



SAPIENZA  
UNIVERSITÀ DI ROMA

*Human Computer Interaction 2018/2019*

## PintFinder

**Presented by:**

Alessandro Lo Presti  
Leonardo Di Paolantonio  
Valerio Longo

**Submitted to:**

Tiziana Catarci  
Valeria Mirabella

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Requirement analysis</b>	<b>5</b>
2.1	Persona 1: the newbie . . . . .	5
2.1.1	User Profile . . . . .	5
2.1.2	Description . . . . .	5
2.1.3	Scenario . . . . .	5
2.2	Persona 2: The expert . . . . .	6
2.2.1	User Profile . . . . .	6
2.2.2	Description . . . . .	6
2.2.3	Scenario . . . . .	6
2.3	Persona 3: The pub owner . . . . .	7
2.3.1	User Profile . . . . .	7
2.3.2	Description . . . . .	7
2.3.3	Scenario . . . . .	7
2.4	Persona 4: The admin . . . . .	8
2.4.1	User Profile . . . . .	8
2.4.2	Description . . . . .	8
2.4.3	Scenario . . . . .	9
<b>3</b>	<b>Competitors</b>	<b>10</b>
<b>4</b>	<b>Questionnaire analysis</b>	<b>11</b>
4.1	General information . . . . .	11
4.2	Questions about the users preferences . . . . .	12
4.3	Evaluation of user knowledge . . . . .	14
4.4	Pub Owner questions . . . . .	15
4.5	Conclusions . . . . .	16
<b>5</b>	<b>Mockup</b>	<b>17</b>
5.1	Signing-Up . . . . .	17
5.2	The <b>PubLover</b> . . . . .	18
5.3	The <b>PubOwner</b> . . . . .	20
<b>6</b>	<b>Task Analysis: HTA &amp; STN</b>	<b>22</b>
6.1	Hierarchical Task Analysis . . . . .	22
6.2	State Transition Network . . . . .	22

6.3	A pub lover updates his "tasted beers" list . . . . .	23
6.4	A pub owner adds a new beer to his menù . . . . .	25
<b>7</b>	<b>First Prototype</b>	<b>27</b>
7.1	Adding of admin persona . . . . .	27
7.2	New HTA & STN of adding a beer as pubOwner . . . . .	27
7.3	The booking feature . . . . .	28
7.4	The offers feature . . . . .	29
<b>8</b>	<b>Expert Based Evaluation</b>	<b>31</b>
8.1	Heuristic Evaluation . . . . .	31
8.2	Molich and Nielsen's Heuristics . . . . .	31
8.3	Expert Report . . . . .	32
<b>9</b>	<b>Second Prototype</b>	<b>34</b>
<b>10</b>	<b>Think Aloud</b>	<b>42</b>
10.1	Conclusions of the think aloud session . . . . .	43
<b>11</b>	<b>Controlled Experiment</b>	<b>44</b>
11.1	The Problem . . . . .	44
11.2	The ANOVA results . . . . .	46
<b>12</b>	<b>Final Product</b>	<b>48</b>
12.1	The PubLover . . . . .	49
12.2	The PubOwner . . . . .	53
<b>13</b>	<b>Conclusions</b>	<b>56</b>
13.1	Future works . . . . .	56
13.2	References . . . . .	56

# 1 Introduction

## The idea behind PintFinder

Today is very easy to find all the nearest pubs nearby, but it's almost impossible to find the "right one". In fact, there are no efficient systems that can let the user know in advance what type of beer is on the menu, the relative prices, how much free spots are left, if there is some special offers for a certain beer, etc..

The *PintFinder* project comes out from this necessity: drive the user to the most suitable pub for its needs. This Android application is made to connect the client *Pub Lover* to the *Pub Owner*. Furthermore, the App provides the *PubLover* with a system of beer rating, so that he always has the list of all the ones he likes best or hates. The application is focused more on beers offered by a certain pub rather than on the general reviews provided.

## **2 Requirement analysis**

### **2.1 Persona 1: the newbie**

#### **2.1.1 User Profile**

Name: Angelo

Age: 24 years

Gender: Male

Location: Rome

Education: University student

Income: 0 \$

Technology: Smartphone friendly

Family: Single

#### **2.1.2 Description**

Angelo is a student from Sapienza University of Rome. He is planning to work in Artificial Intelligence company in Google, California. He has a huge passion for soccer but he doesn't play. He just loves watching matches on tv, especially for supporting A.S. Roma. He is always up to date with computer and smartphone technologies. Angelo usually spends his time with his friends going to pubs to watch sports of any kind.

#### **2.1.3 Scenario**

It's Friday afternoon and Angelo with his colleagues have just completed the Algorithm Design lecture. Since tonight his favourite team is going to play the Champions League final in Bruxelles, he wants to enjoy the match while drinking a good beer with his friends. He doesn't want to get too tipsy to not miss any details that is happening on the pitch. Angelo runs PintFinder and

searches for the closest pub with a wide range of light beers. At this point, he finds the Lancelot, a pub offering lots of Belgian beers and he decides to go there like lucky charm.

## **2.2 Persona 2: The expert**

### **2.2.1 User Profile**

Name: Georg

Age: 29 years

Gender: Male

Location: Milan

Job Title: Employee

Income: 35,000 \$ /year

Technology: Smartphone friendly

Family: Married

### **2.2.2 Description**

Georg is an employee at CapGemini in Milan. His main task, as a counselor, is to take care of company customers by giving them the best advice to grow their business. Since he was a child, one of his hobbies is practicing magic tricks. He found out it is a good way to make new friends as well. As a German, he loves beer and he participated to many beer festivals all over Germany.

### **2.2.3 Scenario**

It is Friday night, after a long shift of work, Georg just wants to have one of his favourite beers with his colleagues. They don't care too much about beers, but Georg, as a German beer lover, every time he steps into a Pub,

he struggles in finding what he was searching for: a certain beer with his preferred characteristics. This time he decided to use PintFinder for checking in advance nearby pubs with the beers he would like to drink.

### **2.3 Persona 3: The pub owner**

#### **2.3.1 User Profile**

Name: Maria

Age: 36 years

Gender: Female

Location: Rome

Job Title: Business Owner

Income: 20,000 \$ /year

Technology: Smartphone friendly

Family: Engaged

#### **2.3.2 Description**

Maria is owner of PubAmo, a pub with a wide selection of beers. She loves watching movie and drinking beer, so she has just installed on her pub a 4kHD tv. She is a tattoo lover: she has about 20 tattoos all over her body and she is even planning to go for more. She is planning to extend her customer base but she doesn't know how to do it. Maria knows everything about beers and she is planning to create her own brewery. Maria has a lot of friends because she is very nice with everyone.

#### **2.3.3 Scenario**

It is Wednesday night and Maria is thinking about how to make the Pub more crowded during the week. She would like to promote it in such a way

that as many people as possible can be aware about the beers offered by the Pub. Since she already tried using traditional methods as leaflets, without receiving the expected feedback, she decides to try a smarter approach. So she runs PintFinder and, logging as a Pub Owner, she puts her Pub on the platform providing all the information about her beer menu.

## 2.4 Persona 4: The admin

This persona has been created and added after the first prototype, so it will not appear in the mockups.

### 2.4.1 User Profile

Name: Marco

Age: 21 years

Gender: Male

Location: Naples

Job Title: Student

Income: 0 \$

Technology: Smartphone friendly

Family: Single

### 2.4.2 Description

Marco is a student from Federico II University of Naples. He is planning to work as manager to Amazon Warehouses. He has a huge passion for videogames and sometimes wins in official tournaments. He loves watching movies on tv, especially teen dramas. He is always up to date with computer and smartphone technologies.

### **2.4.3 Scenario**

It is Thursday afternoon, Marco is playing with his smartphone and suddenly arrives a notification from PintFinder: A pub owner requested the add of a very specific beer made in a small town near Innsbruck. Marco, as admin, has the purpose to add new beers and is the only one who can apply edits to the global list of the beers. So, he access the list and adds the Innsbruck beer.

*NOTE: the implementation of this persona is part of "Future Work".*

### 3 Competitors

By a brief research on the Play Store, we found four possible competitors:

- The Pub Finder
- Tipple
- Pub Finder
- Pubster

Now we analyze each competitor in detail:

1. *The Pub Finder* does not provide a duality between a Pub Lover and a Pub Owner. There is a mechanism that allows the user to send a new Pub (just the name and the address) to an administrator that will eventually add it to the system. In addition provides a review system in which an user can add a comment and a rating about the Pub.
2. *Tipple* presents a more update interface, but the main view is a bit confused, as an new user I don't feel very comfortable about the meaning of the various button on the screen. Moreover it seems to be very slow for performing certain simple operations on the map. This app is only on the search of Pubs and Bar, mainly social based.
3. *Pub finder* exposes an old fashioned mechanism of continuous suggestions to the user, that has to decline or accept the offer to drink at the suggested pub. Moreover this application was never updated after the release (April 2013).
4. *Pubster* has a very minimal interface and it's the only app to provide a fidelity sale system based on Qr-codes. However it is not beer-centered and does not provide any description of menùs with relative prices. Pubster is not updated since January 2017 and the servers appears closed.

All these Apps are not focused about the beers offered by the Pub but just on the Pub as a public place for enjoying free time. None of them is about giving to the user a beer prospective. Instead, our aim is to have a **beer-centered** service.

## 4 Questionnaire analysis

To collect the requirements, it was decided to proceed with a questionnaire

### 4.1 General information

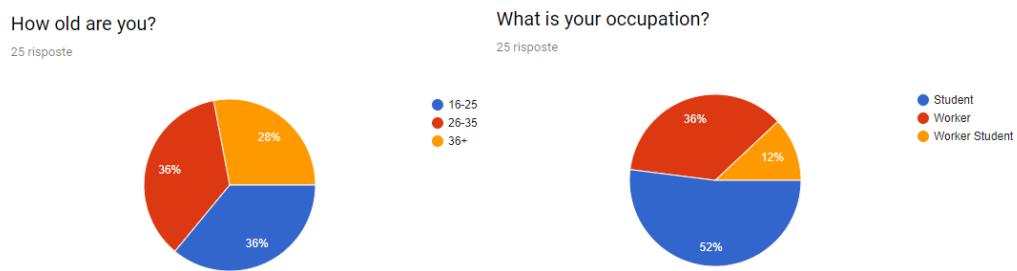


Figure 1: Demographic Information

## 4.2 Questions about the users preferences

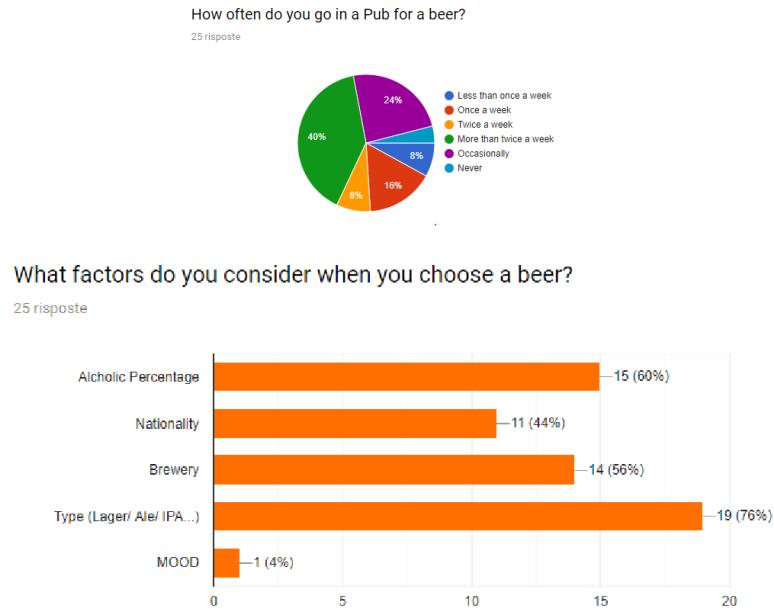
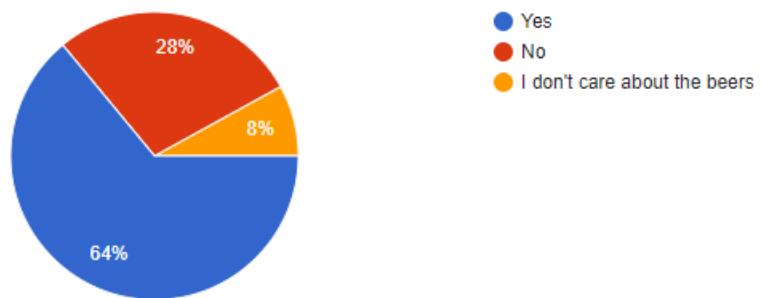


Figure 2: Customers preferences

Do you struggle in finding the Pub with the right beers for you?

