Student Project: The RecipEasy case study

Carlo Fanciulli

Executive Summary:

This document describes the requirements of the Recipeasy case study proposed as student project in the RE course, aa. 2019-2020.
It specifically presents the proposed application to facilitate the process of finding recipes.
This case study aims to evaluate and define the issues of the current situation and their possible solutions with an overview of goal-oriented requirements for both the as-is and to-be cases. In addition to this, a clear schedule of activities and requirements needed to develop the program is provided.
Recipeasy is a service available to everyone that allows users to obtain recipes by selecting ingredients they already have on hand and that they want to cook.
Moreover, the app offers an innovative approach to the recipe finding process by suggesting to the user recipes filtered based on their personal needs and by allowing them to buy missing ingredients through the app.
In order to understand the elements to be added or modified to improve the system, a survey has been conducted and taken into consideration for the aim of developing the to-be model.
The document also contains an OO requirements analysis with the use case, sequence model and class model, traceability matrices that show the relationships between requirements, prioritization to ensures that the project focuses on the most important elements first and eventually the user feedback analysis.

Revision History

Date	Version	Description	Author
17-Oct-19	0.1	Analysis-brainstorming	Carlo F.
28-Oct-19	0.2	Development	Carlo F.
03-Nov-19	0.3	Overall review	Carlo F.
02-Dic-19	0.4	Second version	Carlo F.
10-Jan-20	1.0	Final version	Carlo F.

Table of Contents

1. Glossary

Acronym	Description
Recipe DB	Recipe Database (Internet)
RE	RecipeEasy
RU	Registered User

RecipEasy Software System Requirements

2. The application domain

Nowadays, the world can be considered as a global village, where everything is accessible through technology. The advent of mobile phone has shaped the life of many people, which means that mobile applications have already made their ways into our lives.
Everyone in their daily routine must cook, whether it is out of passion or necessity.
RE offers an easy method to optimize the process of finding suitable recipes, making cooking with a smartphone a lot quicker and tastier thanks to a personalized combination of recipes.

The proposed solution is to develop a “RecipEasy” mobile application, with a clean and simple interface, for people who love to cook and try out new recipes but also for who does not have a passion for cooking but needs to realize tasty and easy recipes with a short time available. Therefore, the main stakeholders of this application would be both young people, like students, and middle-aged individuals who need to cook every day for their family in a short time.

The app is a time saver, providing recipes in few clicks, starting from selecting the ingredients that the user has in their pantry, with the possibility to filter them according to personal preferences (gluten-free, vegan, preparation time). Thanks to an artificial intelligence system, RE takes recipes from millions of different websites choosing them based on the preferences selected by the user and allowing the RU to read and review recipes.

The system also allows users to visualize list of recipes that contain missing ingredients, but offers only to RU the possibility to buy them directly in the app through an integrated purchasing platform.
The RU can also view added favorite recipe list and use a voice recognition system to select the ingredients needed to search for the recipe.
The artificial intelligence algorithm and the payment process for purchasing ingredients will not be better considered and developed as requirements in this document.

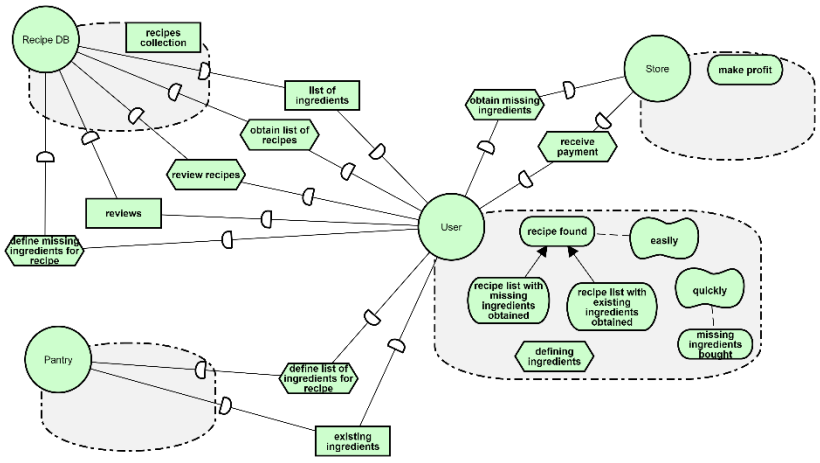
3. Goal-oriented requirements analysis: As-Is situation

