# SEProject: Report - Requirements, methodology and planning

Groupe C


| | | |
|---|---|---|
| Mokaddem Sami | Heymans Pierre | Francotte Florian |
| 7408-12-00 | 3154-12-00 | 8327-14-00 |
| Rollin Antoine | Marin Alicia | Murhububa Serge |
| 1346-15-00 | 1149-15-00 | 0064-14-00 |


September 25, 2015

# Introduction

In the course LINGI2255, we have been asked to develop a website concerning tennis tournaments. Particularly, this website include players and owners registration and staff management helped with useful algorithms.

In this report, we will talk about our software development method, the analysis of requirements, the initial planning that might change during the weeks and finally, how the work was split.

# Software development method

In this project, we are free to select our development method. As this project must be done in parallel with other course projects, we are going to use an **iterative development method**. Particularly, after some discussion concerning the kind of methods we could use, we selected the **agile method** for several reasons.

In the first place, we are a small group of developers. So it is easier to attribute some small tasks to a member or a pair.

Secondly, we are not very familiar with the web development. So, the project will certainly evolve during its development which involve a method that can easily respond to change: agile method.

Finally, we all used to use this method in our previous course and it works smoothly. So, it will be easier for us to perform the different steps of the iteration.

In this project, we will use several tools. Obviously, we will use a version control system: git, hosted on GitHub. To have a good view of our project progression, we will use a project management application: Trello. It will be useful to assign task to member, discuss about some point of the user stories and so on.

# Analysis of requirements

We know that the analysis of requirements is a very difficult and risky part of the job. So, we will try our best to put them in evidence.

## Use cases

### Player registration

- Players enter their information (name, address, phone, ... + payment method and extras).
- System verifies that all the information is correct with a basic check.
- System attributes an identifier to the pair and the players.
- System sends a confirmation email to the players.

### Staff user account

- The staff can view all the players and the pairs sorted as they want.
- The staff can view all the information about a specific player and pair.
- The staff can perform a search with the identification number or names.
- A member of the staff can be responsible for a specific category. His information is present on all the documents sent to the players.

**Courts registration**

Owners

- Owners can register their courts in the same way that the players.
- System checks the validity of the information.
- The system compare the information with these stored in the database. If there are not present, there are inserted. Otherwise, there are updated.
- System sends a verification email to the owners.

Staff

- They can edit, create or suppress any courts
- They can view all the information about a specific court.
- They can perform a search with the courts identification or owner.
- The staff can send multiple emails to check if the courts' owners are still available.

**Group creation**

- Staff can manually create a group through a friendly interface.
- System checks the validity of the information.
- System sends emails to the player with the court location, the schedule and the information of the HQ address.
- The staff can encode the results of each matches in the DB.

**Knock-off tournament**

- Staff can add a courts and players.
- Staff can print the table with the address and contact information.
- Staff can encode the result.
- A specific page can display the result of the day.

## Use cases diagram

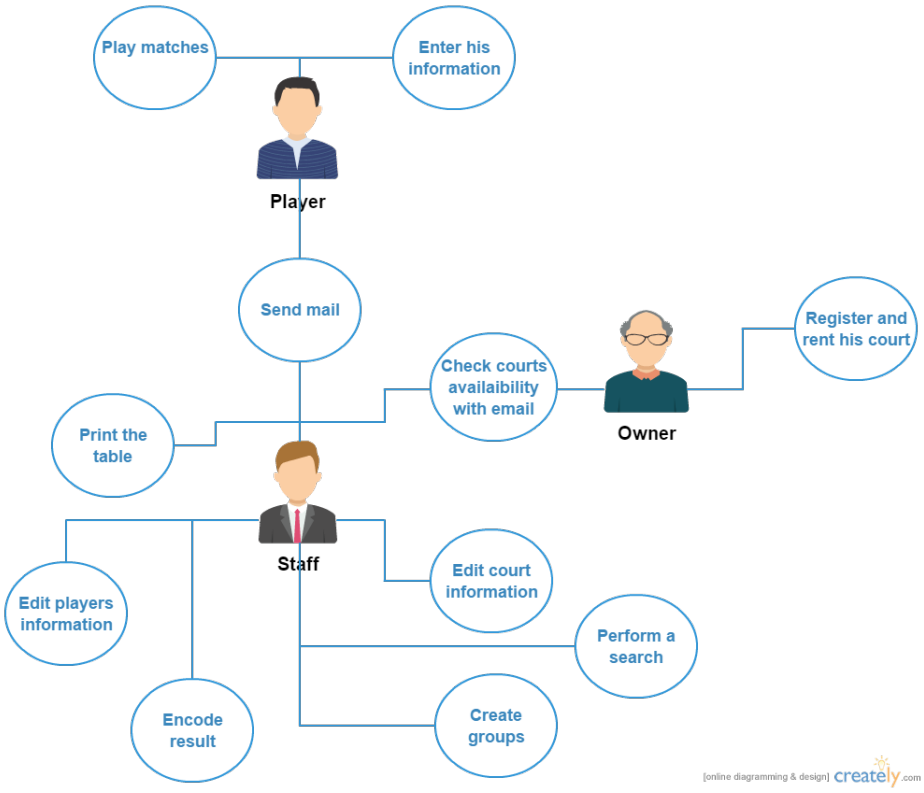


Figure 1: The Use case of players registration, courts registration and group creation.