25 risposte



Would you like to know the prices of the beers of a Pub before entering?

25 risposte

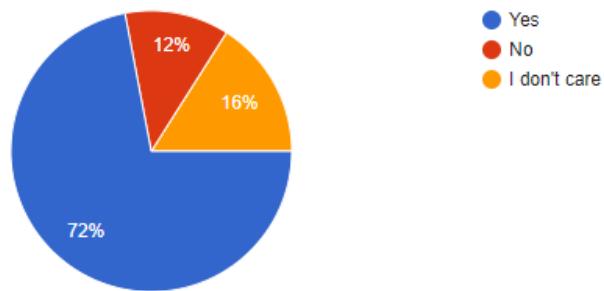


Figure 3: Customers habits

### 4.3 Evaluation of user knowledge

How much do you think to know about Beer?

15 risposte

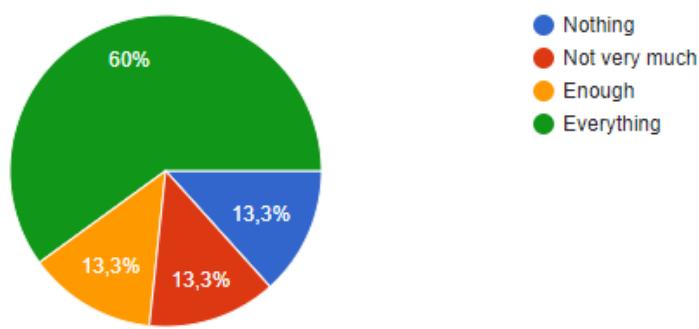
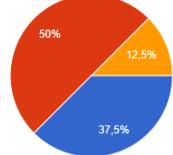


Figure 4: How the users perceive their knowledge

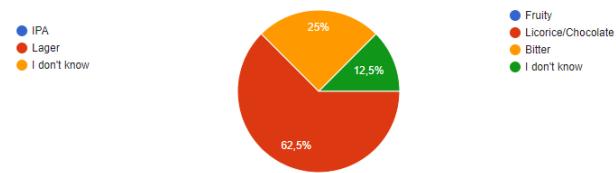
Which beer takes less time to ferment?

8 risposte



What is the typical taste of Stout beer (e.g. Guinness)?

8 risposte



What are the roots of Trappist beer?

8 risposte

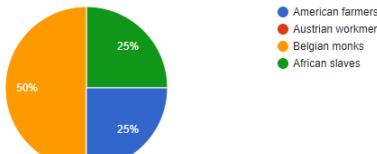
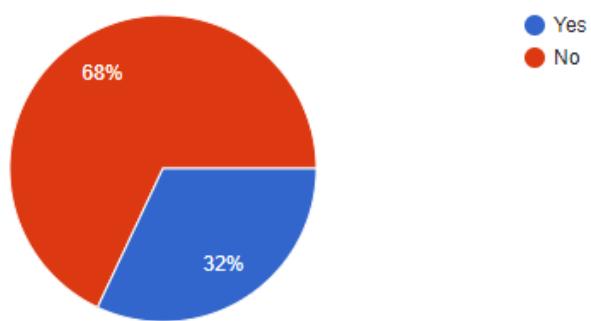


Figure 5: Three questions of incremental difficulty to test the user

## 4.4 Pub Owner questions

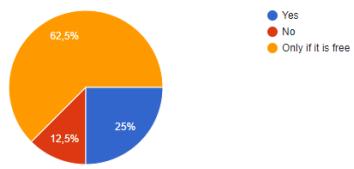
Are you a Pub owner?

25 risposte



Would you like to have a platform for sponsoring your Pub?

8 risposte



Would you share your beer menu with all information (description, nationality, prices...)?

8 risposte

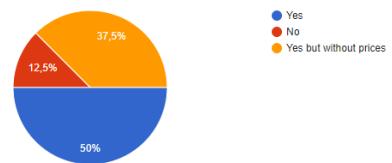


Figure 6: Questions about the desire and the conditions to sponsor their own business

## 4.5 Conclusions

From this questionnaire it emerged that:

- Every user uses a different scale to choose the right beer.
- There is a **real needs** of an application beer-oriented.
- Not all those who call them-selves experts actually are.
- the majority of pub owners wants a free application to share information about menù and beers.

## 5 Mockup

### 5.1 Signing-Up

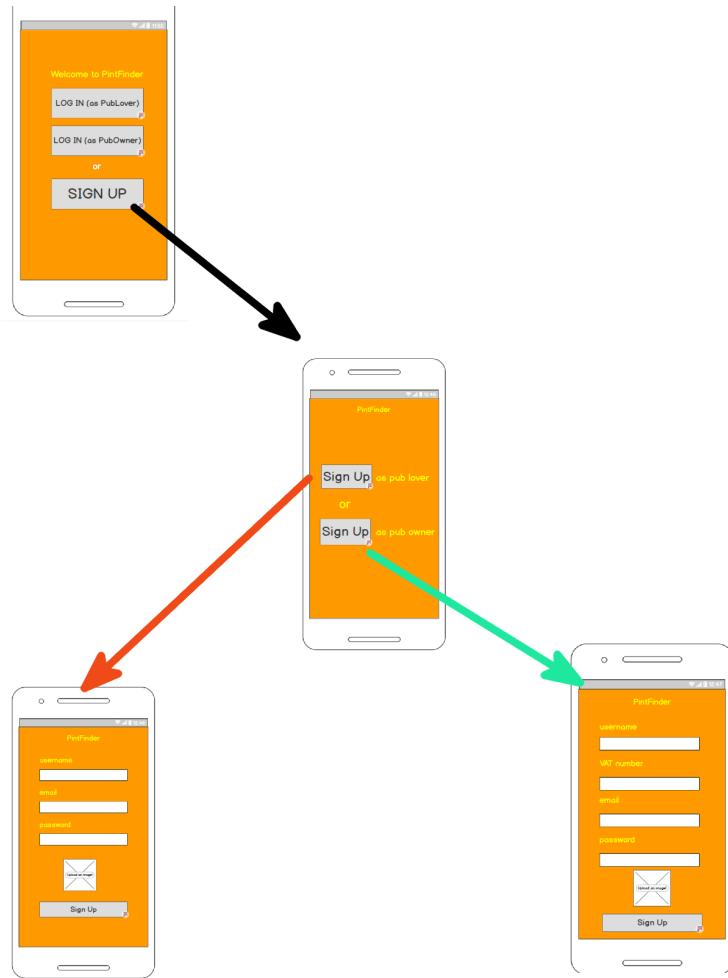


Figure 7: Steps to sign-up

As we said, the app can be used both from customer (to search a pub) that from a pub owner (to sponsor his business). So we have to split first the two

types of users.

