Generate pdf documents:** This button lets you create in batch, the pdfs of all food recipients of family type A or B. All pdfs created are stored in the Cáritas folder generated by the system, with day\_month\_year\_familyType.
- **Volunteers account management:** This button takes you to the volunteer accounts management screen in which you can delete volunteers accounts, modify them, grant them different privileges or simply consult the information of accounts.

- **Backup management:** This button takes you to the backup management screen in which you can create backups and also load old ones. Generated backups are stored in the Cáritas folder generated by the system, with backup\_day-month-year\_hour-minutes.
- **Food recipient tracking:** This button takes you to the food recipient tracking screen in which you can see the tracking of food recipients. The food recipient tracking screen also allows the user to access a specific delivery to check information, such as delivered food, comments and signature.

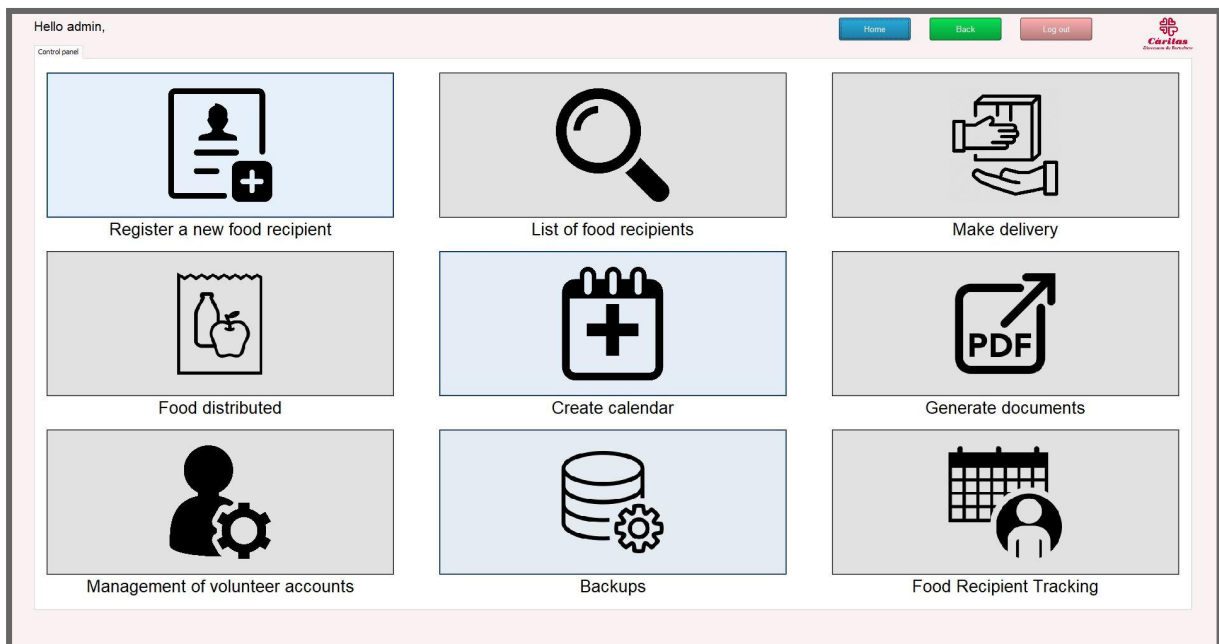


Figure 19: Menu screen with admin privileges

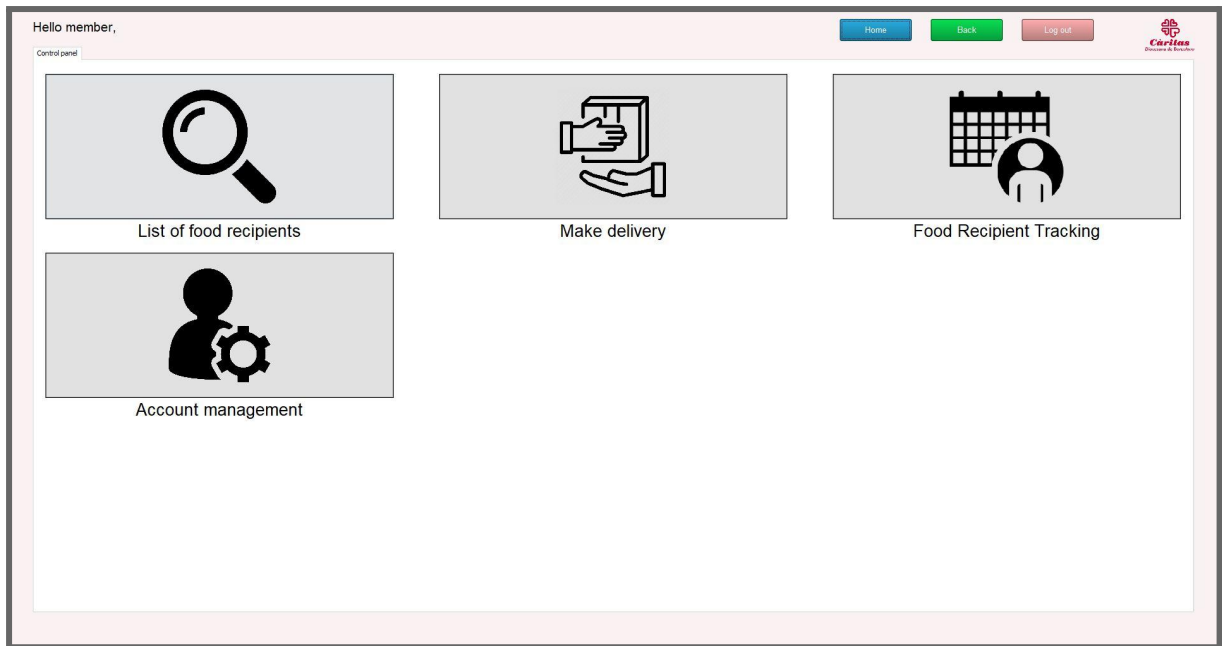


Figure 20: Menu screen as a member

#### 10.4.4. Food recipient registration screen

Figure 21 shows an example of the food recipient registration screen. Only the administrator role has access to this functionality. This screen lets the user register new food recipients. To register a food recipient, the information must be entered in the different fields:

- **Name:** The name of the family representative. This field accepts only alphabetic characters and the Hyphen symbol “-” for composed names. The result introduced in the table of database is passed to Proper Case, meaning that if someone introduces “JUan maNuel” it is registered as “Juan Manuel”.
- **Surname:** The surname or surnames of the family representative. This field accepts characters between “a-z” and “A-Z” and the Hyphen symbol “-” for composed surnames. The result introduced in the table of database is passed to Proper Case, meaning that if someone introduces “rODRiGuez pErEz-ARa” it is registered as “Rodriguez Perez-Ara”.
- **DNI/NIE/Passport:** The DNI, NIE or passport of the family representative. This field accepts numeric and alphabetical characters.

A few validations are done in order to check whether the DNI and the NIE are correct. In case the DNI or NIE is wrong, some exceptions may be thrown.

- DNI/NIE last letter is not correct: The system throws an exception and shows a message informing that the letter or the numbers associated to that letter are not correct (Table 56 - Message error RFR-1).
- NIE first letter is not X,Y or Z. The system throws an exception and shows a message informing that the first digit must be the character X, Y or Z (Table 56 - Message error RFR-2).
- DNI/NIE not contains 9 digits. The system throws an exception and shows a message informing if the DNI or NIE contains more or less than 9 digits (Table 56 - Message error RFR-3 and RFR-4).
- DNI first 8 digits are not numeric. The system throws an exception and shows a message informing that the first 8 digits must be numeric (Table 56 - Message error RFR-5).
- NIE from 2 to 8 digits are not numeric. The system throws an exception and shows a message informing that from second digit to eight digit must be numeric (Table 56 - Message error RFR-6).

As for the passport, it is not possible to do the same, for there is no legal way for a civilian to know.