3.1 Strategic dependency view

Table 1: Main Dependencies

Depender	Dependee	Dependum	Description
User	Recipe DB	Obtain list of recipes	Task in which the user is provided by the recipe DB with a list of recipes based on the ingredients he selected. The recipe may contain ingredients the user does not own.
User	Recipe DB	Define missing ingredients for recipe	Task in which the user extrapolates from the recipe selected the ingredients he doesn't have on hand.
User	Recipe DB	Reviews	Resource available to the user through the recipe DB
User	Recipe DB	Review recipes	Task performed by the user to rate the recipe made.
Recipe DB	User	list of ingredients	Resource needed by recipe DB to provide recipes according to user needs
User	Store	Obtain missing ingredients	task in which the user relies on the store to obtain the missing ingredients for the preparation of the recipe
Store	User	Receive payment	task in which the store receives money from the buyer in exchange for the ingredients
User	Pantry	Define list of ingredients for recipe	Task in which the user relies on the pantry to know the ingredients available for the realization of the recipe
Pantry	User	Existing ingredients	Resource needed by the pantry to perform its function of collecting ingredients

3.2 Strategic rationale view



4. Eliciting requirements

In order to better understand the issues faced in the process of finding a recipe and to solve them by creating an app suitable to the needs and habits of the users, a survey was created and a population of 18 individuals was asked to answer the questions.

The first section of the questionnaire concerns the interviewee's basic information, made with the aim of analyzing the nature of the population regarding age, profession, family status, familiarity with the use of a smartphone and general daily routine. The majority of the individuals (the 55 %) resulted to be located in a range of age between 16 and 25 years, and the remaining portion of the population resulted to be respectively 28% between 45 and 60 years old and 17% between 25 and 34 years old.
All of the population interviewed declared to be familiar with the use of electronic devices and of online-shopping websites. This last aspect highlights the concrete possibility for the application to successfully insert a platform that allows the user to easily buy ingredients online.
Moreover, the 90% of the individuals stated to be at home not so often (average score of 3 out of 5) and the average distance from a supermarket is of a range between 500 meters and 1,5 kilometers.

Moving on to the next section of questions, the people were asked about their habits and relationship with cooking and finding recipes. To the question of evaluating how easy it is to find a suitable recipe online, the 85% of the subjects answered with an evaluation of 3 out of 5; however, when asked to state how often the recipes chosen contain only ingredients available at home, all of the interviewees declared it to be a rare event (score of 2 out of 5). Moreover, the 70% of the individuals stated that they often use a recipe when cooking (score of 4 out of 5) and the remaining 30% gave an also positive answer (score of 3 out of 5). However, the 90% of the total asserts that if they do not have all of the ingredients available at home, they would rather changing recipe than buying the missing components (even though the supermarket is not far).
This data made it possible to understand that finding a recipe that contains only ingredients present at home is a consistent issue for the average population. The app would therefore have the opportunity to become part of the market by meeting a need felt by the majority of people.

On the last part of the survey the population was asked questions aimed at getting knowledge about the participants' interest into some possible features of the application. More specifically, the questionnaire unexpectedly showed an interest (mostly 3 and 4 scores) in a voice recognition tool, which gave us the idea of adding a feature that allows the user to orally select ingredients in order to find the recipe.
In addition to this, another important result of the survey is that almost everyone claims to want a better and more personalized result when looking for a recipe. This would be a focal point for the application, since the implementation of filters for the recipe research process would drastically improve the output displayed to the user, making their experience better customized and therefore more precise.
The last important result shows that only 2 people out of the 18 interviewed has an effective way to store their recipes in order to easily find them when needed. This consideration highlights another important need of the user that can be well performed by the system through the realization of a specific interface within the app that allows users to add and remove favorite recipes.

In conclusion, the questionnaire shows that there is a considerable margin of improvement of the current situation that could be fulfilled by the RecipEasy application with the to-be model.