## 5.2 The PubLover

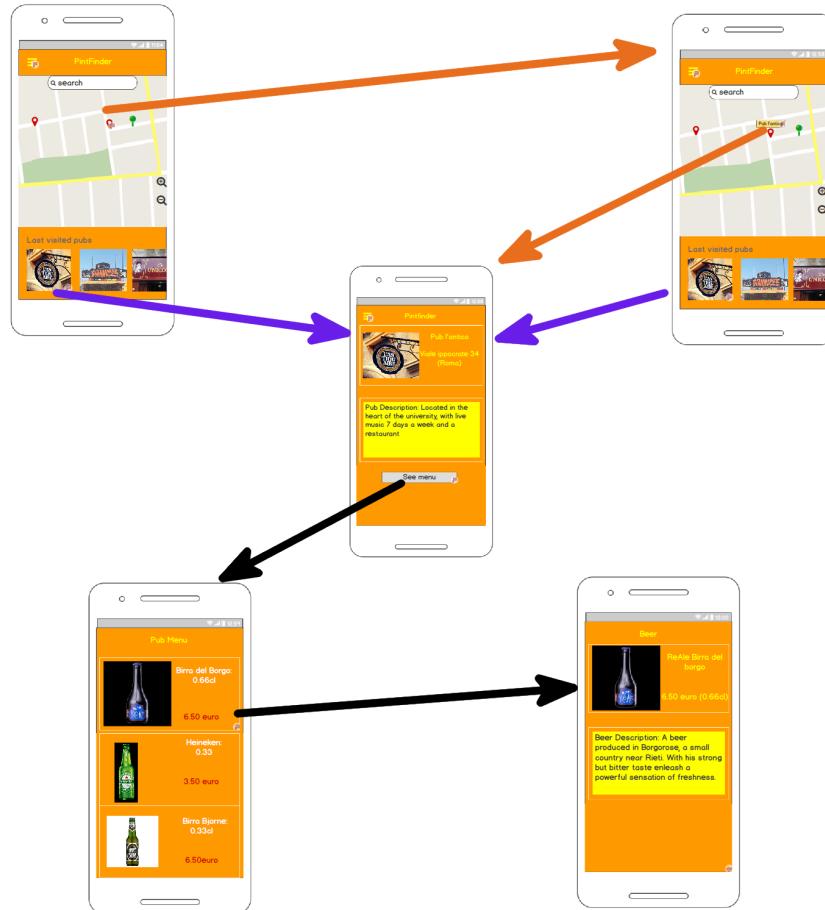


Figure 8: Steps to view a particular beer of a certain pub

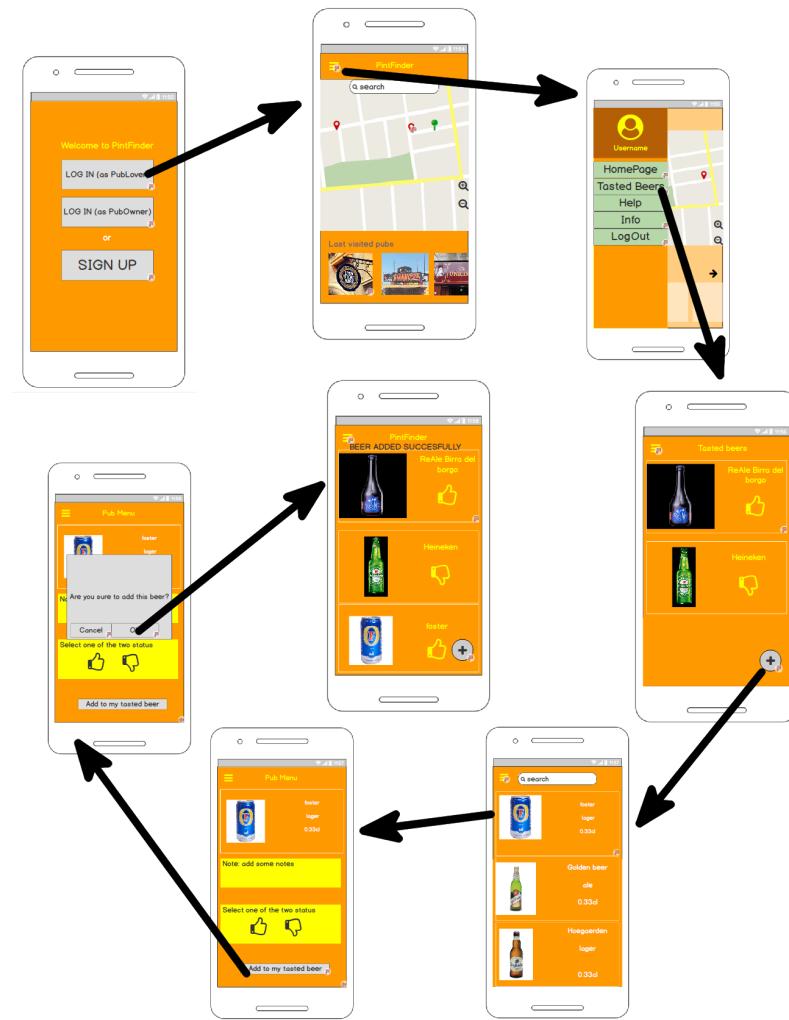


Figure 9: Steps to add a new beer in the "tasted beer" menu

### 5.3 The PubOwner



Figure 10: Steps to add a new pub. A PubOwner can manage multiple pubs

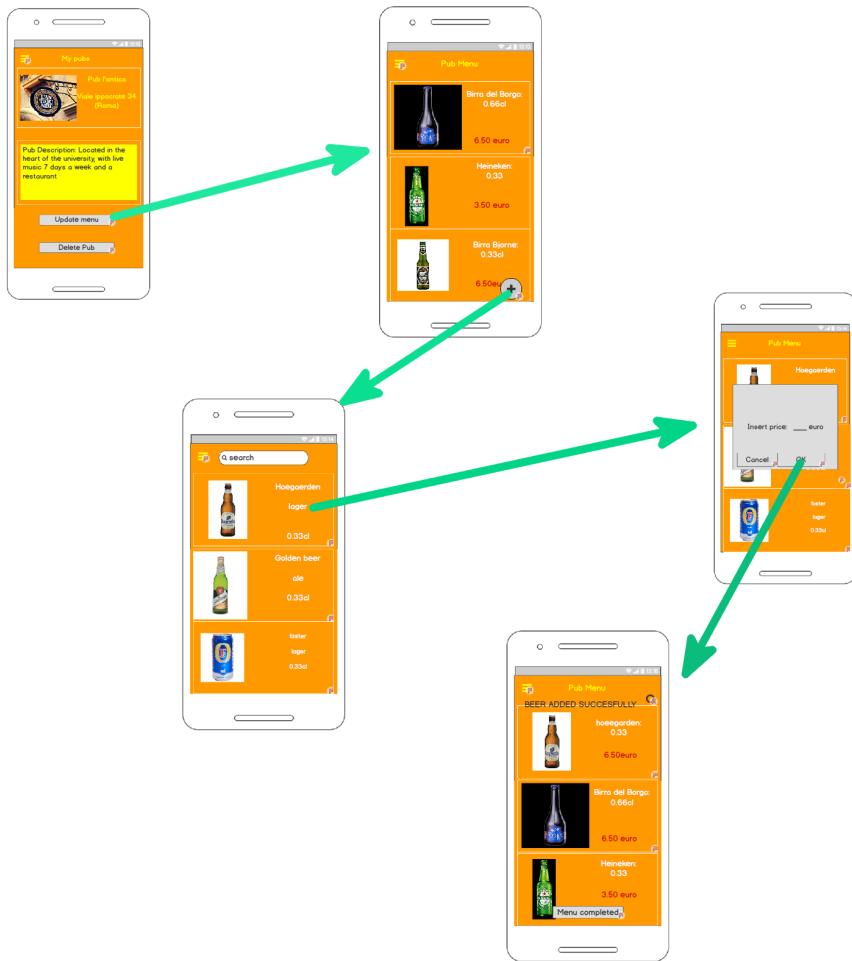


Figure 11: Steps to add a new beer in pub owner menù

## 6 Task Analysis: HTA & STN

We decided to make the focus this kind of analysis on three main actions:

- A pub lover seeks for a pub
- A pub lover updates his "tasted beers" list
- A pub owner adds a new beer to his menù

### 6.1 Hierarchical Task Analysis

Hierarchical task analysis is a task description method and a variant of the *task analysis*. This is used to produce an exhaustive description of tasks in a hierarchical structure of goals, sub-goals, operations and plans. In this kind of analysis, tasks are broken down into pieces progressively smaller.

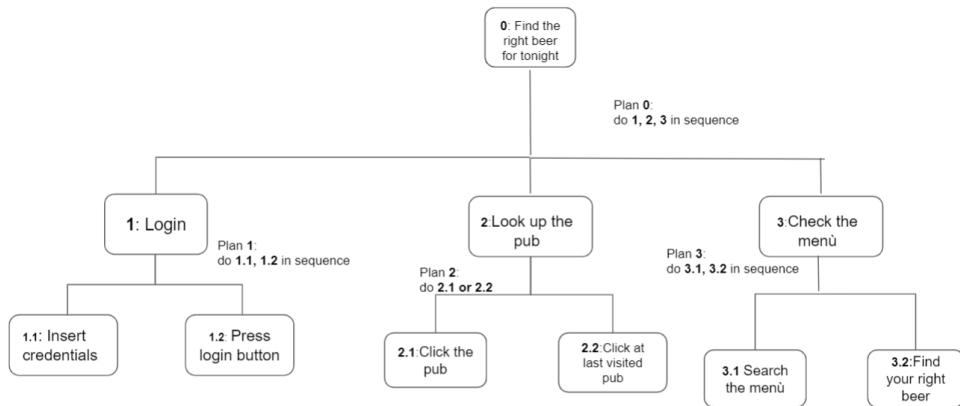


Figure 12: HTA for: A pub lover seeks for a pub

### 6.2 State Transition Network

A state transition network is very intuitive diagram. It assumes that a dialog essentially refers to a progression from one state (or node) of the system to the next in the system state space.

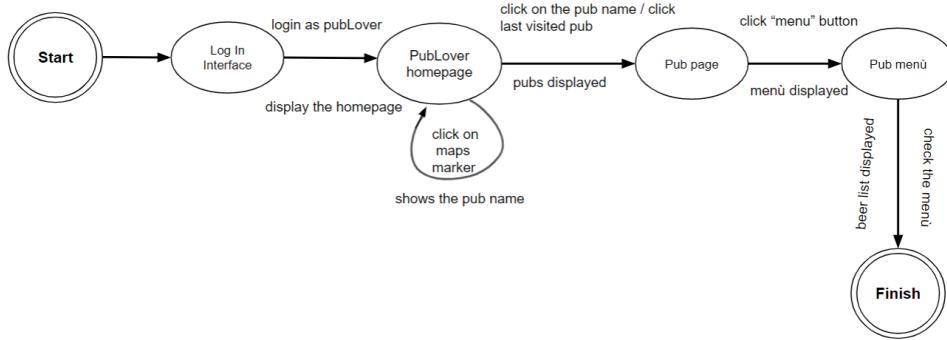


Figure 13: STN for: **A pub lover seeks for a pub**

To perform this task, the user must log in as PubLover, once he clicked on the maps marker inside the homepage, the pub is displayed. If he really wants to go in that pub, can check the menu.

### 6.3 A pub lover updates his "tasted beers" list

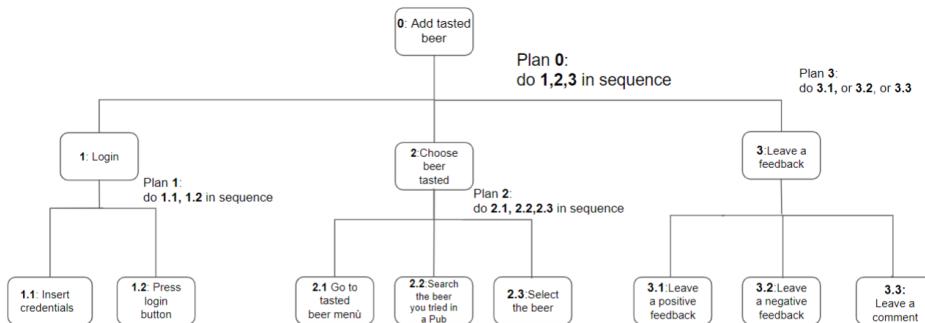


Figure 14: The Hierarchical Task Analysis

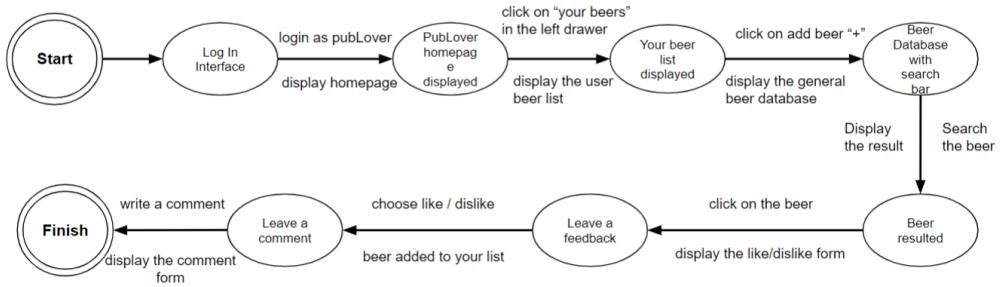


Figure 15: The State Transition Network

To perform this task the user must first log in as PubLover. Once he clicked on the drawer menù, he must clicks on the "tasted beers" section. Once it appeared the entire list of tasted beers, he must click on the plus button and search the beer. After choose the right one, leave a feedback (like/dislike) and an optional comment.