The result introduced in the table of the database is passed to UpperCase, meaning that if someone introduces "XXXXXXXXh" (where X is a number between 0 and 9) it is registered as "XXXXXXXXH".

- **Telephone:** The mobile or landline of the family representative. This field only accepts numeric characters, space and "+" characters.
- **Email:** The email of the family representative. It is not necessary to fill this field. If introduced it must be a valid email. This is done by matching the email with a regex pattern.
- **Address:** The address where the family lives. This field accepts any character.
- **Total members:** Number of members in the family unit. This field accepts only numeric characters. This field can not be empty, if so it will throw an error exception. (Table 56 - Message error RFR-7).
- **Number of children:** Number of children between 0 and 2 years old within the family unit. This field accepts only numeric characters. The number of children can't be equal or higher than total members, if so it will throw an error exception. (Table 56 - Message error RFR-8).

- **Family type:** This field is divided into two groups, “A” and “B”. “A” are families with 3 or less members and “B” families with 3+ members. This field is filled automatically when the total members field is filled.

There are also 2 buttons:

- **Register:** When we press it, the system performs the necessary validations. If everything is correct, the new food recipient is registered. In case of any error, the system displays the corresponding message.
- **Clean:** When we press it, all the fields are cleared.

Figure 21: Register new food recipient screen

Error ID	Error description
RFR-1	Either the last DNI/NIE letter is not correct, or there is a wrong digit.
RFR-2	The first letter of the NIE must be X,Y or Z.
RFR-3	A DNI or NIE cannot have less than 9 digits.
RFR-4	A DNI or NIE cannot have more than 9 digits
RFR-5	The first 8 digits of the DNI must be numbers.
RFR-6	The central 7 digits of the NIE must be numbers.
RFR-7	There is an empty field that should be filled.
RFR-8	The number of members of the family unit cannot be equal to or less than the number of children under 2 years of age.

Table 56: Register new food recipient (RFR) list of errors

### 10.4.5. Food recipient list screen

This screen is accessible by every *Cáritas* volunteer. Figure 22 shows a list of users matching the searched criteria.

For confidential reasons only certain data is visible, such as the ID (DNI, NIE or passport), the name of the food recipient, the surname and the family type. Only administrators may see more information at the consult food recipient information screen.

This screen has a search engine to easily find food recipients. This search engine allows searches with multiple words, regardless of their order. In other words, searching for "Domínguez Pérez" is the same as searching for "Pérez Domínguez". At the same time, it allows non-exact searches, that is, a person named "Javi Pérez Domínguez" can be found by searching for "Ja Pe Do" and also for "J a vi Perez", for example. In any case, it does not allow us to find people if there are words or letters that do not exist: "Javi Pérez Rodriguez" would not have a result if we are looking for "Javi Pérez Domínguez". The only fields you can search for are ID (DNI, NIE or passport), name and surname.

Once the user has been found, this screen allows access to their information by double clicking (this function is disabled for users who are not administrators, in which case the system will redirect them to the delivery screen).

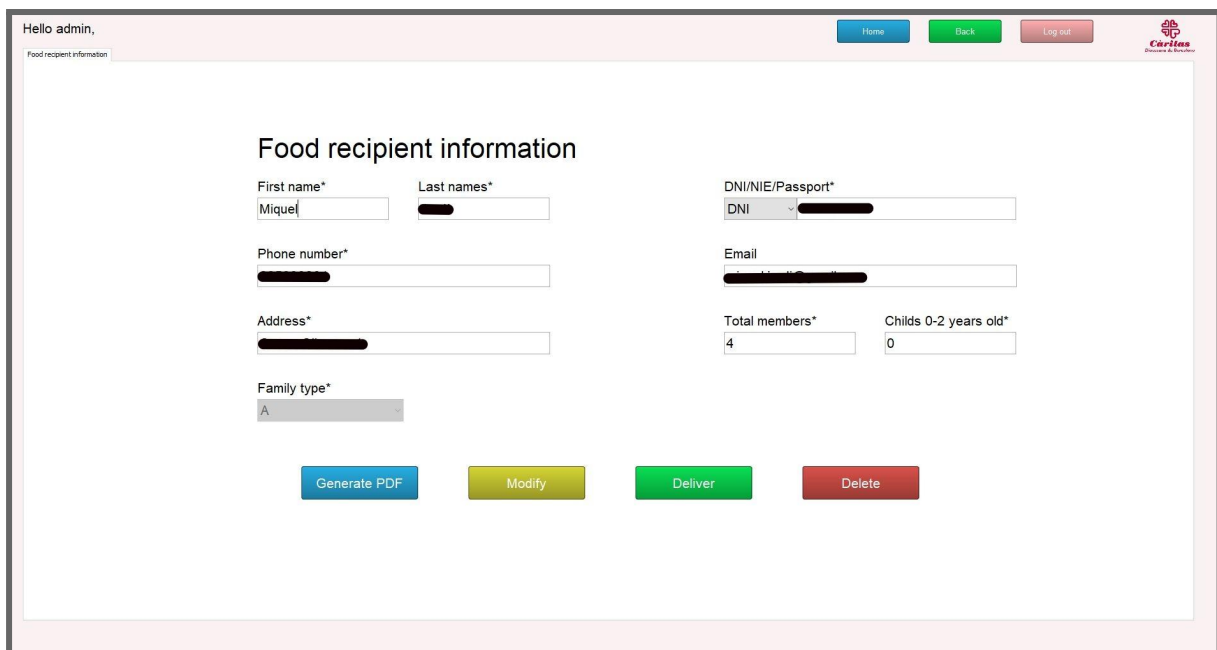
DNI/NIE/Passport	First name	Last names	Family type
██████ 11H	Miquel	██████	A
██████ 22B	Pol	██████	B
██████ 78A	Rosa Maria	██████	B
██████ 78Z	Alex	██████████████	B
██████ 94Q	Laura	██████	A
██████ 04Z	Pepe	██████	B
██████ 92A	Xavi	██████	B
██████ 23P	Xavi	██████	A
██████ 32L	Joan	██████	B
██████ 50S	David	██████	B
██████ 93H	Jaime	██████	B
██████ 48S	Miriam	██████	B
██████ 15S	Joan	██████	B
██████ 15X	Joan Manel	██████	B
██████ 29P	Cristina	██████████	B
██████ 49A	Miquel	██████	B
██████ 94D	Carles	██████	B
██████ 31K	Assumpta	██████	B
██████ 86P	Eric	██████	B

Figure 22: Food recipient screen list

## 10.4.6. Food recipient information screen

Figure 23 shows an example of the food recipient information screen. We will access this screen from the list of food recipients and only if we have administrator privileges. This screen includes the same fields as the Food Receiver Registration screen plus:

- **Generate PDF button:** This button allows you to generate a PDF, with the data of up to 7 days (Since the document "delivery note" only has space for 7 days, as shown in Figure 24) and the information of the selected food receiver, in order to be able to report how the food received by the European Aid Fund has been distributed.
- **Modify button:** This button allows you to modify the data of the food receiver. It follows the same rules as generating a new food recipient.
- **Make a delivery button:** This button takes you to the delivery screen.
- **Delete button:** This button removes a user from the database.



The screenshot displays a web interface for managing food recipients. At the top left, it says "Hello admin," and at the top right, there are navigation buttons for "Home", "Back", and "Log out", along with the Caritas logo. The main content area is titled "Food recipient information" and contains a form with the following fields:

- First name\*: Mique[...]
- Last names\*: [...]
- DNI/NIE/Passport\*: DNI - [...]
- Phone number\*: [...]
- Email: [...]
- Address\*: [...]
- Family type\*: A
- Total members\*: 4
- Childs 0-2 years old\*: 0

At the bottom of the form, there are four action buttons: "Generate PDF" (blue), "Modify" (yellow), "Deliver" (green), and "Delete" (red).