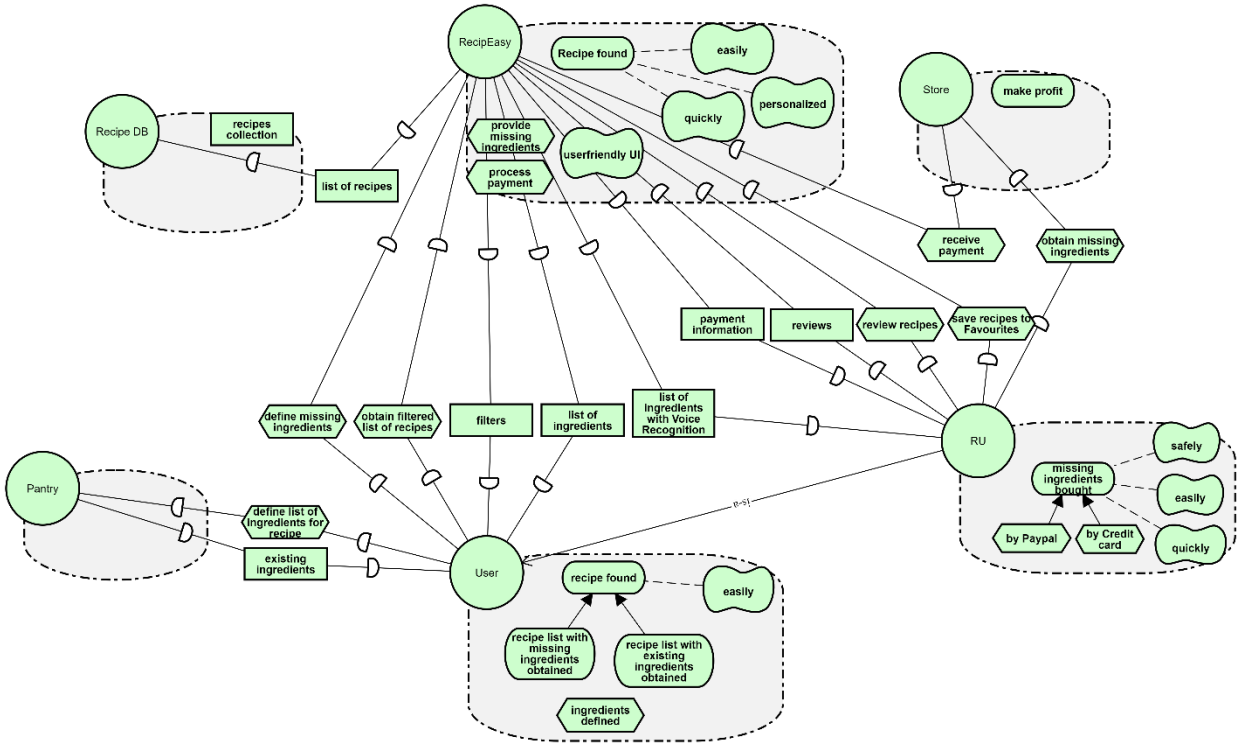
5. Goal-oriented requirements analysis: to-be situation

5.1 Strategic dependency view

Table 1: Main Dependencies

Depender	Dependee	Dependum	Description
RecipEasy	RecipeDB	List of recipes	Resource from Recipe DB needed by RE to provide recipes according to user needs
RecipeEasy	User	filters	Resource needed by RE to provide personalized results according to user preferences
RecipeEasy	User	List of ingredients	Resource needed by RE to provide a restricted list of recipes
RecipeEasy	RU	List of ing. with voice recognition	Resource needed by RE to provide recipes through voice command
RecipEasy	RU	Payment information	Resource needed by RE to perform the payment to the store
RU	RecipEasy	Review recipes	Task that the RU performs through the RE to evaluate recipes
RU	RecipEasy	Reviews	Resource that the user can access through RE
RU	RecipEasy	Save recipes to favourites	Task that the RU performs through the RE to save recipes he likes
RU	Store	Obtain missing ingredients	Task that the RU performs through the Store receiving ingredients he bought
Store	RecipEasy	Receive payment	Task in which the Store receives the money of the RU through RE
User	RecipEasy	Obtain filtered list of recipes	Task in which the user obtains a restricted collection of recipes based on filters selected
User	RecipEasy	Define missing ingredients	Task in which RE communicates the user the missing ingredients for the recipe he selected

5.2 Strategic rationale view



6. Requirements

The specifications, written using the "SHALL" patterns, defined for this project are listed below in this section.

FUN1: The application shall allow the user to register to the platform.
FUN2: The application shall allow the user to login to the platform.
FUN3: The application shall allow the user to visualize ingredients in the pantry.
FUN4: The application shall allow the user to select ingredients to find recipes efficiently.
FUN5: The application shall allow the user to filter the recipes obtaining personalized results in an effective way.
FUN6: The application shall allow the user to visualize details of a certain recipe.
FUN7: The application shall allow the registered user to add recipes to favorites.
FUN8: The application shall allow the registered user to access the recipes that he/she marked as favorite.
FUN9: The application shall allow the registered user to visualize missing ingredients in the shopping list.
FUN10: The application shall allow the registered user to buy ingredients that he/she does not have through the application.
FUN11: The application shall allow the registered user to add ingredients to the pantry through voice command easily.
FUN12: The application shall connect to recipe DB to obtain a list of recipes that will be filtered for the user to visualize them quickly.

FUN13: The application shall allow the registered user to review recipes through the app.
FUN14: The application shall allow the registered user to view other registered users' reviews of recipes with an accessible UI.
FUN15: The application shall allow the registered user to purchase missing ingredients paying by Paypal or by credit card

NFR01: The Store shall allow the registered user to receive at home missing ingredients in less than 12 hours.
NFR02: The application shall allow the registered user to add ingredients with voice command in less than 05 seconds.
NFR03: The application shall be easy to use and intuitive
NFR04: The application shall be fast and robust when loading.
NFR05: The application shall be protected from any external danger or attacks.