## 6.4 A pub owner adds a new beer to his menu

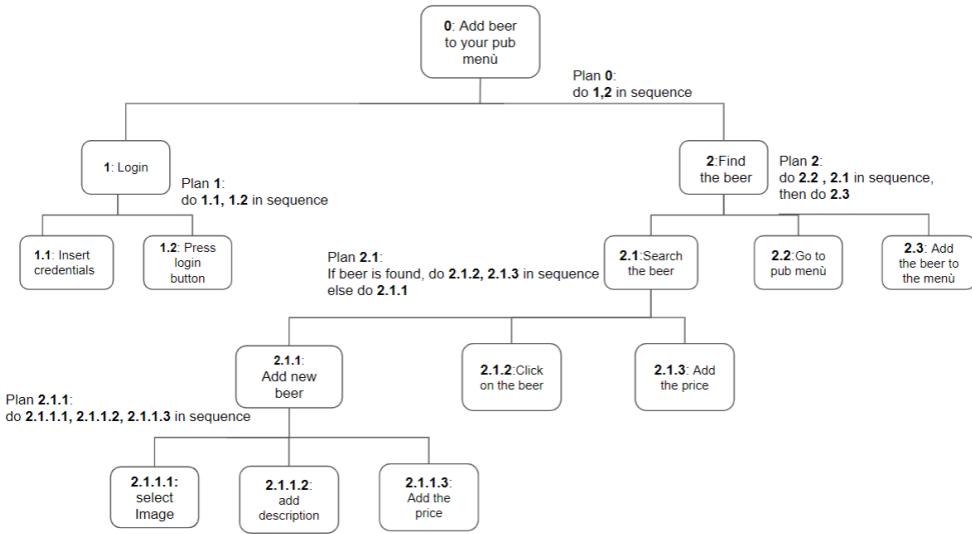


Figure 16: The Hierarchical Task Analysis

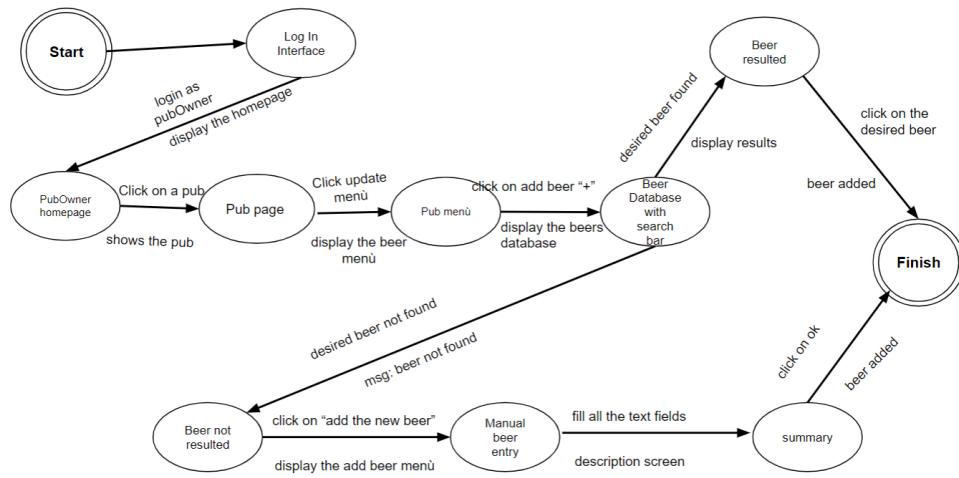


Figure 17: The State Transition Network

To perform this task the user must first log in as PubOwner and then click on the pub menu inside the homepage. Once he clicked on the plus button inside the pub menu he must search the beer in the PintFinder database. If he finds the beer he wants to add, then he has finished. Else, he must manually add it.

## 7 First Prototype

In this development phase we added several features:

- The adding of the *admin persona*
- A PubLover can book a pub spot available.
- A PubOwner can make offers on his beers menù.

### 7.1 Adding of admin persona

We decide to create the *admin persona* because we only noticed at this stage that if two or more pubOwners want to add the same beer that is not in the database, each one must update his menu locally. With the admin instead, just the first pubOwner send a request and the database is updated **globally** for every pubOwners.

### 7.2 New HTA & STN of adding a beer as pubOwner

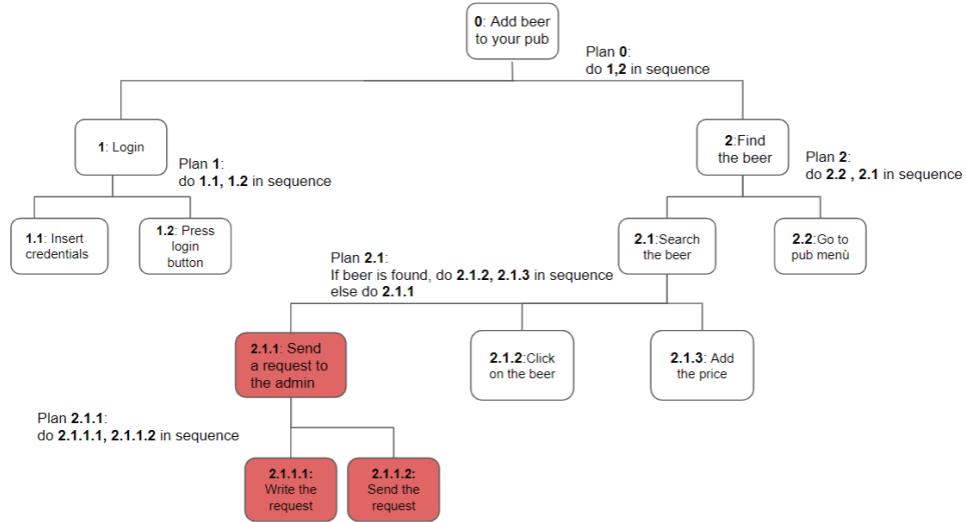


Figure 18: The Hierarchical Task Analysis

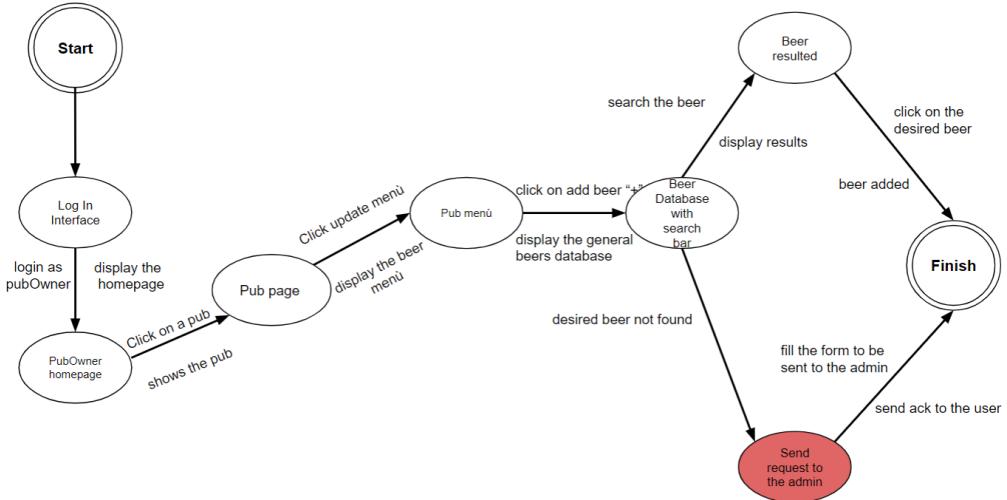


Figure 19: The State Transition Network

To perform this task the user must first log in as PubOwner and then click on the pub menù inside the homepage. Once he clicked on the plus button inside the pub menù he must search the beer in the PintFinder database. If he finds the beer he wants to add, then he has finished. Else, he must send a request to the admin to add a new beer on the database.

### 7.3 The booking feature

We noticed that if a PubLover want to go in a certain pub, it can happen that the pub is full and no table is available. Due to this fact we thought that, instead of calling the pub to reserve a spot, the customer books his spot directly on the App, gaining time.

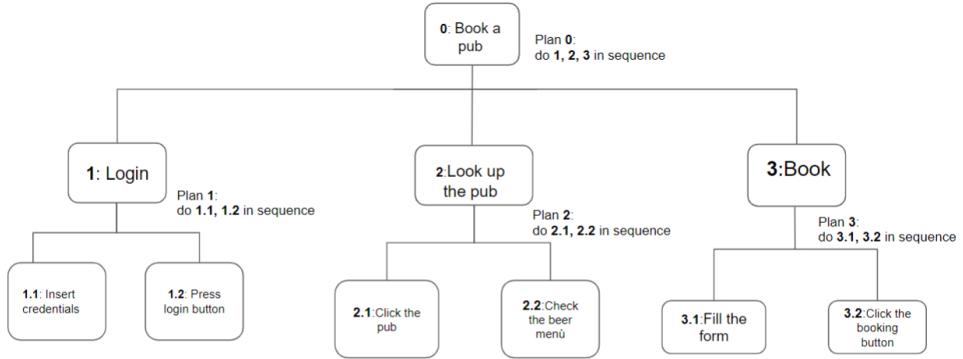


Figure 20: HTA of the book feature

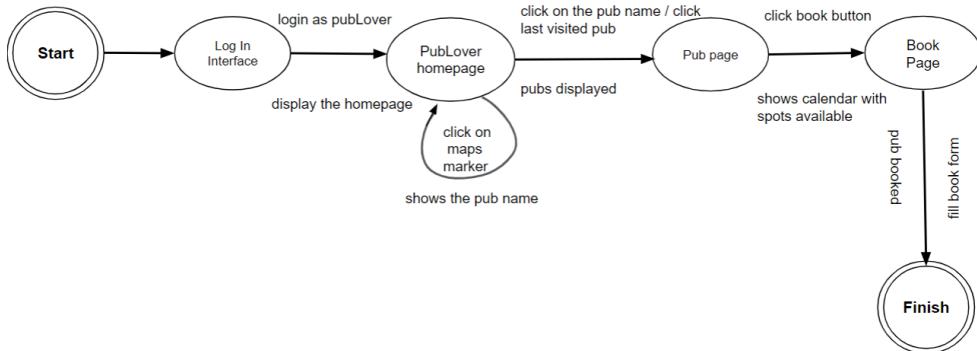


Figure 21: STN of the book feature

## 7.4 The offers feature

Moreover, we noticed that if a PubOwner want rightly sponsor his business, maybe he wants to make offers and deals on his beers like promotion, sales etc...

So, We implemented a simple mechanism of publishing offers. In that way,

all pubLovers whom go to any pub description can see if there are offers and which ones.