Figure 23: Food recipient information screen

### 10.4.6.1. Generate PDF

If you decide to generate a pdf only for a specific food recipient, you must enter the pdf document on which you want to enter the data (as seen in Figure 25), then you

have to select up to seven delivery dates that you want to import (as seen in Figure 26). If you put more than seven, the system will throw an error exception. (Table 57 - Message error FRI-1).

Once entered, a new screen will open with the resulting PDF. We can save this PDF or send it to print from this screen (Figure 27).

**NOTA DE ENTREGA DE ALIMENTOS 2022**

**NOMBRE DE LA OAR**

**PROGRAMA DE AYUDA ALIMENTARIA A LAS PERSONAS MÁS DESFAVORECIDAS 2022**

**ALIMENTOS GRATUITOS - PROHIBIDA SU VENTA**

**DIRECCIÓN DE LA OAR**

**DATOS DE LA UNIDAD FAMILIAR**

<b>Nombre y apellidos del representante de la unidad familiar:</b>			
<b>DNI/NIE/Pasaporte: (1)</b>		<b>Teléfono:</b>	
<b>Domicilio:</b>			
<b>Localidad:</b>		<b>CP:</b>	

<b>MIEMBROS DE LA UNIDAD FAMILIAR</b>	<b>Niños 0-2 (ambos inclusive)</b>	<b>Nº Miembros de otras edades</b>	<b>TOTAL MIEMBROS UNIDAD FAMILIAR</b>

Alimentos	NÚMERO DE ENVASES RECIBIDOS POR UNIDAD FAMILIAR EN LA FECHA DE RECOGIDA INDICADA: (2)							
	/ /2022	/ /2022	/ /2022	/ /2022	/ /2022	/ /2022	/ /2022	/ /2022
Arroz blanco								
Alubia cocida								
Conserva de atún								
Pasta alimenticia tipo macarrón								
Tomate frito en conserva								
Cajetas								
Macedonia de verduras en conserva								
Fruta en conserva								
Cacao soluble								
Tamitos infantiles con pollo								
Tamitos infantiles de fruta								
Leche entera UHT								
Aceite de oliva								
RECIBI Fdo. por el representante de la unidad familiar								

**Recepción de información de medidas de acompañamiento.**

De conformidad a lo dispuesto en el artículo 13. b), de la Orden AAA/2205/2015, de 15 octubre, de bases reguladoras.

Como representante de la Unidad Familiar, manifiesta haber recibido, como medida de acompañamiento, la información sobre los recursos sociales más cercanos con fecha: \_\_\_\_ de \_\_\_\_ de 202\_\_.

Recibi:  
El representante de la Unidad Familiar

En cumplimiento del Reglamento (UE) 2016/79 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, y de la normativa vigente en materia de protección de datos, ponemos en su conocimiento que el responsable del tratamiento de sus datos es esta Organización Asociada de Reparto (OAR) que le suministra los alimentos FEAD.

<b>Responsable del tratamiento</b>	OAR
<b>Finalidad del tratamiento</b>	Ayuda alimentaria del fondo de Ayuda Europea para las personas más desfavorecidas en España (FEAD 2014-2020). Reparto de alimentos a las personas desfavorecidas autorizadas a participar en el Programa FEAD. <ul style="list-style-type: none"> <li>Reglamento nº 223/2014 del Parlamento Europeo y del Consejo, de 11 de marzo de 2014, relativo al fondo de Ayuda Europea para las personas más desfavorecidas.</li> <li>Decisión de ejecución de 17 de diciembre de 2014 de la Comisión Europea por la que se aprueba el Programa Operativo sobre ayuda alimentaria para la solidaridad de ayudo del FEAD en España (CC 2014S05MOP000).</li> <li>Orden AAA/2202/2015, de 15 de octubre, por la que se establecen las bases reguladoras de las subvenciones a las organizaciones asociadas de distribución, para sufragar los gastos administrativos, de transporte y almacenamiento de los alimentos en el marco del programa operativo sobre ayuda alimentaria del Fondo de Ayuda europea para las personas más desfavorecidas.</li> </ul>
<b>Legitimación del tratamiento</b>	FEAGA, Organizaciones Asociadas de Distribución designadas por el FEAGA y Unidades de control responsables de la supervisión del programa.
<b>Destinatarios de cesiones o transferencias de datos</b>	Podrá ejercer los derechos de acceso, rectificación, cancelación y oposición, así como otros derechos recogidos por el Reglamento 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, dirigiendo escrito a esta entidad social.
<b>Derechos de los interesados</b>	

Figure 24: Food delivery note 2022 in spanish

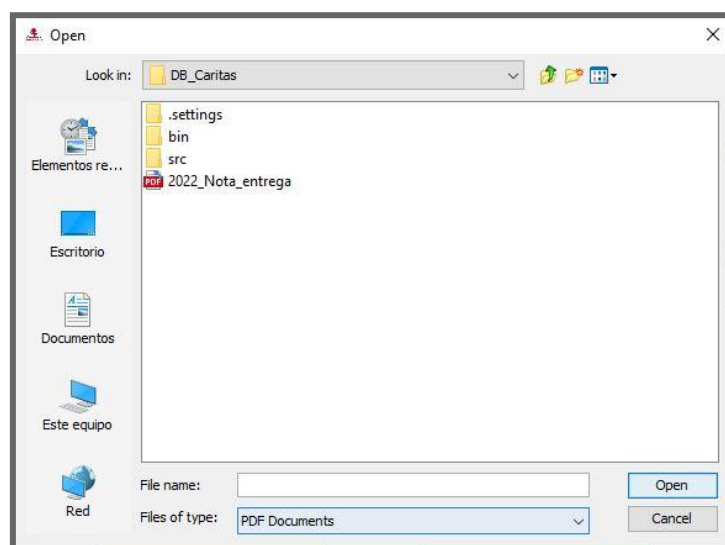


Figure 25: Food recipient information screen - Select file modal





Figure 26: Food recipient information screen -  
Select day modal

Error ID	Error description
FRI-1	Only up to 7 options can be selected

Table 57: Food recipient information (FRI) list of errors

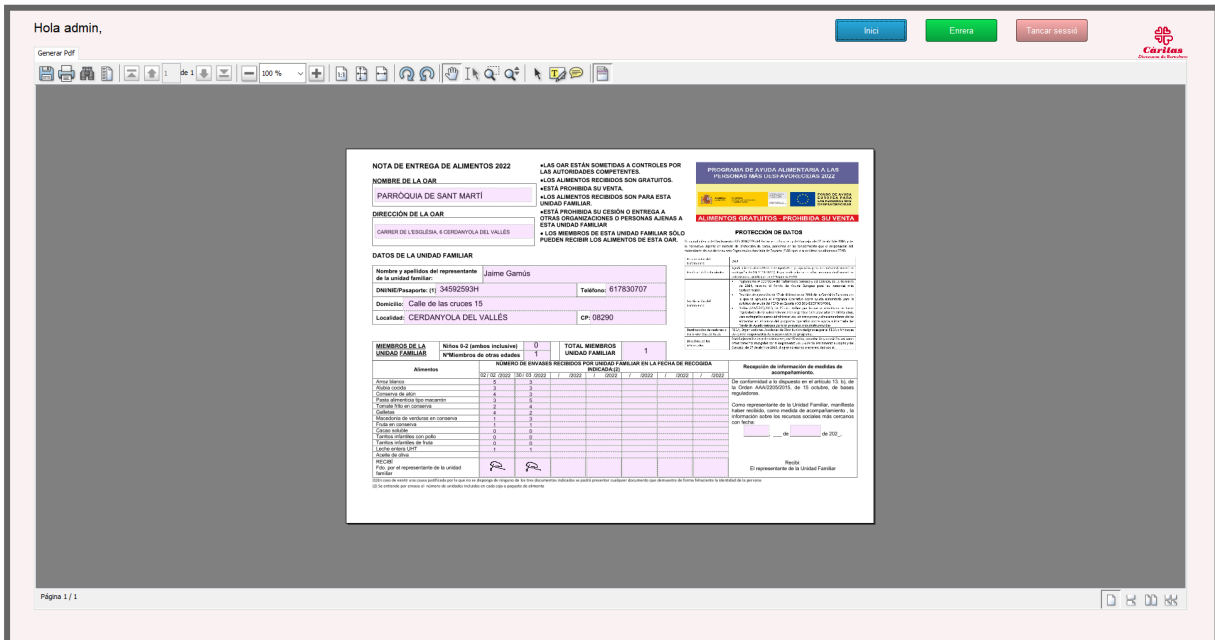


Figure 27: View pdf screen

### 10.4.7. Delivery screen

Figure 28 shows an example of the delivery screen. This screen can be accessed in different ways.