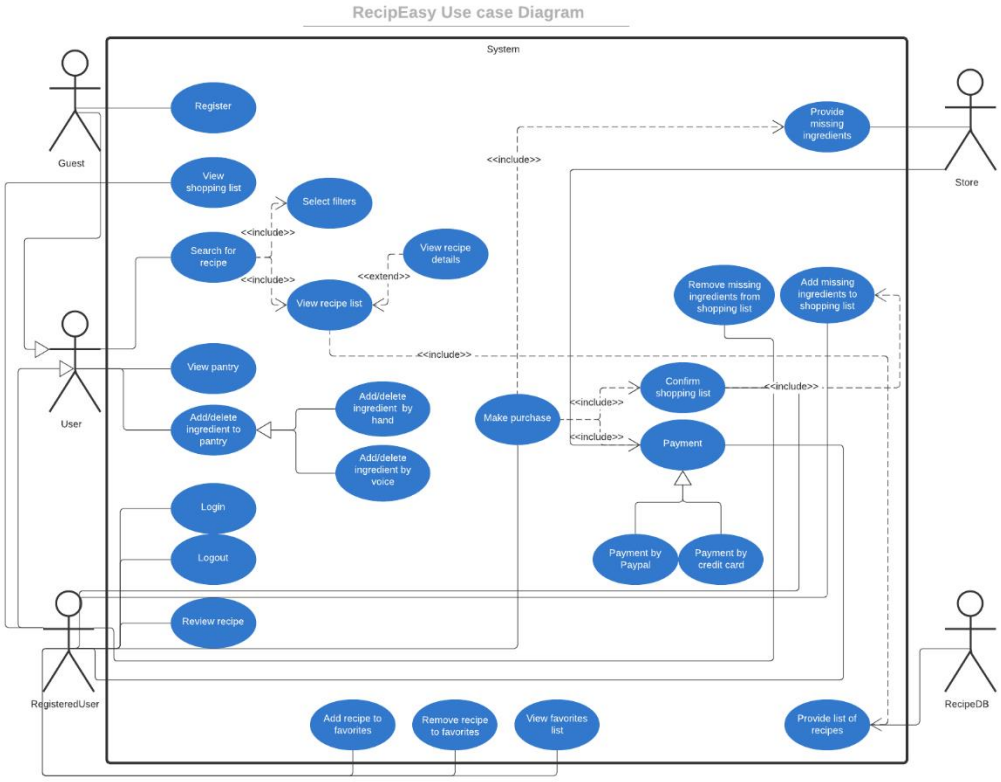
Scenario-oriented Acceptance Criteria:

FUN7:
-Given that I'm in a role of registered user
-And I'm on a recipe page
-When I click the 'add to favorites' button
-Then the system saves recipes in the favorite list page

FUN5:
-Given that I'm in a role of registered user or guest user
-And I'm on the recipe list page
-When I select one or more available filters
-Then the system shows personalized results in an effective way

FUN10:
-Given that I'm in a role of registered user
-And I open a recipe page with a missing ingredient
-When I click 'buy ingredient' button
-Then the system shows the page to proceed with the purchase of the product

7. OO requirements Analysis



Use Case Specification

Use case Name:	**Add ingredient to pantry**
Actor:	User
Description:	UC that allows user to upgrade pantry status with desired ingredients
Preconditions:	None
Postconditions:	The pantry status is now upgraded
Normal Course of Events:	The user navigates to the Pantry page, searches for the ingredient, and adds it to the pantry by pressing the specific button
Alternative Courses:	The user can search for the ingredient by pressing another button with a microphone icon that allows to use voice recognition
Exceptions:	

Use case Name:	**View pantry**

Actor:	User
Description:	UC that allows user to view the current status of ingredients present in the pantry
Preconditions:	None
Postconditions:	User is presented with the pantry status information
Normal Course of Events:	The user navigates to the Pantry page, presses the “My Pantry button allowing to visualize all the selected ingredients.
Alternative Courses:	
Exceptions:	

Use case Name:	**Search for recipe**
Actor:	User
Description:	UC that allows user to search for recipes containing the ingredients selected in the pantry
Preconditions:	Internet connection; more than one ingredient selected.
Postconditions:	User is now provided with a list of recipes containing the selected ingredients
Normal Course of Events:	The user navigates to the Pantry page, presses the “See recipes” button being directed to the “recipe list” page.
Alternative Courses:	The user can view the recipes with the selected ingredients by pressing directly in the "recipe list" page
Exceptions:	

Use case Name:	**Select filters**
Actor:	User
Description:	UC that allows user to filter results based on his/her personal preferences
Preconditions:	Adding at least one ingredient to display recipes
Postconditions:	User is provided with a list of recipes filtered based on his/her preferences
Normal Course of Events:	The user navigates to the “recipe list” page, presses the “Add filter” button, fills some fields and presses the confirm button
Alternative Courses:	
Exceptions:	

Use case Name:	**View recipe details**
Actor:	User
Description:	UC that allows user to have access to instructions and other details for the execution of the selected recipe
Preconditions:	Select a recipe
Postconditions:	User is now provided with instructions and other details for the recipe execution
Normal Course of Events:	The user navigates to the recipe list page, clicks on a recipe and is presented with all the information related to that.