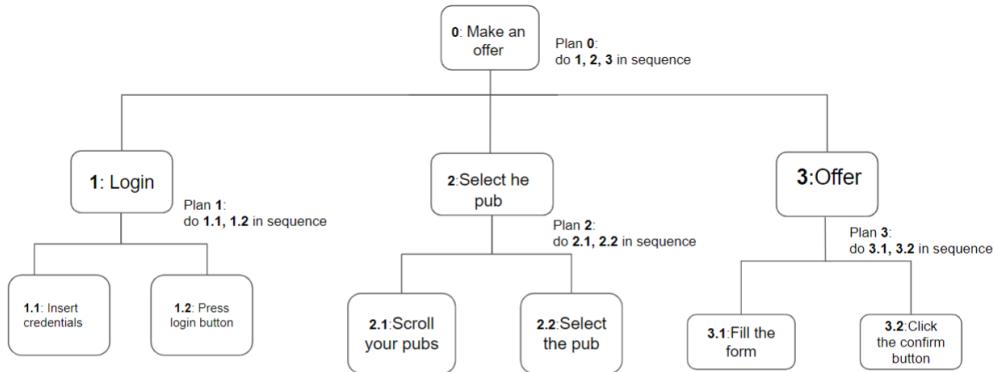


Figure 22: HTA of the offer feature

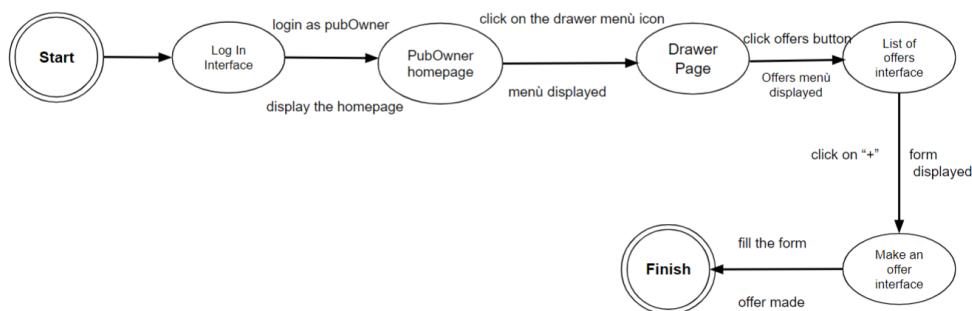


Figure 23: STN of the offer feature

## 8 Expert Based Evaluation

Evaluation occurs in laboratory or field with the support of users. They should be considered at all stages in the design life cycle, especially on the embryonic phases. The evaluations assess extent of system functionality, effect of the user interface on the user (the user's experience of the interaction e.g., easy to learn, easy to use, satisfaction), identify specific problems (e.g., errors, confusion, unexpected results). They are based on the response of an external expert that **does not have any role on the development**.

### 8.1 Heuristic Evaluation

A kind of expert based evaluation is the *heuristic* one. This kind of evaluation is based on some heuristics as standard to measure the project. In our case, due to the fact that is practically difficult using these evaluations until the finished product, we used them only on the first prototype. In particular we used the **Molich and Nielsen's Heuristics** as kind of **heuristic evaluations**.

### 8.2 Molich and Nielsen's Heuristics

Molich and Nielsen developed a heuristics list which is composed by 10 heuristics:

1. **Visibility of system status:** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. **Match between system and the real world:** The system should speak the user's language, following real-world conventions, using words, phrases and concepts familiar to the user and making information appear in a natural and logic order, rather than using system-oriented terms.
3. **User control and freedom:** Given that users often make mistakes, choosing the wrong system function, they need a clearly "emergency" exit to leave the unwanted state. Because of that, the system should support undo and redo.

4. **Consistency and standards:** Follow platform conventions so that users don't have to understand if different words, situations or actions mean the same thing.
5. **Error prevention:** Having a careful design which prevents a problem from occurring in the first place is better than a good error message, so eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6. **Recognition rather than recall:** Minimize the user's memory load by making objects, actions and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. **Flexibility and efficiency of use:** Allow users to personalize frequent actions with accelerators. Accelerators, unseen by the novice user, may often speed up the interaction for the expert user such that the system can provide for both inexperienced and experienced users.
8. **Aesthetic and minimalist design:** Dialogues should not contain information which is irrelevant or rarely needed, because every extra unit of information in a dialogue, competes with the relevant units of information reducing their visibility.
9. **Help users recognize, diagnose and recover from errors:** Error should not be expressed in codes, but in plain language, precisely indicating the problem and constructively suggesting a solution.
10. **Help and documentation:** It may be necessary to provide help and documentation, even though it is better if the system can be used without documentation. Any information should be easy to search, focused on the user's task, so make a list of concrete steps to be carried out, and not be too large.

### 8.3 Expert Report

After the expert based evaluation, it has been reported that the following heuristics have been violated:

## Heuristic Evaluation

Date: 26.05.2019

Evaluator: Valeria Mirabella

Prototype: PintFinder

Frame	Heuristic violated	Severity	Description / Comment
All	User control and freedom	3	Allow user to recover from error if he choose by mistake the wrong action putting a back button. Remember you should be consistent in every page.
Search	Flexibility and efficiency of use	2	Allow favorite searches for experienced users
Home	Visibility of the system status	3	Keep user informed about the profile he used for login and if logged or not (if applicable)
Home page new pub lover	Recognition rather than recall	2	Add some details on last visited pubs adding for example the data, notes of rating left
Tasted beer	Match between the system and the real world	3	User could be interested in the availability of the tasted beer in the pubs
Add beer as pun lover	Flexibility and efficiency of use	3	Supply search engine and accelerators
Lovers see pub menu beer	Recognition rather than recall	3	Help user is remember if he already tasted the beer and give the information available in tasted beer section
Lovers see pub menu beer	User control and freedom	4	Impossible to exit the page
All	Help and documentation	2	Provide help and documentation
Help	Recognition rather than recall	3	Usually personal information are not in help section. It will be not easily retrievable. Moreover help is missing
Create menu	Flexibility and efficiency of use	3	Supply search engine and accelerators

Figure 24: The report of the evaluation

As levels of severity as been used :

- 0 I don't agree that this is a usability problem at all
- 1 Cosmetic problem only
- 2 Minor usability problem
- 3 Major usability problem
- 4 Usability catastrophe

## 9 Second Prototype

After the expert evaluation report, we tried to solve the most serious problem. The highest severity problem ( level 4) was:

- Impossible to exit the page

This was a problem strictly bounded to the mockups. In fact the back button in android is implemented and through Balsamiq we don't know how to implement the possibility to go back except doing that in the presentation mode. However, we added an home button to return to the homepage. So, during the implementation of the app this problem **was fully resolved**.

The other major problems were:

- 1 - *Allow user to recover from error if he choose by mistake the wrong action putting a back button.*

-Severity : 3.

We solved with confirmation pop-ups. Here some examples:

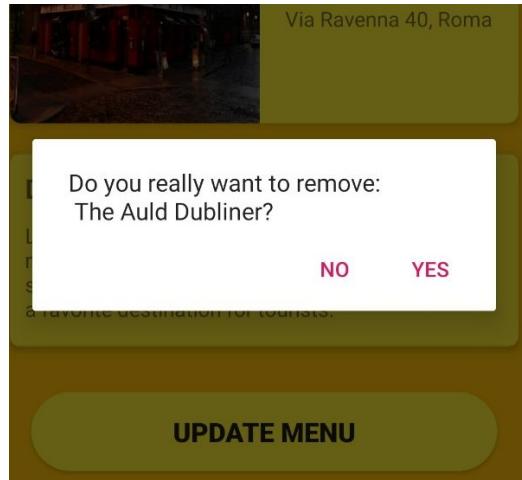


Figure 25: confirmation pop-up when a PubOwner wants to delete a pub.

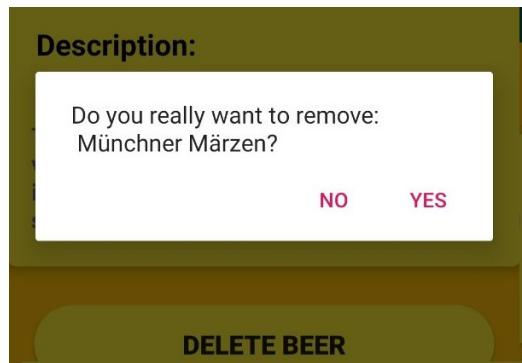


Figure 26: confirmation pop-up when a pubOwner wants to delete a beer from his menù.

- 2 - *Allow favourite searches for experienced users*

Severity : 2.

We solved adding an advance search feature. With this feature the expert user can choose the right beer and clicking on the "check pub near you" button can find all the pubs in the neighborhood with that beer. This button has been added even in the tasted beer list

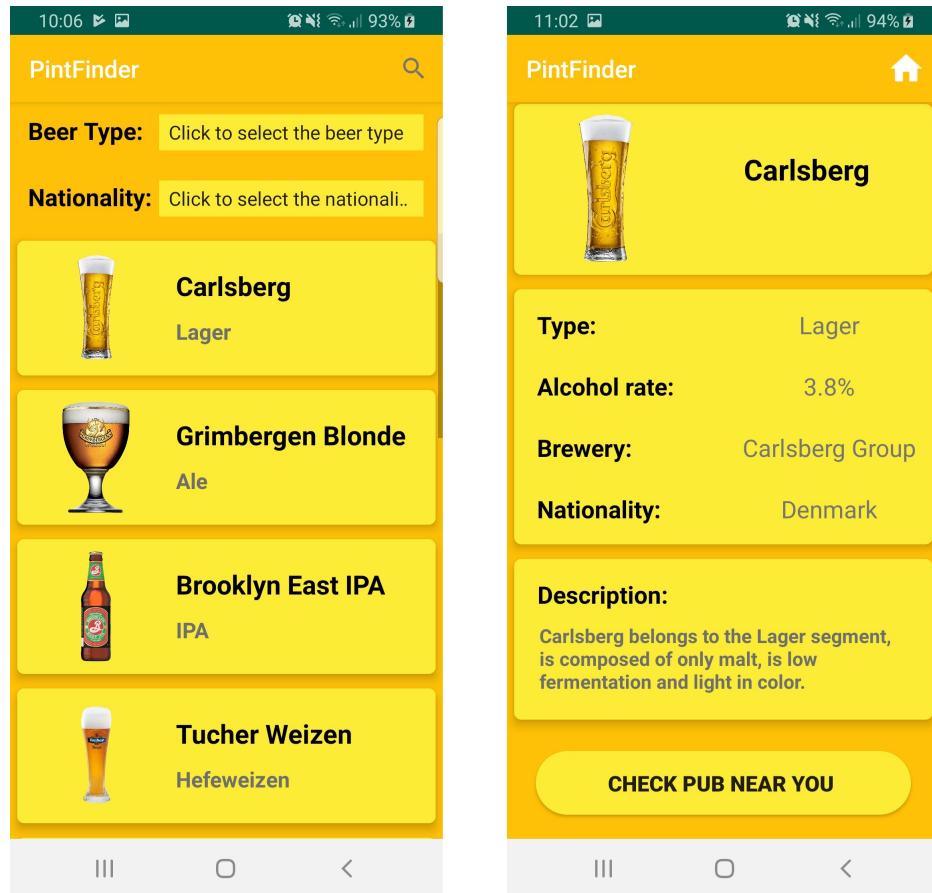


Figure 27: The advance search feature

- 3 - *Keep user informed about the profile he used for log-in and if logged or not.*  
-Severity : 3.

We solved adding the "class" of user in the drawer menù.