1. The first way to access the delivery screen is by pressing the “make a delivery” button on the menu screen (Figure 19 and Figure 20), as seen in the section 10.4.3.
2. The second way to access the delivery screen is by accessing the consult information screen about a food recipient (Figure 23), through the "make a delivery" button, as described in section 10.4.6.
3. The third option is through the list of food recipients (Figure 22), in case you are using a volunteer account without administrator privileges. If this is the case, when double-clicking, as has been mentioned before, we will access the delivery screen.
4. The fourth and final way is through the food receiver tracking screen, in order to consult a past delivery. Changes can only be made in this screen if we have administrator privileges, otherwise we can only consult it.

This screen has different sections. At the top we have the basic information about the person to whom we are going to deliver: This information consists on the ID, name + surname, delivery date and type of family.

In the center we have the amount of food that should be distributed. This retrieves the delivered day feeds, which are generated/modified, from the distributed foods screen.

At the bottom we have two information boxes, one with the comments that have been added to that delivery and another with the signature, once the product has been delivered.

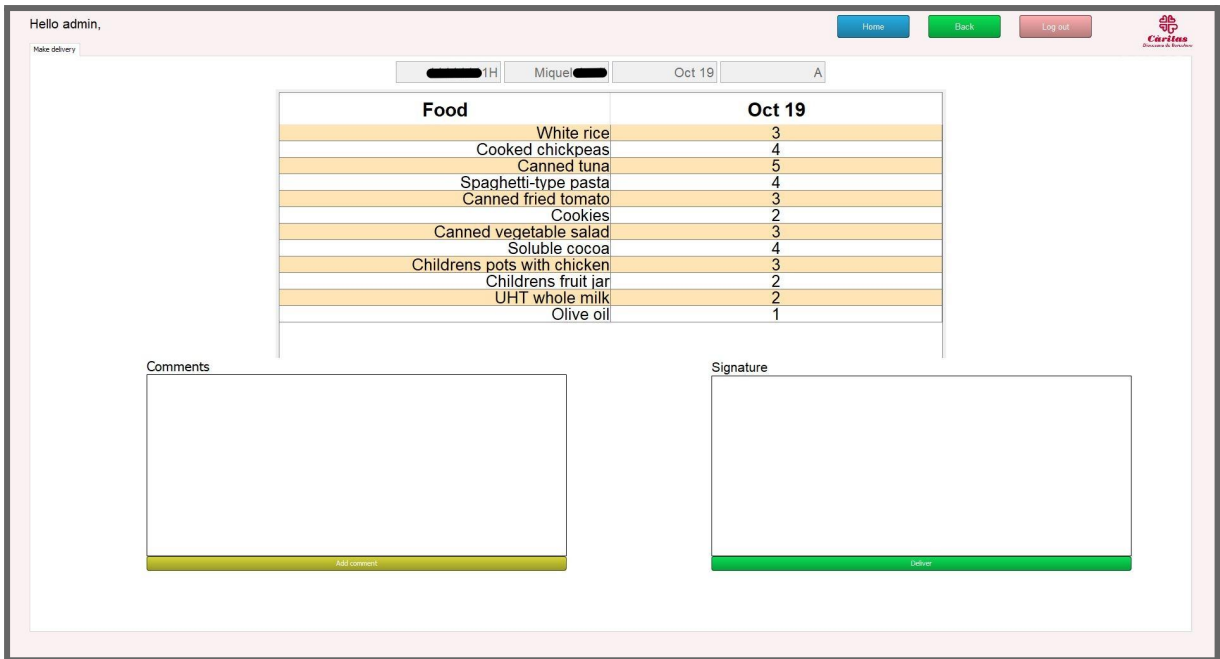


Figure 28: Delivery screen

#### 10.4.7.1. Signature

To formalize the delivery, the recipient of food must sign, to record that she/he has come for it and, in turn, to be able to use that signature for the delivery sheet that must be sent to the European Aid to the Most Deprived.

A modal like the one in Figure 29 appears, where the recipient of food must sign, this can be done with the mouse or using an electronic tablet.

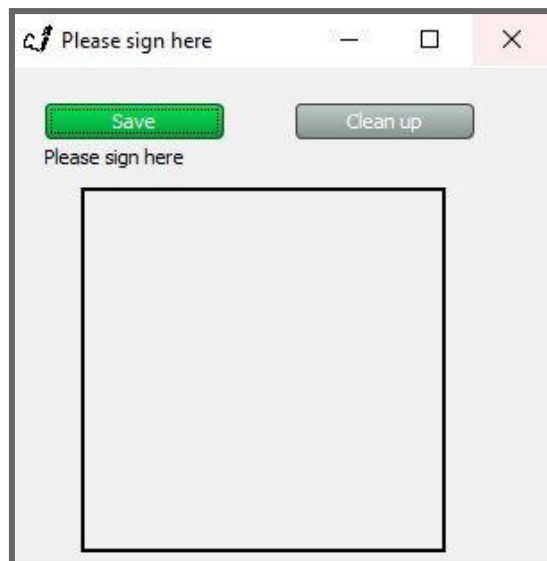


Figure 29: Delivery screen -  
Signature modal

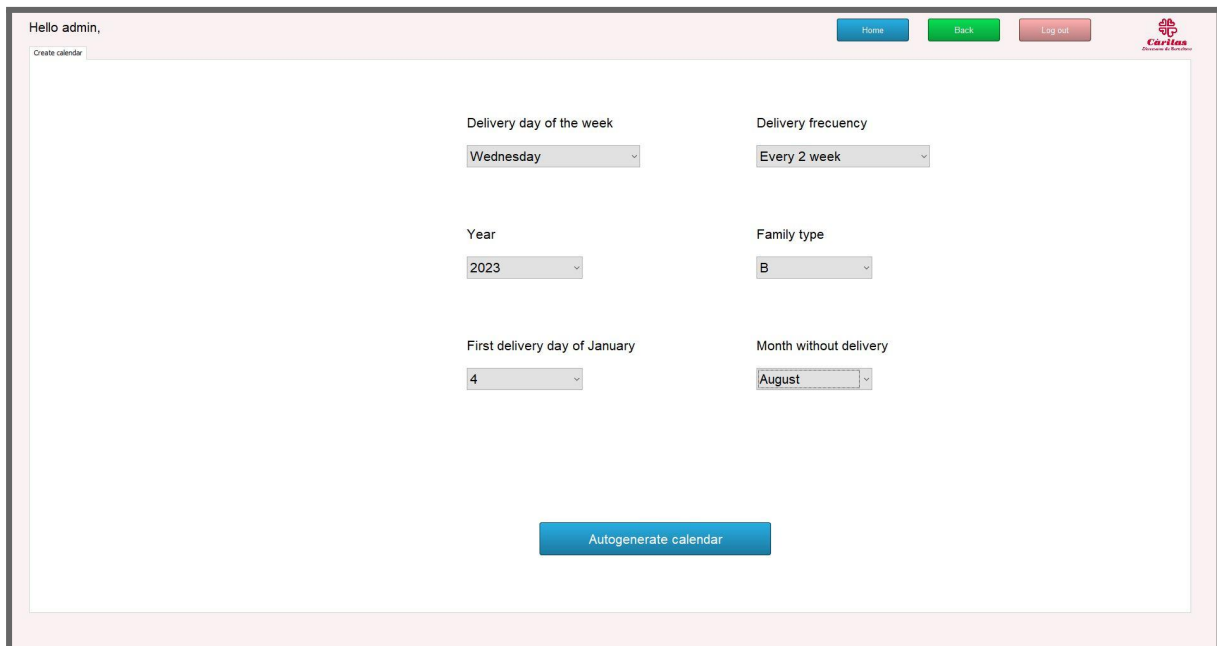
## 10.4.8. Create calendar screen

On this screen we can generate delivery schedules.