## Activity diagram

Below, the activity diagram for a player.


Figure 2: Activity diagram for a player

## Activity diagram

Below, the activity diagram for a staff member.

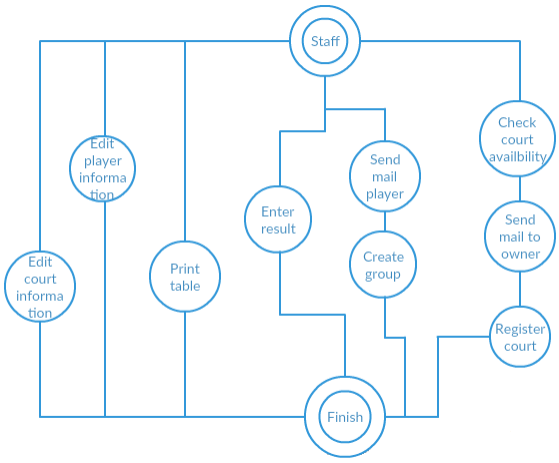


Figure 3: Activity diagram for a staff member

# User stories

**As a player, I want**

- [1] to subscribe to the tournament by entering my information in order to take part to the tournament
- [2] to receive a confirmation email to be sure that I'm part of the tournament
- [3] to send an email to the staff if I have any problems

**As a owner, I want**

- [4] to register my courts so that players can come and play on it
- [5] to receive a confirmation email that my court will be used, in order to prepare it accordingly

**As a staff member, I want**

- [6] the system to check the information filled by the players and the owners to make sure that their registrations are in order
- [7] a confirmation email to be automatically send to the players and owners
- [8] to view the players, pairs or courts sorted as I want
- [9] to be able to view every single information about a player, a pair, or a court
- [10] to perform a search on the names (even partial) or ID of the players and courts
- [11] to be responsible of a category, so that the players and owners can contact me
- [12] the system to compare the information of a registered court with those in the database, in order to insert the new court or update the old one
- [13] to edit, create or suppress any court or player
- [14] to send multiple emails to the owners to check if the courts are still available
- [15] to create a group for the morning tournaments
- [16] the system to verify that there is no mistake in the group I create manually

- [17] emails to be send to the players to tell them where they have to play during the group matches
- [18] emails to be send to players that are not in order of payment to tell them where the HQ is so they can come and pay
- [19] to encode the results of any matches to the corresponding table
- [20] to link players and court for the Knock-off tournament
- [21] to print the table with the address and contact information of the courts where players will be playing on
- [22] a specific page to display the results of the tournament

### Non-functional requirement

Our website will be designed to provide these features:

Must have functionality

- A brief documentation for the staff member on "*how to use this website*".
- The website must be responsive and up-to-date when the database is used. Thus, the player registration and the staff management do not take ages and are reliable.
- The website must be a minimum secure, so that only the staff can edit all the information and not a random user.

Optional functionality

- The website must be accessible and compatible from a lot of platform. For example, the most popular browser and smartphones.
- The website must be as user friendly as possible, so that the staff member (which is not an experimented user) can edit and manage information easily.

## Optional functionality

Below, the optional functionality. This is the features we would like to implement to the website if we do not lack of time. They all will be done after the "must have" features.

- [23] The system will keep an history of every modification done by the staff.
- [24] Players will stay in the database from one year to another.
- [25] A verification email will be send to users to make sure that the email is correct.
- [26] The Knock-off table will be generated automatically, but will have to be approved by a staff member.
- [27] It will be possible to do payments on-line.
- [28] Players will be able to register individually and play with an unknown person. The system will ask the players to pay once a match is found.
- [29] Accounts will be used for the players and the owners to see their evolution and ease their registration year after year. Owners will know who played on their court.

## Initial planning

As we are using the agile development method there is not a very precise schedule that we have to follow. Thus, the figure 4 is just an approximation of what we expect to achieve during the development period.

Figure 4: Our planning approximation using the Gantt diagram method.

## The split of the work

As we just have met each other, we have done this report together over two meetings. Obviously, we will split the work in the future.

## Conclusion

Definition of requirements is useful phase in software engineering project, it requires to be well done and represent all of the customers need. In this project we have used some methods to define requirements to provide usability and reliability of the project. Organisation of work is focused on tasks that are represented in the schedule project and a test will be done at the end of each phase project in the weekly meeting to validate if the work is achieved and well done by all members of the team.