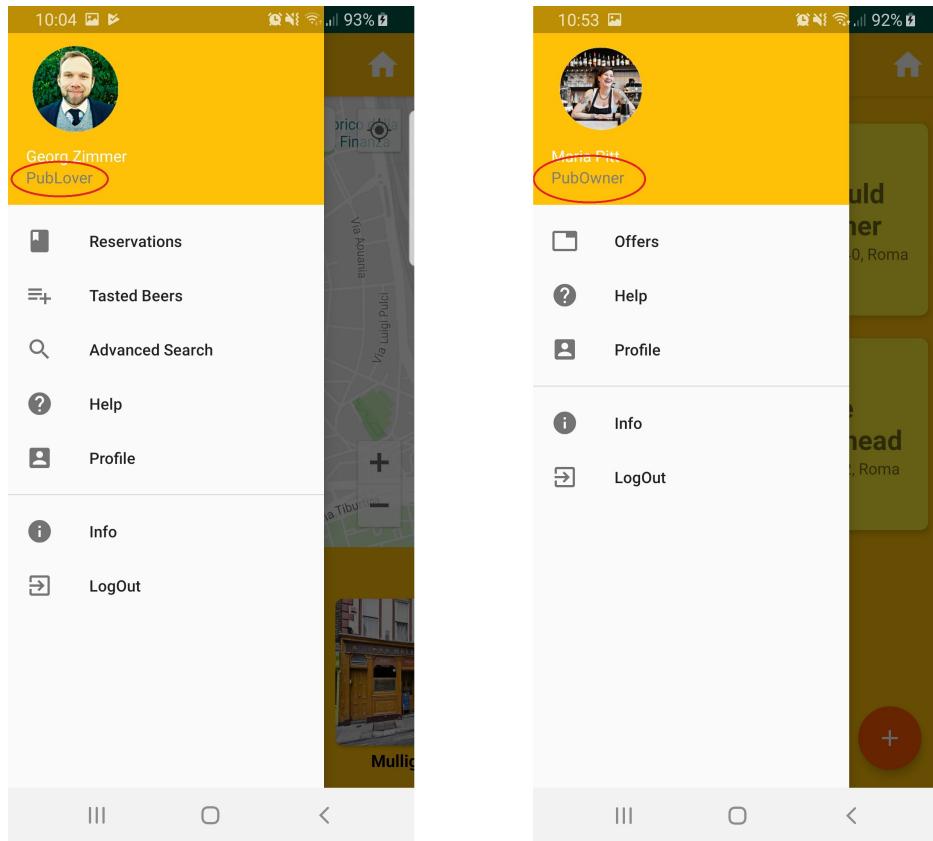


Figure 28: the drawer menù with the "PubLover" and "PubOwner" wording

- 4 - Add some details on last visited pubs adding for example the data, notes of rating left.

Severity : 2.

We solved adding a *note* area where a pubLover can write opinions, rates and so on.

- 5 - User could be interested in the availability of the tasted beer in the pubs.

Severity : 3.

We solved adding a "check pub near you" seen before.

- 6 - Supply search engine and accelerators.

Severity : 3.

We solved adding the search bars.

- 7 - Help user in remember if he already tasted the beer and give the information available in tasted beer section.

Severity : 3.

We solved adding a *tasted* wording to the side of a beer in a menù, if it is inside the tasted beers menù



Figure 29: If a PubLover sees a menu of a pub, he can check which beer is tasted and which not.

- 8 - Provide help and documentation.

Severity : 2.

We solved distinguishing the "Help" view from the "info" view.



Figure 30: the "Info" view

- 9 - *Usually personal information are not in help section. It will be not easily retrievable. Moreover help is missing.*  
-Severity : 3.

We solved making the "Help" view and the "Profile" view very distinct.

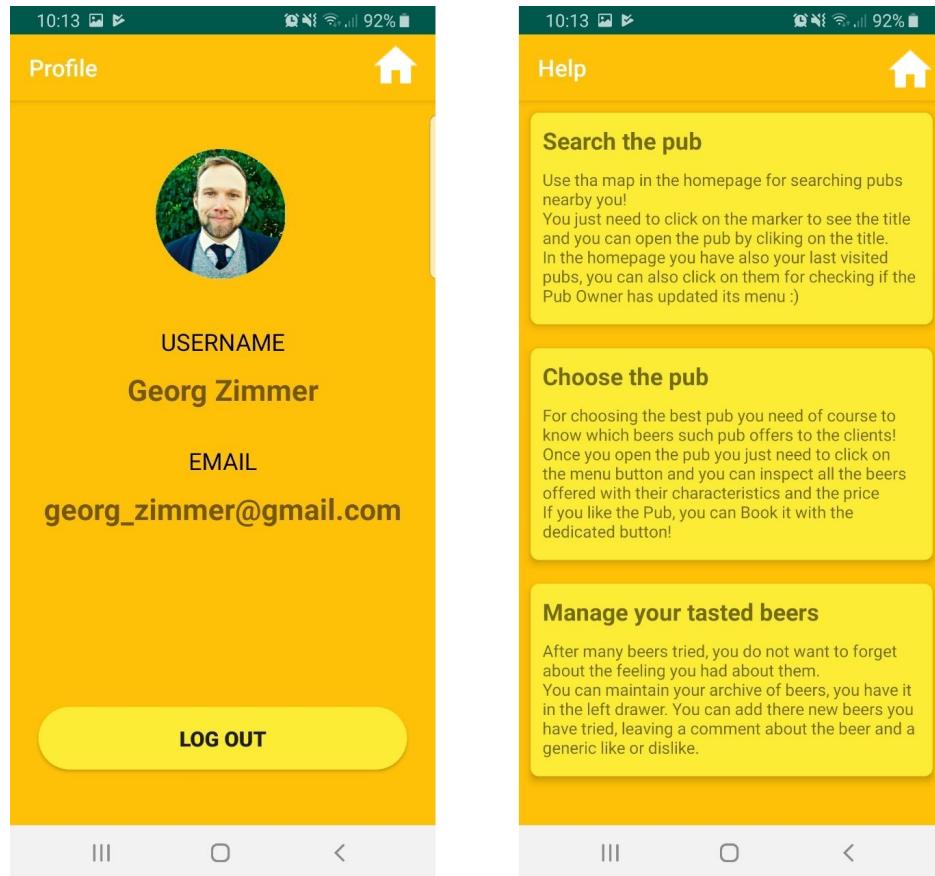


Figure 31: On the left the "Profile" view, on the right the "Help" view

## 10 Think Aloud

After explained to a group of 10 people who we are, what we are doing and what is about our application, we started a *Think Aloud session* with this parameters:

- The group was formed by people of different ages, to produce more interesting results.
- Each member of the group had to accomplish the same task individually.
- We explained the main task they had to accomplish: **Create a pub.**
- The experiment took place inside a room without distractions, and was executed one by one with a single smartphone where Pintfinder was installed.
- After one had finished the experiment, he had the duty to say nothing to the rest of the group.
- While the subject was executing the task, he had to say aloud all about what he was doing.
- During the execution of the whole experiment, we took notes with pen and paper.

We chose this task because from the questionnaire analysis we got that the mean age of pub owner is over 31, so we were worried that a non smartphone friendly user would have encountered problems in the very early stage of the app usage.

## 10.1 Conclusions of the think aloud session

What we obtained is that in general participants did not find any problem to accomplish the task they were asked. The interaction was smooth and they appreciated the responsiveness.

Only one main problem was revealed, one user created its pub without adding any beer in the menu. This was because he directly clicked in the "menu completed" button, because was unclear that the beers displayed must be clicked to be added.

We defined a solution for this problem and in the end we implemented it.

Priority of the incident	Description of the incident	How the incident was found	Potential solution
2	User created a Pub without adding any beer in the menu	User in the menu creation page directly clicked to "menu completed" because was unclear that the beers displayed must be clicked to be added	Add a textual panel in which you advise the user to click on the beer for adding it

Figure 32: Report of the user incident

# 11 Controlled Experiment

## 11.1 The Problem

We decided to make two different interfaces to accomplish the same task: delete a beer from *tasted beer* menu of a PubLover. The image on the left (Interface n1) is a screenshot from the first interface that we implemented and to complete the task you simply have to touch the red cross icon to delete the selected beer. Instead, in the Interface n2 we added a dedicated button with the description: *Delete from my tasted Beer*.

We decided to make a different interface because some of us thought that the last one was more immediate and direct than the first one. For this reason it has been decided to do a controlled experiment, to see which assumptions were right.

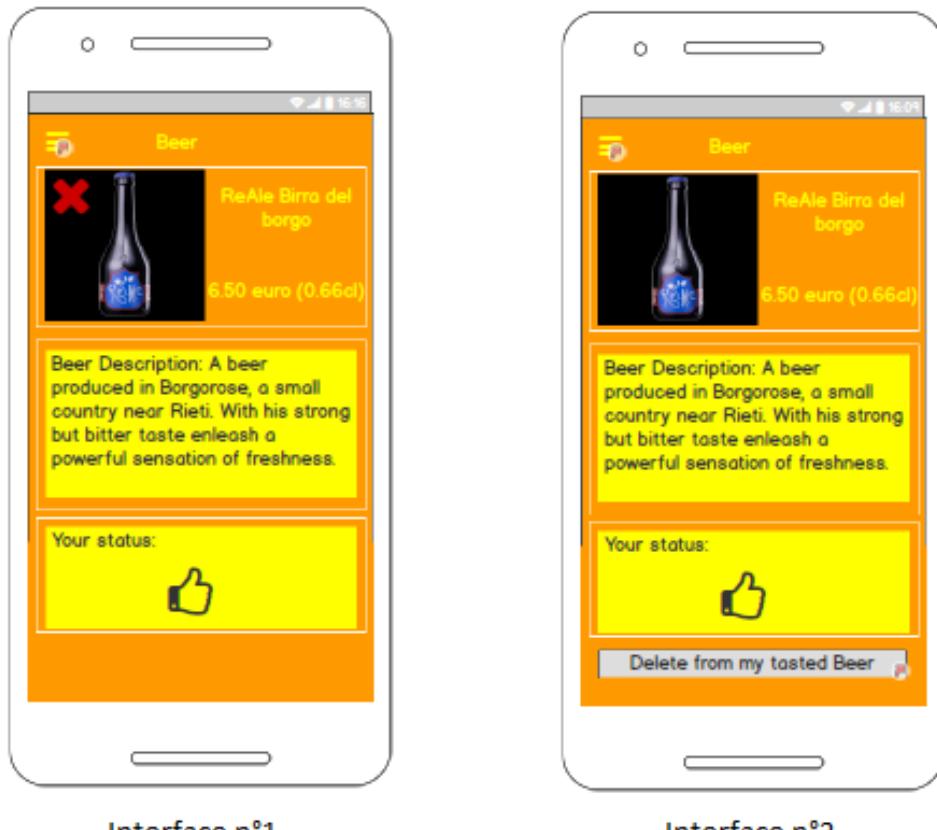


Figure 33: The two implementations of the deleting beer from tasted beer menu task. The left has the red "X" icon, the right has the dedicated button

To see if our hypothesis was true, we decided to use ANOVA as tool for a controlled experiment.

With a group a 10 people, we set the **variables**:

- Interface n1 and Interface n2 as **independent variables**.
- The time to execute the task, starting from the logging-in as **dependent variables**.

The Hypothesis were:

- Null Hypothesis: there are no differences between using the application with interface n1 or interface n2.
- Our Hypothesis: users will take less time to perform the task using the application with interface n2 than with interface n1

As said, the experiment starts with the logging-in of the user. All the users has to do the same task. After collecting the data, we applied the ANOVA One-Way Analysis.