Figure 30 and 31 shows an example of creating a calendar screen. This screen has two parts that we explain below.

### 10.4.8.1. Select parameters for calendar screen

The first screen serves to mark the initial parameters. First we select the delivery day and frequency between deliveries. The day can be any weekday and the frequency between every 1 to every 4 weeks. Then we select the year we want to create the calendar for (if there is already a calendar for that year, an error message will appear Table 58 - Message error CC-1) and the type of family (given that the delivery days are different). The first day of distribution must be indicated in order to mark the first week from which the succession will begin. If no delivery is made in any month, it can be indicated in the last field. These months will not be considered when generating the calendar.



The screenshot shows a web interface for creating a calendar. At the top left, it says "Hello admin," and at the top right, there are buttons for "Home", "Back", and "Log out", along with a logo for "CarViss". The main content area is titled "Create calendar" and contains several dropdown menus:

- Delivery day of the week: Wednesday
- Delivery frequency: Every 2 week
- Year: 2023
- Family type: B
- First delivery day of January: 4
- Month without delivery: August

At the bottom center, there is a blue button labeled "Autogenerate calendar".

Figure 30: Create calendar screen - First screen

### 10.4.8.2. Select days for calendar screen

In the second screen we can see the automatically generated values, following the rule that it is a Wednesday every two weeks from the start day of January. Entered days can be deleted or even new ones added. They must meet certain criteria,

otherwise error messages will appear that will guide the user to enter the dates correctly.

- The day of that month does not exist. “January 0” or “February 30” are examples of this error. The system throws an exception and shows an error message informing about this error (Table 58 - Message error CC-2).
- The month is not written correctly. “Agust 4” is an example of an error, since it’s “August 4”. The system throws an exception and shows an error message informing about this error (Table 58 - Message error CC-3).
- The format is not correct. Format must be Month + Day of month. “January” is an example of bad formatting. “30” is also a bad formatting. The value must be “January 30”. The system throws an exception and shows an error message informing about this error (Table 58 - Message error CC-4).

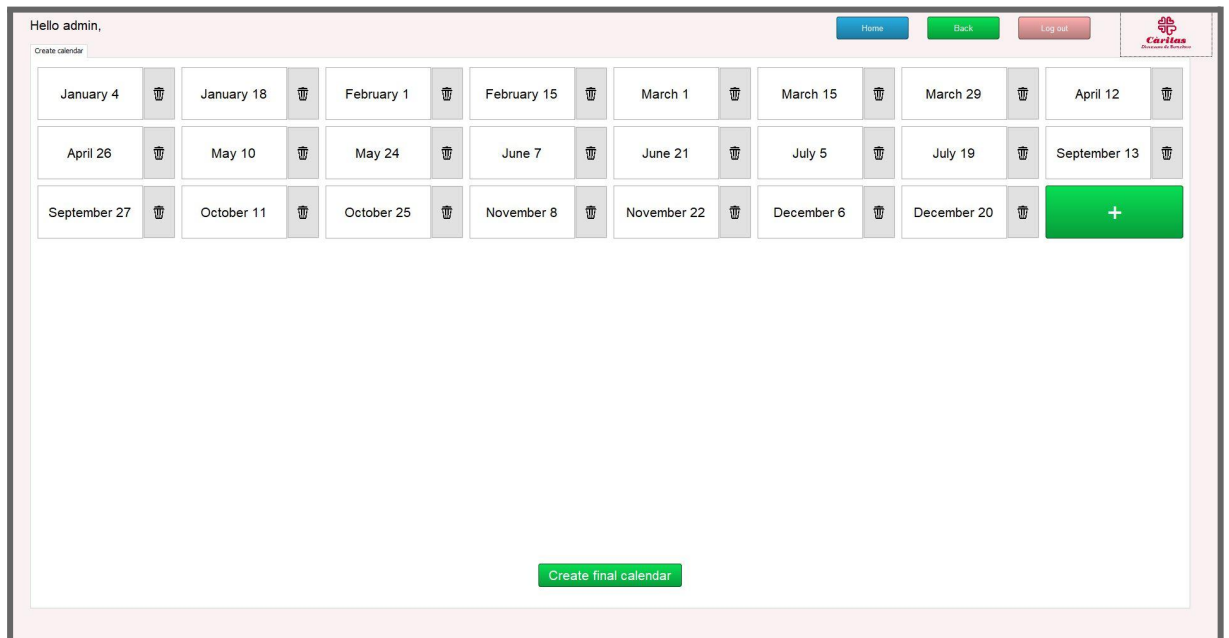


Figure 31: Create calendar screen - Second screen

Error ID	Error description
CC-1	A calendar exists for that year/family type
CC-2	There is a combination Month Day that does not exist.
CC-3	There is a field where the name of the month is not correct. Remember that you have to write the full name.
CC-4	Construction has to be Month Day

Table 58: Create calendar (CC) list of errors

## 10.4.9. Distributed food screen

Figure 32 shows an example of the distributed food screen. This screen is used to manage the amount of food that is going to be distributed on each distribution day. This screen has a dropdown that allows us to choose the year and type of family we want to view.

It also has three buttons:

- **Save:** This button allows you to save the changes made to the table.
- **Add rows:** this button allows you to add new food to distribute.
- **Delete rows:** this button allows you to delete food. A modal appears like the one in Figure 33 that shows you the possible values to delete.

Food distributed

Year 2022 A

Food	Jan 5	Jan 19	Febr 2	Febr 16	Mar 2	Mar 16	Mar 30	Apr 13	Apr 27	May 11	May 25	Jun 8	Jun 22	Jul 6	Jul 20	Sept 7	Sept 21	Oct 5	Oct 19	Nov 2	Nov 16	Nov 30
White rice	4	3	5	3	6	6	3	2	5	6	3	4	5	10	3	4	5	2	3			
Cooked chickpeas	5	4	3	3	3	2	3	2	3	3	2	3	4	5	2	5	2	3	4			
Canned tuna	2	5	4	2	1	2	3	4	5	6	2	3	3	2	2	7	3	2	5			
Spaghetti-type pasta	3	6	3	3	5	6	5	4	3	4	6	5	2	3	3	6	2	1	4			
Canned fried tomato	4	3	2	1	2	3	4	5	1	1	2	2	1	4	3	2	1	3	3			
Cookies	5	2	4	3	5	3	2	1	4	3	3	3	1	5	2	4	1	4	2			
Canned vegetable salad	2	1	1	2	1	2	3	2	1	3	2	3	1	6	2	3	1	5	3			
Soluble cocoa	1	1	1	2	1	2	1	1	1	2	2	2	1	2	5	5	2	2	4			
Childrens pots with chicken	0	0	0	0	0	0	0	0	0	0	0	0	1				3	1				
Childrens fruit jar	0	0	0	0	0	0	0	0	0	0	0	0	1				4	2				
LHT whole milk	1	0	0	0	0	2	0	0	0	2	0	1	1	2	3	1	5	4	2			
Olive oil	2	4	1	1	1	1	1	1	1	1	1	1	1	2	3	2	6	5	1			

Save Add row Delete row(s)

Figure 32: Distributed food screen



Figure 33: Distributed food screen -  
Delete rows modal

### 10.4.10. Food recipient tracking screen

Figure 34 shows an example of the food recipient tracking screen. On this screen we can track the food receptors. It has a search engine similar to the food receiver search screen. It also has a dropdown to select the year and type of family we want to see the follow-up on. We can obtain the details of the deliveries by double clicking on the “X” in the table.