Alternative Courses:	
Exceptions:	

Use case Name:	**Provide missing ingredients**
Actor:	Store
Description:	UC that allows user to get missing ingredients directly through the app by buying them.
Preconditions:	
Postconditions:	Registered User is now provided with missing ingredients by the app
Normal Course of Events:	The Registered User navigates to the “shopping list” page, presses the “Add missing ingredient” button and is presented with all the information related to the ingredient, the price and the store. Then, the user confirms the list with the related button and presses on the “pay” button.
Alternative Courses:	The Registered User can add the missing ingredients in the Pantry page when ingredients are displayed
Exceptions:	

Use case Name:	**Review recipe**
Actor:	RegisteredUser
Description:	UC that allows Registered User to leave a review after cooking the selected recipe
Preconditions:	Registered User must be logged in; internet connection; recipe must have been made
Postconditions:	Other Registered Users can now have access to the review of the recipe
Normal Course of Events:	The Registered User taps on a recipe which has to be rated and selects the review option, generates the review and presses the confirmation button
Alternative Courses:	
Exceptions:	

Use case Name:	**Add recipe to favorites**
Actor:	RegisteredUser
Description:	UC that allows the Registered User to have a collection of his/her favorite recipes and to have easy and fast access to them.
Preconditions:	Cook at least one recipe
Postconditions:	The Registered User is now provided with a form to add recipes to favorites
Normal Course of	The Registered User navigates to the “recipe list” page, clicks

Events:	on the recipe he prefers and click the “add to favorites” button
Alternative Courses:	
Exceptions:	

Use case Name:	**Confirm shopping list**
Actor:	RegisteredUser
Description:	UC that allows the Registered User to confirm that the list of missing ingredients to buy is ready
Preconditions:	The Registered User has selected ingredients to buy
Postconditions:	The Registered User is ready to proceed with the payment
Normal Course of Events:	The Registered User has selected all the missing ingredients and can press the confirmation button to proceed with the payment
Alternative Courses:	
Exceptions:	

Use case Name:	**Payment**
Actor:	RegisteredUser,Store
Description:	Make payments for ingredient purchases from selected stores
Preconditions:	The Registered User has selected ingredients to buy and has confirmed the shopping list
Postconditions:	The Registered User has received the receipt
Normal Course of Events:	If bank information saved when sign up, then the Registered User can select one of the payment methods to pay his billing He confirms the payment and after the payment went through user will receive a receipt for recording
Alternative Courses:	
Exceptions:	

Use case Name:	**Provide list of recipes**
Actor:	RecipeDB
Description:	The app receives recipes from thousands of external sites
Preconditions:	
Postconditions:	
Normal Course of Events:	the algorithm inside the app allows the reception of recipes from thousands of external sites
Alternative Courses:	
Exceptions:	