## 11.2 The ANOVA results

Interface						
n1	n2					
14	11					
18	10					
12,5	7,5					
17	15					
20	15,5					
11	8,5					
10	6					
16,5	15					
11,5	9	Analisi varianza: ad un fattore				
13	8					
RIEPILOGO						
Gruppi	Conteggio	Somma	Media	Varianza		
Colonna 1	10	143,5	14,35	11,16944		
Colonna 2	10	105,5	10,55	11,96944		
ANALISI VARIANZA						
Origine della variazione	SQ	gdl	MQ	F	Valore di significatività	F crit
Tra gruppi	72,2	1	72,2	6,240576	0,022395611	4,413873
In gruppi	208,25	18	11,56944			
<b>Totale</b>	<b>280,45</b>	<b>19</b>				

Figure 34: ANOVA One-Way Analysis results

As we see from the Excel table our results are positive, because we know that if  $F > F_{crit}$ , we can reject the null hypothesis. So, in this case we reject the hypothesis that there are no differences between using the application with interface n1 or interface n2. Moreover, a lot of users found some trouble

finding the red cross because they thought that it was just a part of the image. So, in this case we reject the hypothesis that there is no difference between using the application with interface n1 or interface n2. Moreover, a lot of users found some trouble finding the red cross because they thought that it was just a part of the image. As conclusion, the null hypothesis is rejected. Moreover, the mean of the execution time of n1 is greater than n2, and as conclusion **our hypothesis is true**.

## 12 Final Product

Here there are some screenshots of the final App:

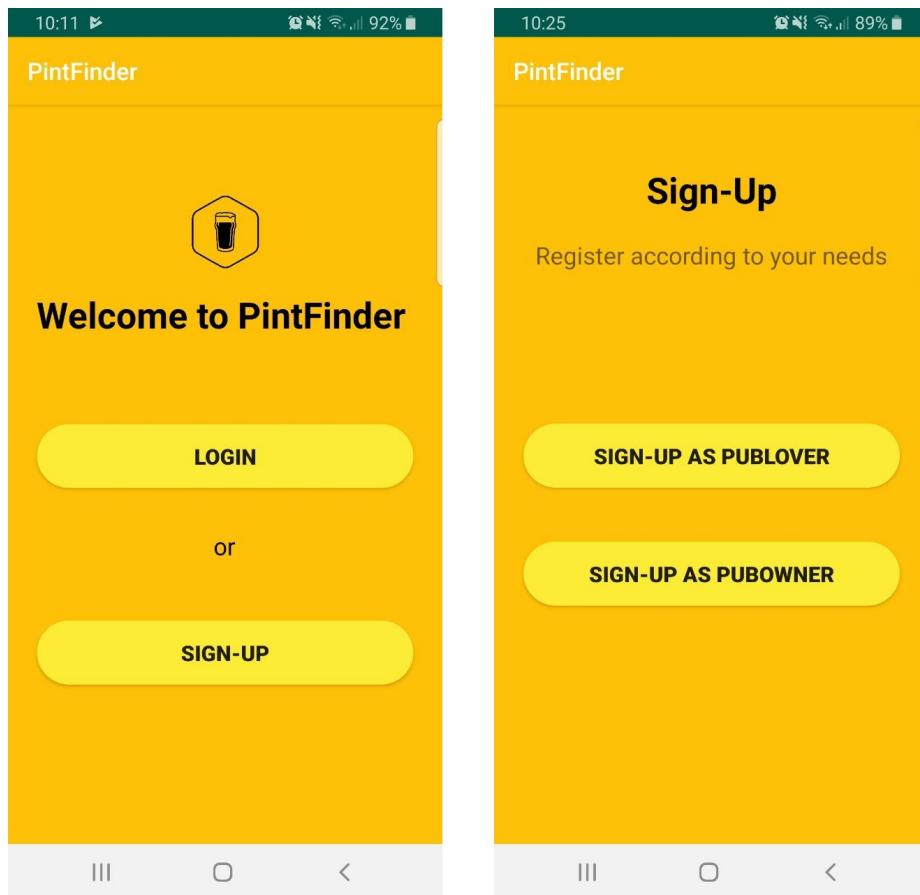


Figure 35: logging-in and signing-up

## 12.1 The PubLover

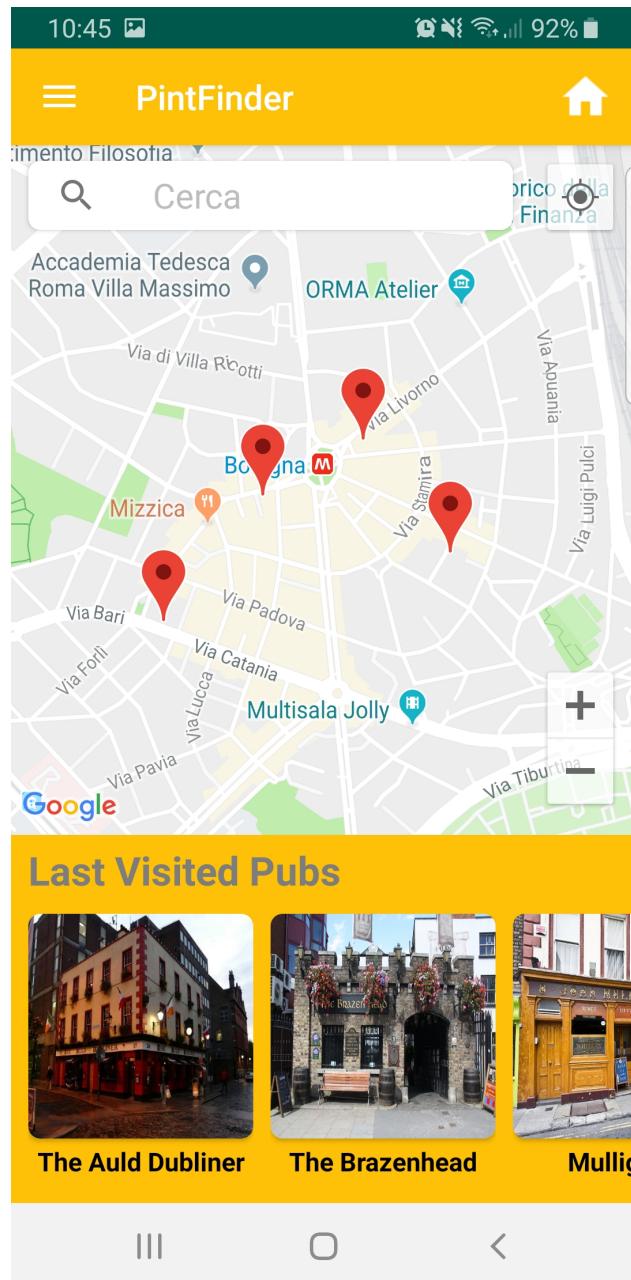


Figure 36: The **PubLover** homepage

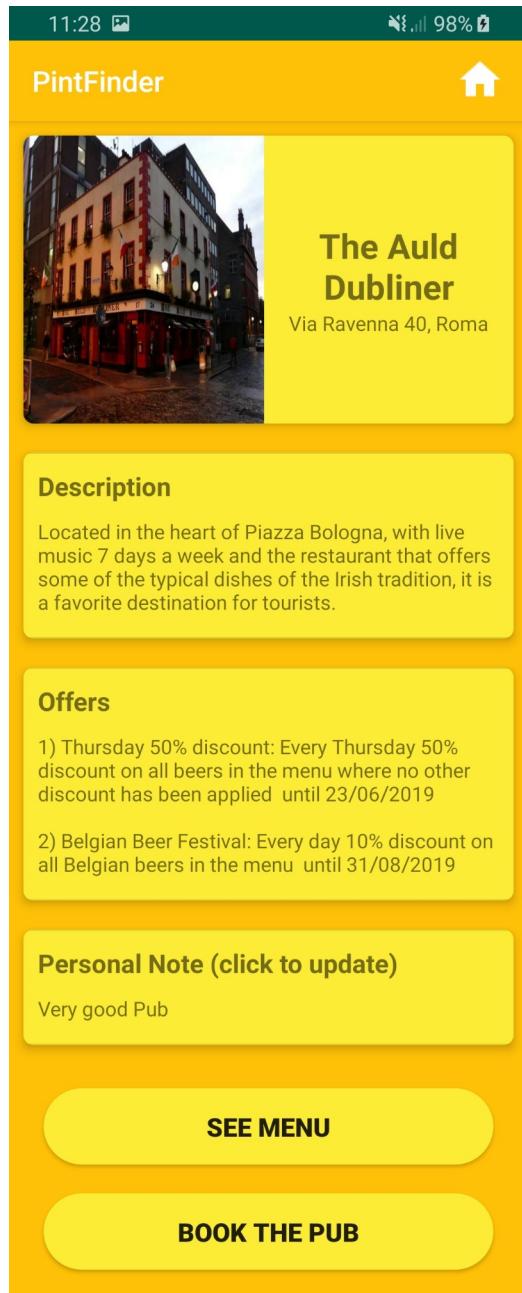


Figure 37: View of the pub from a **PubLover** user

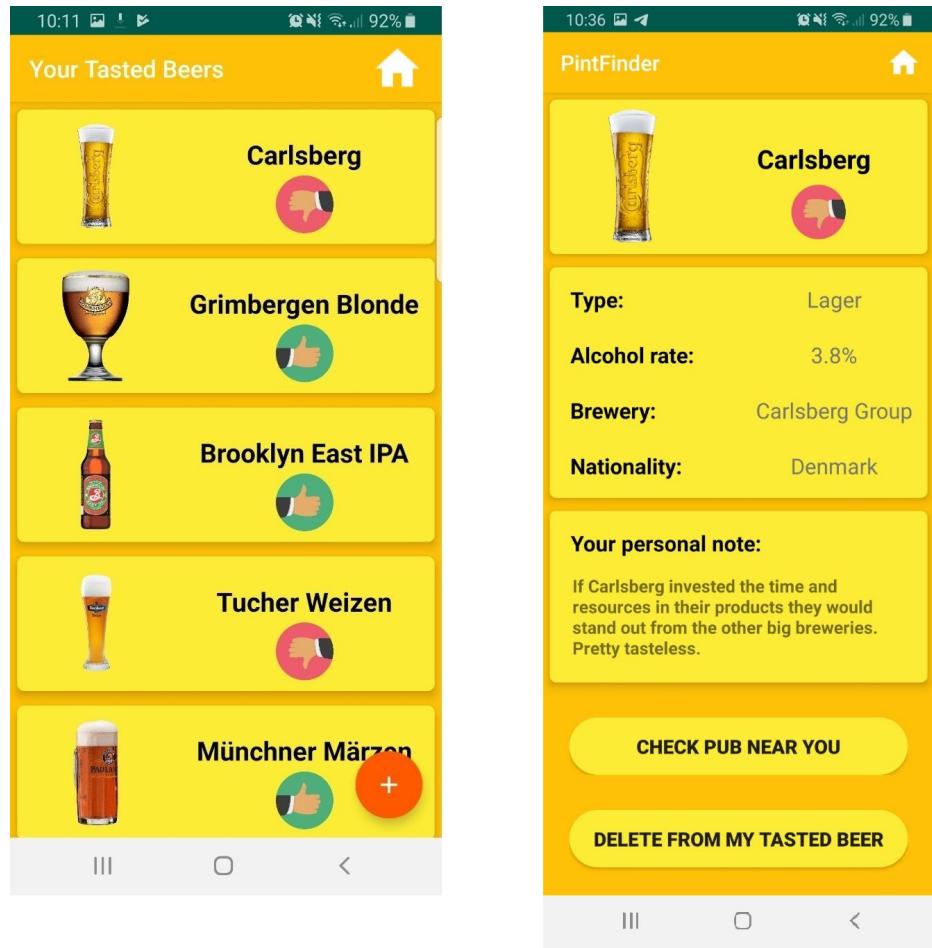


Figure 38: **PubLover** "tasted beer" feature

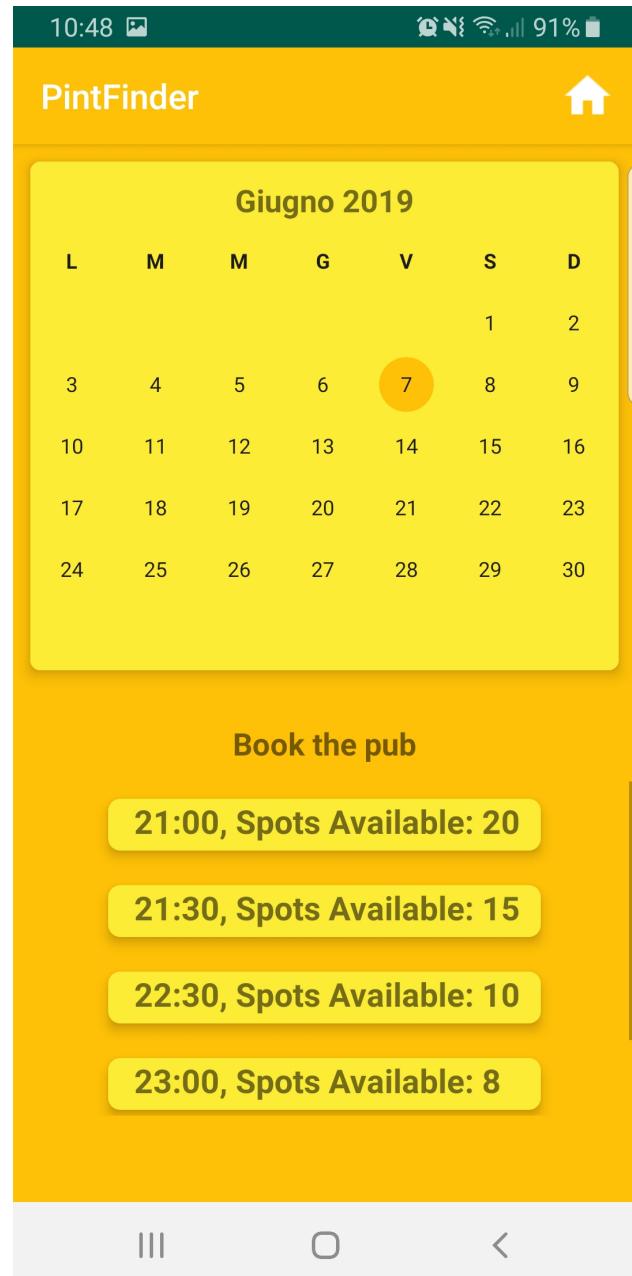


Figure 39: **PubLover** booking feature

## 12.2 The PubOwner

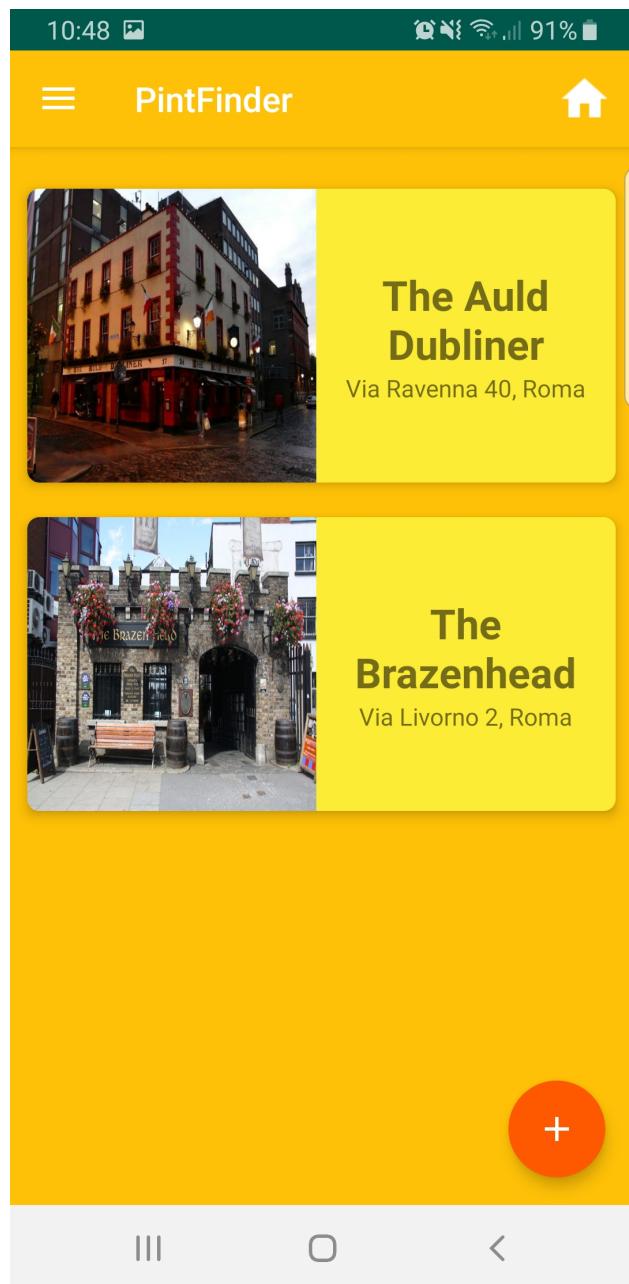


Figure 40: The **PubOwner** homepage

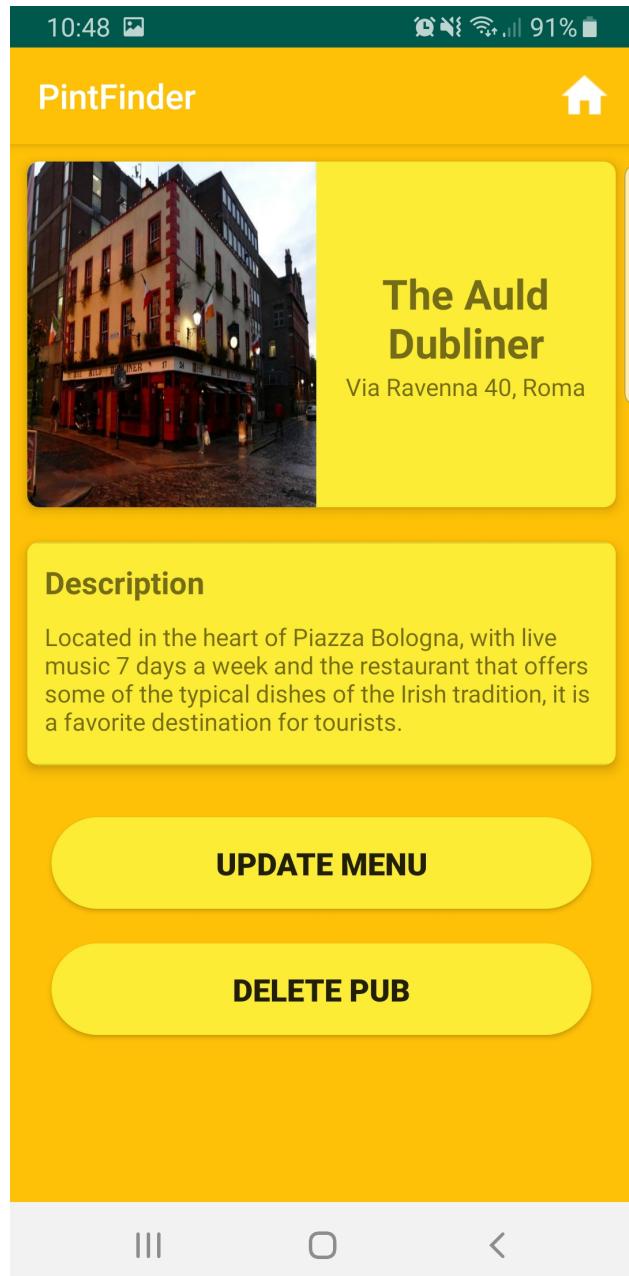


Figure 41: View of the pub from a **PubOwner** user

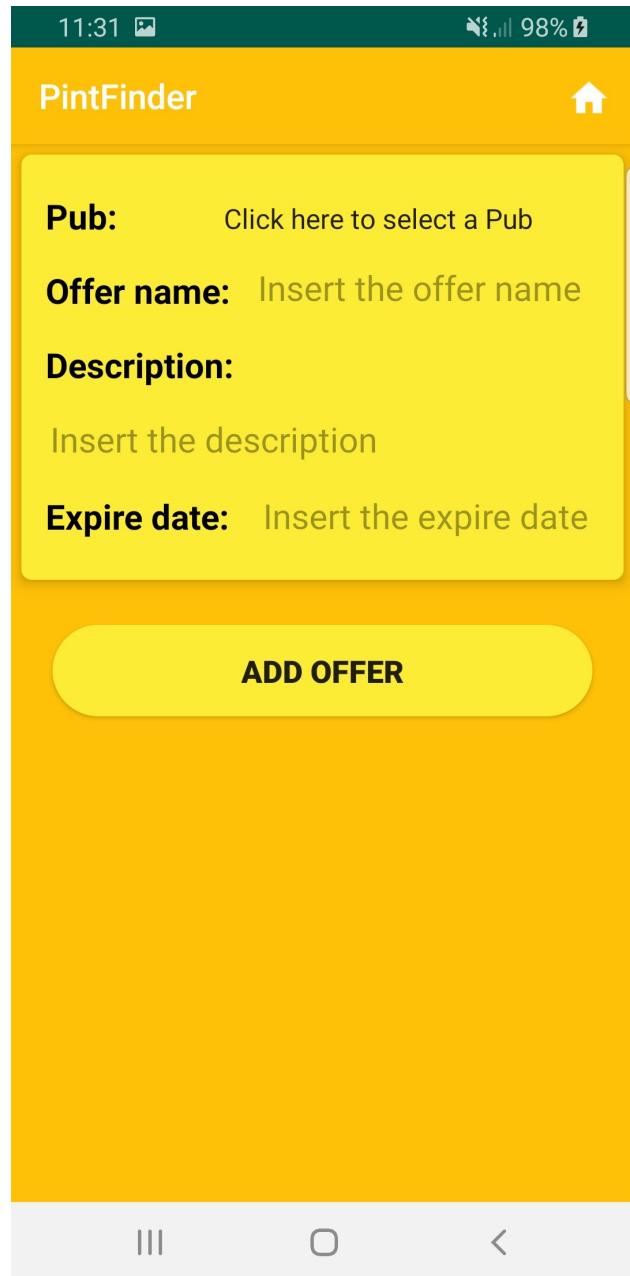


Figure 42: **PubOwner** offers feature

## 13 Conclusions

### 13.1 Future works

From this experience we learned that is not simple see an application as user. We started the idea from our point of view and modified the project through the questionnaire and improvements thanks to the expert-based evaluation and the think aloud experiment. Our next goals are to:

- Put the app to the Play Store.
- Implement the Admin persona.
- A iOS Systems porting.

### 13.2 References

- Human-Computer-Interaction course material.
- <http://www.excel-easy.com/examples/anova.html> for statistical analysis and Anova.
- Balsamiq mock-up: <https://balsamiq.com/products/mockups>.