Hello admin, Home Back Log out

Food Recipient Tracking

Year: 2022 A. Search

DNINIE/Pasport	Name	Jan 5	Jan 19	Febr 2	Febr 16	Mar 2	Mar 16	Mar 30	Apr 13	Apr 27	May 11	May 25	Jun 8	Jun 22	Jul 6	Jul 20	Sept 7	Sept 21	Oct 5	Oct 19	Nov 2	Nov 16	
██████████	Miquel	X			X	X					X												
██████████	Laura	X		X			X		X			X				X							
██████████	Jaime	X	X			X	X							X					X				
██████████	Xan	X		X										X									
██████████	David	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
██████████	Miguel	X	X							X				X					X				
██████████	Xavi	X				X				X				X									
██████████	Carles			X	X		X					X											

Figure 34: Food recipient tracking screen

## 10.4.11. Account Management

### 10.4.11.1. Volunteer accounts management screen

On this screen we can manage the accounts of the different volunteers. Figure 35 shows an example of the volunteer accounts management screen.

We have different fields:

- **User:** In this field, we write the name of the user to register, modify or delete.
- **Password:** In this field, we write the password of the user that we want to create, and the password that we want to put on the user to modify.
- **Email:** In this field, the volunteer's email is entered, necessary in case you need to recover the password and there is no volunteer with administrator privileges nearby.
- **Administrator checkbox:** With this checkbox we mark the privileges of the volunteer; if it is marked she/he will be an administrator and if not, she/he will not.

And different buttons:

- **See password:** This field is used to have visibility over the password that we are entering.
- **Consult active volunteer accounts:** With this button we will see a list of all the volunteers registered in the system.
- **Register:** With this button we will register a new user, with the entered parameters.
- **Modify:** With this button we can modify the password, email or privileges of a volunteer.
- **Clear:** With this button we will reset the entered values.
- **Delete:** This button is used to delete the user with a username equal to that of the user field.



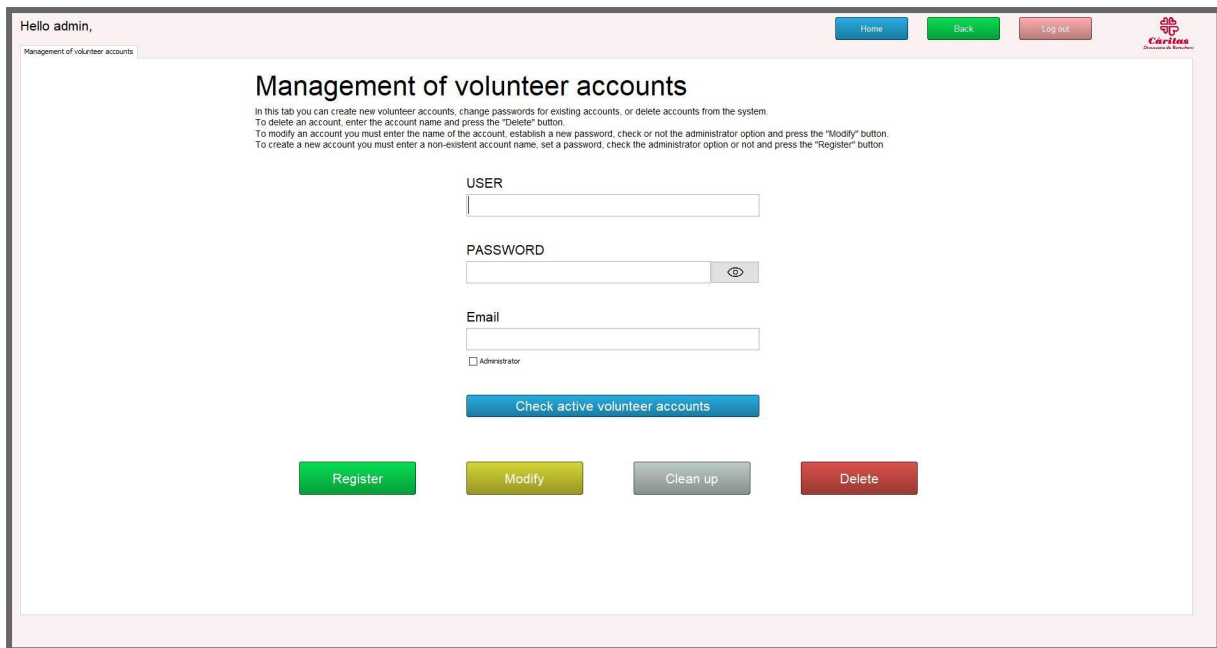


Figure 35: Volunteer accounts management screen

#### 10.4.11.2. Member account management screen

On this screen we can manage our account, check our data or modify it for a new one. Figure 36 shows an example of the member account management screen. We have different fields:

- **User:** In this field we see the username of the volunteer account we are logged in. This value can not be modified as a member.
- **Password:** In this field we can check our password or write a new one to modify it.
- **Email:** In this field we check our email or write a new one to modify it.
- **Administrator checkbox:** With this checkbox we can check our privileges. We can't modify this parameter as a member.

And different buttons:

- **See password:** This field is used to have visibility over the password that we are entering.
- **Modify:** With this button we can modify the password and email of our account.
- **Clear:** With this button we will reset the password and email values.

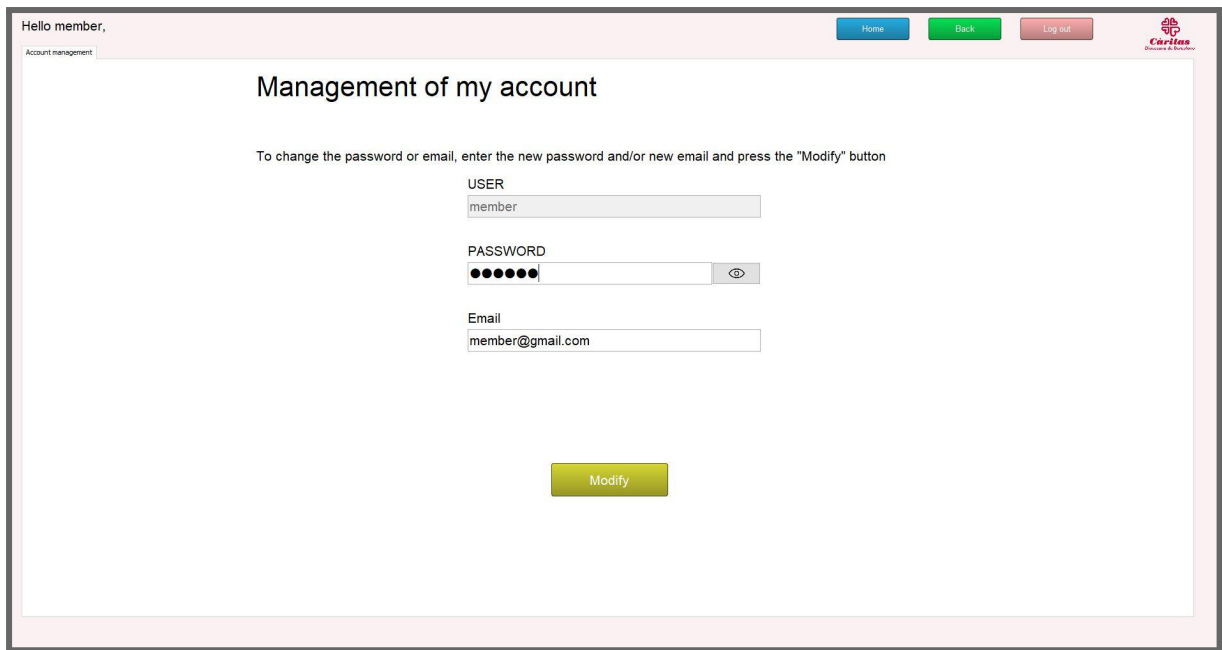


Figure 36: Volunteer my space management screen

## 10.4.12. Backup management screen

On this screen we can manage everything related to backups. Figure 37 shows an example of the backup management screen. This screen has two buttons:

- Create backup: with this button we are going to save a backup with the name backup\_day\_month\_year.
- Restore backup: with this button we can load an old backup, in case some fatal error has occurred. It takes the backup file and turns it back into a database. Restoring the old values and deleting the new ones. If a corrupted file or non sql file is selected, the system throws an exception and shows an error message informing about this error (Table 59 - Message error BM-1). If the restore fail from any other reason, the system throws a different exception informing you the backup was not successful (Table 59 - Message error BM-2).