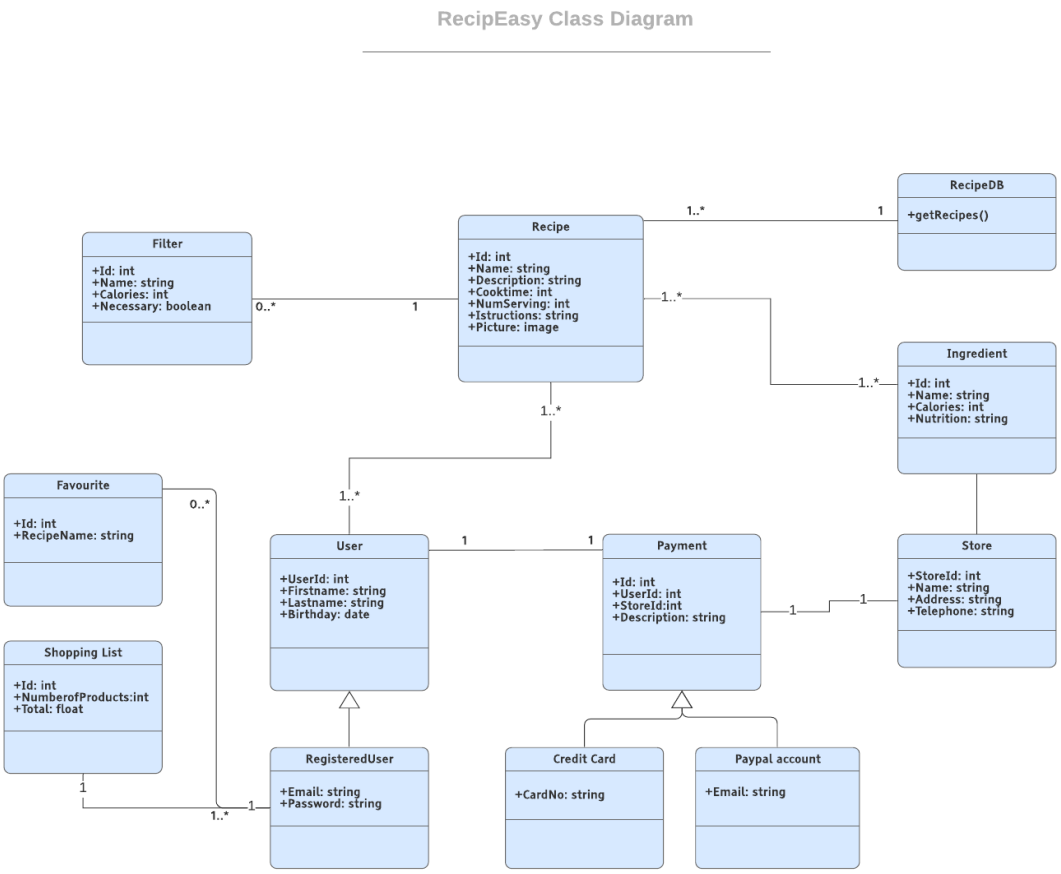
Sequence Diagrams

In the sequence diagram on the left it is possible to observe the interaction between the Regstered User and the Application system during the process of buying ingredients.
The whole precedure is managed by the app, so that the Registered User does not have any interaction with the store. Through the app, the Registered User adds or removes an ingredient to the shopping list. After doing this, the user can display the complete list and proceed to the purchase of the ingredients. At this point the app verifies the payment information and sends the order to the store that after receiving the payment accepts the order. Upon the confirmation from the store, the app sends to the Registered User a notification for the confirmation of the order. The store prepares the ingredients ordered and ships them to the Registered User. The Registered User can access information about his/her order status.

In the sequence diagram on the right it is possible to observe the interaction between the User and the Application system during the process of searching for recipes. The whole precedure is managed by the app, so that the User does not have any interaction with the Recipe database. Through the app, the user selects the ingredients for the recipe and inserts them to the pantry, based on this selection the app displays the pantry overall status to the user. The user can then choose how to filter the recipes based on his/her personal preferences. Based on the selection

of ingredients and filters, the app accesses the database and obtains the optimal list of recipes, which is then showed to the user. If the recipe the user chooses to cook contains ingredients the user does not own, he/she can buy them; the purchase option can be carried out directly by the app, if the user is registered, as explained in the Sequence Diagram 2.

Class Diagram



This class diagram gives an overview of the application and explains how it is structured.
It describes the types of objects in the system and the different types of relationships that exist among them. This diagram includes the class name, attributes, relationships and multiplicities.

8. Traceability

TM1	List of recipes	Define missing ingredients	Obtain filtered list of recipes	Filters	List of ingredients	Llist of ingredients with voice recognition	Payment information	Reviews	Review recipes	Save recipes to favourites	Receive payment	Obtain missing ingredients	Recipe list visualized
FUN 1													
FUN 2													
FUN 3					X								
FUN 4					X								
FUN 5			X	X									
FUN 6													X
FUN 7										X			
FUN 8													
FUN 9		X											
FUN 10												X	
FUN 11						X							
FUN 12	X												
FUN 13									X				
FUN 14								X					
FUN 15							X					X	

TM2	FUN 1	FUN 2	FUN 3	FUN 4	FUN 5	FUN 6	FUN 7	FUN 8	FUN 9	FUN 10	FUN 11	FUN 12	FUN 13	FUN 14	FUN 15
FUN 1															
FUN 2	x														
FUN 3				x											
FUN 4															
FUN 5												x			
FUN 6												x			
FUN 7	x	x										x			
FUN 8	x	x										x			
FUN 9	x	x													
FUN 10	x	x													
FUN 11															
FUN 12															
FUN 13	x	x										x			
FUN 14	x	x										x			
FUN 15	x	x								x					

In the first matrix the requirements are correlated with the intentional elements (goals, tasks and resources) of the to-be goal model.
The second matrix displays (on the rows and columns) the requirements of the app and the relationship between them.