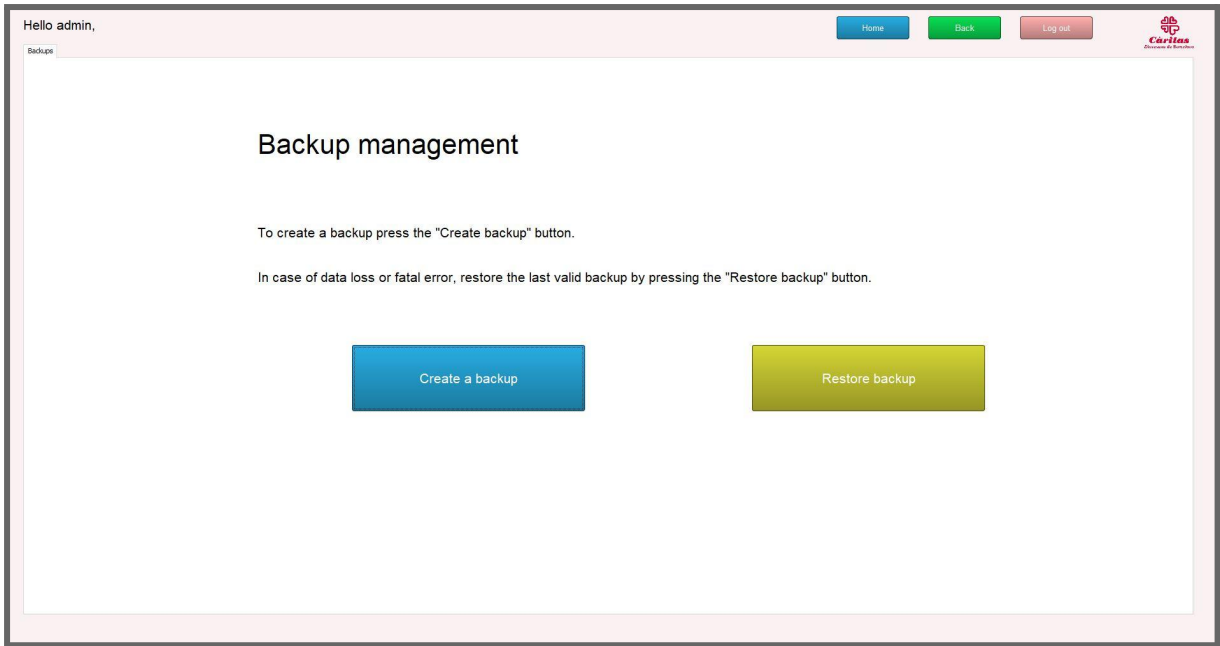


Figure 37: Backup management screen

Error ID	Error description
BM-1	The selected file is not an “.sql” file or it is corrupted.
BM-2	There was a problem performing the backup.

Table 59: Backup management (BM) list of errors

## 11. Results

This section includes three items: Cáritas people's assessments of this system, a satisfaction survey that will be given to volunteers once the application is up and running, and finally a user manual, in less technical language to facilitate their understanding.

### 11.1. Assessments

In this section we talk about the vision of Marina Pous, Cáritas reference technician in Cerdanyola on the application.

First of all, she emphasizes that this application will facilitate work and will largely reduce the hours spent on bureaucratic matters. She claims that the application is very intuitive and believes that all volunteers will be able to use it without problems.

According to her own words, "if they use WhatsApp, they will know how to run this application." At the same time, during the interview, she has taken the opportunity to order a second version of the system with more features, which are described in section 12.4, "Future work".

## 11.2. User manual

In Annex 1 there is a user manual; it contains an installation guide and the description of all the functionalities offered by the application. All of this is made in a non-technical language, to facilitate its understanding.

## 11.3. Use and surveys

Attached in Annex 2 is a satisfaction form, aimed at volunteers, once they have tried the application. This questionnaire has only been passed on to the few users who have been able to see the application, including Father Juan Carlos, Sister Eva and Marina Pous.

# 12. Conclusions

Ever since Father Juan Carlos asked me for help in this project, I was very confident that despite being a challenge I would be successful. It is a project that still needs work and dedication, but it already has the foundations to be able to move forward. In spite of the different difficulties I have encountered, the thought that this project could be of great help to the people of Cáritas has always kept me going and trying to get there, I have always done better and better.

## 12.1. Difficulties encountered

Summarizing, and under my point of view, this project has been a big challenge: knowing the client's needs, holding meetings, seeing how a main idea matured and changed greatly compared to the previous one, making many changes to everything proposed and facing new features.

First of all, the first obstacle I have encountered in this project has been the lack of time, not having been able to combine working life, student life and social life quite

well. The alternative plan to overcome this obstacle has been to extend the delivery period in order to be able to deliver finished work useful to the people of Cáritas.

At some points I have been a bit overwhelmed because of my lack of knowledge in the possibilities offered by Java Swing. The proposed solution to overcome this obstacle has been to invest time in learning the functionalities that it offers.

Another thing I have seen is that the project evolved as it took shape, causing new functional and non-functional requirements to emerge as it went along, because there were things I had not considered necessary or possibilities I did not know about, which meant that I had to redo code, have meetings to discuss these new requirements, research on the subject... In short, a greater investment of time.

## 12.2. Relationship of the project with the degree

To begin with, a WATR (Warm up, Asis, To be, Roadmap) analysis was applied. This method was mainly used in the Information System subjects. This consists of a warm up phase where we see what the possibilities are that the project offers. We proceed to do an ASIS analysis of the current situation in which the project is, and what is there now. We decide on the To be, where we want to reach with the project. And finally we define the Roadmap in order to reach our goal.

I have used SQL, learned in the BD subject. In particular, I have used the most common statements (SELECT, INSERT, UPDATE and DELETE). Having done this subject I have been able to avoid a learning curve. I had to create relational tables from 0, starting with the design and specification of them following the learning of the IES subject. I had to make use of an object-oriented system, as we learned in the Programming subjects.

## 12.3. Achievement of specialist technical skills

Below we comment on the level of achievement of the technical skills that were set as an objective in the definition of the project.

- **CSI2.1: To demonstrate comprehension and apply the management principles and techniques about quality and technological innovation in the organizations.** To a large extent this technical skill has been achieved, I

had to make a plan about the work we would do and define the application quality requirements.

- **CSI2.3: To demonstrate knowledge and application capacity of extraction and knowledge management systems.** All the information extracted from this application can be used to improve the knowledge sources of the Cerdanyola town hall and therefore can help in its Knowledge Management Systems.
- **CSI2.5: To demonstrate knowledge and capacity to apply business information systems (ERP, CRM, SCM, etc.).** The project consists of a CRM and therefore all the knowledge related to this information system had to be used.
- **CSI3.2: To develop the information system plan of an organization.** The entire information system has been designed from scratch, based on the need mentioned by Cáritas.
- **CSI4.3: To administrate databases (CES1.6).** For the achievement of this project several tables had to be created and all the relevant calls had to be made for the creation, modification and deletion of data within these or the tables themselves. At the same time, relationships have had to be created between them.
- **CSI4.2: To participate actively in the design, implementation and maintenance of the information and communication systems.** This is the most achieved objective, given that the project is based on exactly that, the design, implementation and maintenance of an information system for Cáritas.

## 12.4. Future work

In this section we will discuss the work to be done in the future. These sections include points that arose during the project, but that differed from the initial scope. This is a job that will be done after the delivery of this project, to end up giving Cáritas the software best suited to its needs.

### 12.4.1. Maintenance and support

- **Maintenance:** Like any application that works with databases and information that may undergo new changes or new needs, it is necessary to give it a small

maintenance service in order to ensure its proper operation over time. At the same time, during the first months of the application's life, everything necessary will be supported so that Cáritas volunteers can make proper use of the application.

- **Mobile app:** The creation of a mobile application, capable of carrying out all the tasks mentioned in the project, in order to be able to give more freedom for Cáritas volunteers, who do not need to always have a computer in front of them. We have already started to do a design space exploration of how to do that. Thanks to the technology used for the application, it seems that porting a baseline front-end to mobile systems should not be very difficult.
- **Stock Control:** More control of the stock, being able not only to keep track of what is to be distributed but also of all the food they receive, to be able to have greater control. This could have quantities, information about the location of the warehouse where it is and so on.
- **Additional information related to users:** This information includes the name of all members of your family unit, with date of birth and gender. This information will be used to make statistics.
- **A distribution control via coupons:** each registered user can be active or not. In order to be active, your social worker must validate that you need this help. At the time he gives you the help, you are assigned a number of distributions that you can access. This will also require that, at the same time as registering a user, a document can be generated with their data and the distribution days of which the coupon consists.

#### 12.4.2. Scalability of project

If this tool is useful for Cáritas Cerdanyola, it could be scaled up and used in all Cáritas centers in Spain. With all that this entailed, it has been tried at all times to make the code as generic as possible, already looking for this future scalability.

## 12.5. Acknowledgements

Finally I would like to give special thanks to Juan Carlos Montserrat, for giving me his vote of confidence and entrusting me with this project. To Eva, who has been the direct contact of Cáritas, always ready to help in any need. To my coworkers, for helping me in the testing phase. And finally to my director Daniel Jiménez, for always being willing to help by giving me the ideas I had not come up with in order to make this application the most professional possible.



# Bibliography

[AltexSoft website - Functional and non functional requirements.](#)

AltexSoft. «Functional and Nonfunctional Requirements: Specification and Types».

Accedido 17 de septiembre de 2022.

[Ben Lutkevich - Techtargget website - Definition of access control list.](#)

SearchNetworking. «What Is Access Control List (ACL)? - SearchSoftwareQuality».

Accedido 1 de octubre de 2022.

[Blog sqlbak - How to backup mysql database .](#)

«How to Backup MySQL Database on Windows – SqlBak Blog». Accedido 8 de junio de 2022.

[Brandan Jones - youtube - Show a drop down list in Java JOptionPane prompt.](#)

*Show a drop down list in Java JOptionPane prompt*, 2016.

[Cáritas website.](#)

Cáritas. «Conoce nuestra misión y visión». Accedido el 26 de febrero de 2022.

[Chuidiang website - GridBagLayout.](#)

«Ejemplo de uso del GridBagLayout». Accedido 3 de abril de 2022.

[Cloudflare website - What is SSL.](#)

Cloudflare. «¿Qué es SSL (Secure Sockets Layer)?»

Accedido 1 de octubre de 2022.

[Digite website - What is Scrum Methodology](#)

«What Is Scrum Methodology? & Scrum Project Management», 30 de septiembre de 2019.

[EEC website.](#)

European Commission - European Commission. «European Commission, Official Website». Text. Accedido 23 de febrero de 2022.

[FAO - El derecho humano a la alimentación pdf](#)

«El derecho humano a la alimentación», s. f., 2.

[FESBAL website](#)

FESBAL. «Bancos de Alimentos de España | Federación Española de Bancos de Alimentos». Accedido el 27 de febrero de 2022.

[GeeksForGeeks website - How to concat two columns into one.](#)

GeeksforGeeks. «How to Concat Two Columns Into One With the Existing Column Name in MySQL?», 16 de noviembre de 2021.

[Genuinecoder website - Convert Java jar to exe.](#)

Coder, Genuine. «JAR to EXE : Java Program to Windows Executable». *Genuine Coder* (blog), 9 de diciembre de 2017.

[Hartman, James - What Is Java?.](#)

Hartman, James. «What Is Java? Definition, Meaning & Features of Java Platforms», 6 de enero de 2020.

[Hurtado, Javier Sáez Agile Methodologies](#)

Hurtado, Javier Sáez. «Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa». *Thinking for Innovation*, 9 de diciembre de 2021.

[Icoconvert website - Create Icons from PNG & JPG.](#)

«ICO Convert - Create Icons From PNG & JPG Images Online». Accedido 10 de abril de 2022.

[Javatpoint website - Java JTable.](#)

www.javatpoint.com. «Java JTable - Javatpoint». Accedido 3 de abril de 2022.

[Microsoft - Create a full database backup sql server.](#)

MashaMSFT. «Create a Full Database Backup - SQL Server». Accedido 8 de junio de 2022.

[MySQL website - MySQL 8.0 Download.](#)

«MySQL :: Download MySQL Community Server». Accedido el 9 de junio de 2022.

[MySQL website - MySQL 8.0 Reference Manual.](#)

«MySQL :: MySQL 8.0 Reference Manual :: 6.1.1 Security Guidelines»

[Nirmal- thInk beYond - Stackoverflow - How to connect to a remote MySQL database using Java.](#)

iShader. «How to connect to a remote MySQL database using Java?» Forum post. *Stack Overflow*, 30 de agosto de 2011.

[O'Reilly website - Volere Template](#)

«B. Volere Requirements Specification Template - Mastering the Requirements Process, Second Edition [Book]». Accedido 22 de junio de 2022.

[Oracle website - Differences between Java EE and Java SE.](#)

«Differences between Java EE and Java SE - Your First Cup: An Introduction to the Java EE Platform». Accedido 28 de junio de 2022.

[Oracle website - Java JDK 16 Download.](#)

«Java Archive Downloads - Java SE 16». Accedido el 9 de junio de 2022.

[OuterVision website - Power supply calculator.](#)

«Power Supply Calculator - PSU Calculator | OuterVision». Accedido 14 de marzo de 2022.

[Renovar-DNI.net - Calcular la letra del dni.](#)

«Cómo calcular la letra del DNI, comprobar si un DNI es correcto». Accedido 27 de agosto de 2022.

[ResponsabilidadSocial website - ¿Qué es la sostenibilidad?](#)

RSyS. «Sostenibilidad: qué es, definición, concepto, tipos y ejemplos». *Responsabilidad Social y Sustentabilidad* (blog), 8 de enero de 2022.

[Rishabh Mishra - How to send email in Java using Gmail SMTP.](#)

«How To Send Email In Java Using Gmail SMTP? | Pepipost». Accedido 16 de abril de 2022.

[Siordia, Omar Rodriguez - ¿Cuanto cuesta hacer una app?.](#)

Pixan Talks. «¿Cuánto Cuesta Hacer Una App?», 26 de febrero de 2020.

[TechBrothersIT - youtube - How to connect to a MySQL Server Remotely with MySQL Workbench, 2018.](#)

*MySQL Workbench Tutorial - How to Connect to a MySQL Server Remotely with MySQL Workbench*, 2018.

[Universidad de Alicante - Modelo Vista Controlador](#)

Alicante, Servicio de Informática Universidad de. «Modelo vista controlador (MVC)». Accedido 27 de septiembre de 2022.

[Viviek Gite - PostgreSQL: allow remote access tcp connection.](#)

October 9, Author: Vivek Gite. «How Do I Enable Remote Access to PostgreSQL Database Server?» nixCraft, 9 de octubre de 2007.

[Wikipedia European Economic Community](#)

«European Economic Community». En *Wikipedia*, 23 de febrero de 2022.

[Zukowski. John - Swing Threading and the Event-Dispatch Thread.](#)

Zukowski, John. «Swing Threading and the Event-Dispatch Thread». InfoWorld, 21 de agosto de 2007.