9. Requirements Prioritization

	Requirements Statement
FUN1	The registered user can review recipes through the app
FUN2	The user can visualize ingredients in the pantry
FUN3	The user can filter the recipes obtaining personalized results in an effective way
FUN4	The user can visualize details of a certain recipe
FUN5	The registered user can add recipes to favorites
FUN6	The registered user can add ingredients to the pantry through voice command

Criteria C1: **implementation effort**

FOR C1	FUN1	FUN2	FUN3	FUN4	FUN5	FUN6
FUN1	1,00	5,00	0,20	1,00	1,00	0,33
FUN2	0,20	1,00	0,14	0,33	0,20	0,20
FUN3	5,00	7,00	1,00	5,00	5,00	3,00
FUN4	1,00	3,00	0,20	1,00	0,33	0,14
FUN5	1,00	5,00	0,20	3,00	1,00	0,33
FUN6	3,00	5,00	0,33	7,00	3,00	1,00
SUM	11,200	26,000	2,076	17,333	10,533	5,010

Criteria C2: **user value**

FOR C2	FUN1	FUN2	FUN3	FUN4	FUN5	FUN6
FUN1	1,00	0,20	0,11	0,14	0,20	0,33
FUN2	5,00	1,00	0,20	1,00	3,00	5,00
FUN3	9,00	5,00	1,00	3,00	7,00	9,00
FUN4	7,00	1,00	0,33	1,00	3,00	7,00
FUN5	5,00	0,33	0,14	0,33	1,00	3,00
FUN6	3,00	0,20	0,11	0,14	0,33	1,00
SUM	30,000	7,733	1,898	5,619	14,533	25,333

Standardized Matrix

FOR C1	FUN1	FUN2	FUN3	FUN4	FUN5	FUN6	Relative value
FUN1	0,089	0,192	0,096	0,058	0,095	0,067	0,100
FUN2	0,018	0,038	0,069	0,019	0,019	0,040	0,034
FUN3	0,446	0,269	0,482	0,288	0,475	0,599	0,427
FUN4	0,089	0,115	0,096	0,058	0,032	0,029	0,070
FUN5	0,089	0,192	0,096	0,173	0,095	0,067	0,119
FUN6	0,268	0,192	0,161	0,404	0,285	0,200	0,251
	1,000	1,000	1,000	1,000	1,000	1,000	1,000

Standardized Matrix

FOR C2	FUN1	FUN2	FUN3	FUN4	FUN5	FUN6	Relative value
FUN1	0,033	0,026	0,059	0,025	0,014	0,013	0,028
FUN2	0,167	0,129	0,105	0,178	0,206	0,197	0,164
FUN3	0,300	0,647	0,527	0,534	0,482	0,355	0,474
FUN4	0,233	0,129	0,176	0,178	0,206	0,276	0,200
FUN5	0,167	0,043	0,075	0,059	0,069	0,118	0,089
FUN6	0,100	0,026	0,059	0,025	0,023	0,039	0,045
	1,000	1,000	1,000	1,000	1,000	1,000	1,000

Prioritized requirements

FUN1	FUN2	FUN3	FUN4	FUN5	FUN6
0,100	0,034	0,427	0,070	0,119	0,251

Prioritized requirements

FUN1	FUN2	FUN3	FUN4	FUN5	FUN6
0,028	0,164	0,474	0,200	0,089	0,045

Consistency Index		(λmax - n)/ n-1	0,120343
Principal eigenvalue	6,601714		

Consistency Index		(λmax - n)/ n-1	0,115418
Principal eigenvalue	6,57709		

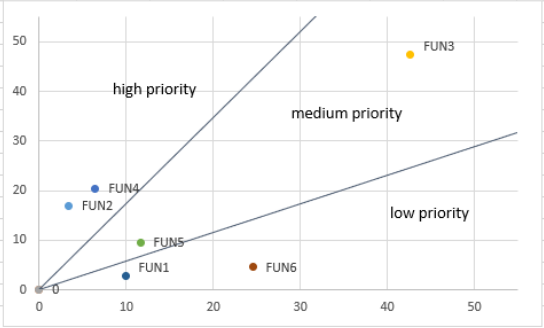
size of comparison matrix=*n*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Random consistency index (RI), for matrix of size n	0	0	0,58	0,9	1,12	1,24	1,32

CR=CI/RI **0,097051** IF CR< 10% the inconsistency is acceptable

CR=CI/RI **0,093079** IF CR< 10% the inconsistency is acceptable

	FUN1	FUN2	FUN3	FUN4	FUN5	FUN6
C1	9,897513	3,387101	42,65525	6,455198	11,62165	24,5293
C2	2,842544	16,88243	47,40201	20,4457	9,492886	4,569557

86,60254 50



The first six requirements have been considered for prioritization.
Using AHP for decision making involves five steps. We'll assume here that we want to evaluate candidate requirements using two criteria that will have the same weight: implementation effort and user value.
1.Set up the n requirements in the rows and columns of an n × n matrix
2.Perform pairwise comparisons of all the requirements according to the criterion
3 Perform the standardized matrix and rank the requirements using the relative values
4.Calculate CI,CR
5.Plot contributions on value-cost diagram

The consistency index (CI) is a first indicator of result accuracy of the pairwise comparisons. CI = (λ max− n)/(n −1), λmax denotes the maximum principal eigenvalue of the comparison matrix. The closer the value of is to n (the number of requirements), the smaller the judgmental errors and thus the more consistent the result.
The consistency indices of randomly generated reciprocal matrices from the scale 1 to 9 are called the random indices, RI. The ratio of CI to RI for the same-order matrix is called the consistency ratio (CR), which defines the accuracy of the pairwise comparisons.
A CR of 0.10 or less is considered acceptable.
The cost–value diagram clearly facilitates requirements selection, by not implementing the requirements that contribute little to stakeholder satisfaction, we can significantly reduce the cost and duration of development.
Since the requirements taken in consideration were only a few, they are all notable and necessary so it appears that only two of them have low priority.
The FUN3 clearly has the highest importance for the user but considering that it has the highest implementation effort, has an average priority compared to the other requirements.

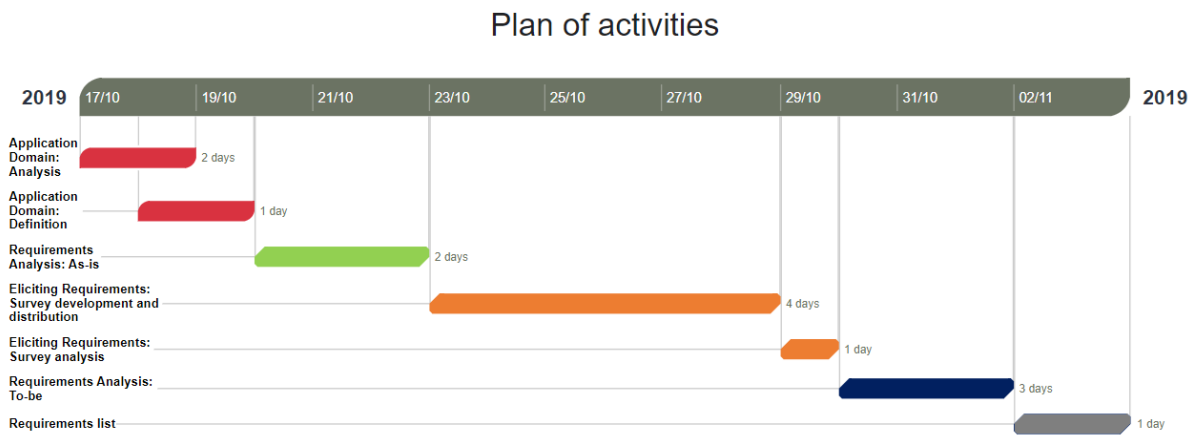
10. User Feedback Analysis

I chose to do the sentiment analysis manually with a sample of 8 Evernote reviews.
It would have been better to take a sample of 50 to 100 reviews in order to get an accurate idea of the guests' sentiments. It'd be perfect to do an analysis on all the reviews, but that's a timely, tedious task that can be resolved using suitable software. However the benefit of doing it manually is to look at the reviews in a much more detailed way, which can help to get a real sense of what people think.
The related spreadsheet includes different columns that identifies the source, the star rating and the review for reference.
I have decided to highlight positive phrase of the reviews green, neutral yellow, and negative red because colors will help to identify how people talk about different areas of property.
Reviews has been quantified numerically using the phrases highlighted giving them one point for each positive phrase, a zero for neutral, and a negative point for a negative review.
Once each sentence has been assigned a score, add them up to determine the overall review general feeling.
As we can see, in the selected sample there is an equality between positive and negative feelings regarding the application, a great variety of negative feedback and also a general appreciation of the app.

Then, using those same highlighted phrases, I made a list of everything mentioned topic and I gave them a score depending on how they were mentioned in the review.
The higher the score, the better people felt about that aspect of your property and vice versa, so it is possible in this way to determine how the general public feels about each aspect.

11. Annexes

11.1 Plan of activities (e.g. Gantt chart)



11.2 GO model source files

The “as-is.txt”, “to-be.txt”,”as-is.png”, “to-be.png” are provided together with this document

11.3 Survey questionnaire

https://docs.google.com/forms/u/1/d/1TpezixkfVenXe5sxjJe4Cto0pi4I3fLUzWBePu5mqhA/edit?usp=drive_web

11.4 Data collected with the survey

The “RecipEasy questionnaire results” google sheet is provided together with this document.

11.5 Model for requirements prioritization

"Req.Prioritization.xlsx." file containing the calculations used for assessing the priorities, is available in directory of this document

11.6 User feedback analysis documentation

"Sent.analysis-topicmodel.xlsx” file containing the reviews analysis and “evernote.csv” file from which I extracted the reviews, are available in directory of this document.