

Setup document

TEAM META

Formal client

Client: Mr. Stevens

The team

Team leader

Name: Veselin Slavchev

Student number: 2812630

E-mail: v.slavchev@student.fontys.nl

Secretary

Name: Yassen Georgiev

Student number: 2837465

E-mail: y.georgiev@student.fontys.nl

Quality manager

Name: Ivan Stoilchev

Student number: 2833360

E-mail: i.stoilchev@student.fontys.nl

Chairman

Name: Ivailo Hristov

Student number:

E-mail: i.hristov@student.fontys.nl

Table of contents:

1. Preface
2. Client
3. Processes
4. Functional requirements
5. User interface
6. Database design

1. Preface

This document will introduce you to our project setup: what is the software supposed to do, what are the requirements and what will the interface look like. The second chapter will tell you more about the client and their needs. The Processes chapter explains the expected use of the software and some exceptional situations. The Functional requirements specifies which features we are going to implement and which not. The User interface chapter lays out images of our designs.

2. Client

The client is Mr. Stevens. He has an upcoming event and needs software to ease the registration of users and the event management by the staff. Our team is responsible for the design and development of:

1. a website for information and registration of attendees of the event;
2. a database design;
3. event entrance application;
4. camp entrance application;
5. shop application;
6. loan materials stand application;
7. event exit application;
8. status inspection application;
9. application responsible for the PayPal log files.

3. Processes

We will give an example of how the process will follow through for a group of 5 friends. Initially one of them finds the website and decides to tell 3 of his friends (the 5th friend will join them later). They agree and tell him to make a reservation and he will be responsible for the organization of the group. This person registers himself, supplies basic information about his 3 friends (names and emails) and selects the camp spot for the group. He then pays for: his entrance fee and the camp spot ($55 + 30 + 3 \times 20$). One of the 3 friends goes to the website and supplies more information: their paypal account, and is therefore able to put money into their account and pay all the fees: 55 euro. The second of the 3 friends will also go to the website and supply money (30 euro), but it won't be enough (25 more is needed). The last of the guest friends will not do anything. We said that the organizer paid 30 euros for the camp; if he/she didn't pay that much they would not have a camp spot reservation, that is they may not be allowed to go to their camp spot as some other visitors have actually paid for it in the worst case; best case: no one has paid for the camp spot, it is unoccupied and the group can take it.

As they get ready to go to the event a 5th person decides to join his 4 other friends. They were never registered with the group and no information about them was ever given. At the entrance the organizer of the group and the person who paid all his/hers fees are allowed to enter. The attendee who paid partly is allowed to pay,

but has to pay an extra 10 euro (for not paying on time). The person who didn't supply anything and 5th person, who joined last, will have to give information about them and will be allowed to pay, also including the extra 10 euro each. The 5th person will have to pay extra 20 euro if they want to also have a slot in the camp spot. While the 5 friends are at the entrance an old friend of theirs recognizes them and decides to join (this person has not supplied any information). This 6th friend will not want to have a camp spot, so they will only pay 55 + 10 to enter. The 6th spot in the camp will remain unclaimed, and will not be paid for.

After getting through the entrance everybody in the group has supplied their personal information, has put some money in their account and has a QR code.

4. Functional requirements

Event entrance application

To be used to check whether a person is allowed to enter, and to enable supplying more information about an attendee who did not give enough details.

Must have:

1. check attendee status;
2. input more information about an attendee;

Should have:

Could have:

Would have:

Camp entrance application

To be used to check whether a user is assigned to a slot in a camp spot and which one it is.

Must have:

1. check attendee status;
2. tell attendee which is their camp spot;

Should have:

Could have:

Would have:

Shop application

To be used to write information about the purchases of products and also subtract money from the user's account.

Must have:

1. a list of available products;
2. cancel the purchase;
3. confirm payment;

Should have:

Could have:

Would have:

Loan materials stand application

To be used to show a list of available materials to loan.

Must have:

3. check attendee status;
4. write in db that a specific material is assigned to a certain person;
5. cancel loaning;
6. return a material to the stand inventory;

Should have:

Could have:

Would have:

Event exit application

To be used to check whether a leaving attendee has returned all the loaned materials and they are allowed to leave. Then the account should be marked as invalid.

Must have:

1. check attendee status;
2. mark account as invalid;

Should have:

Could have:

Would have:

Status inspection application

To be used to show information about the status of a user.

Must have:

1. check attendee status;

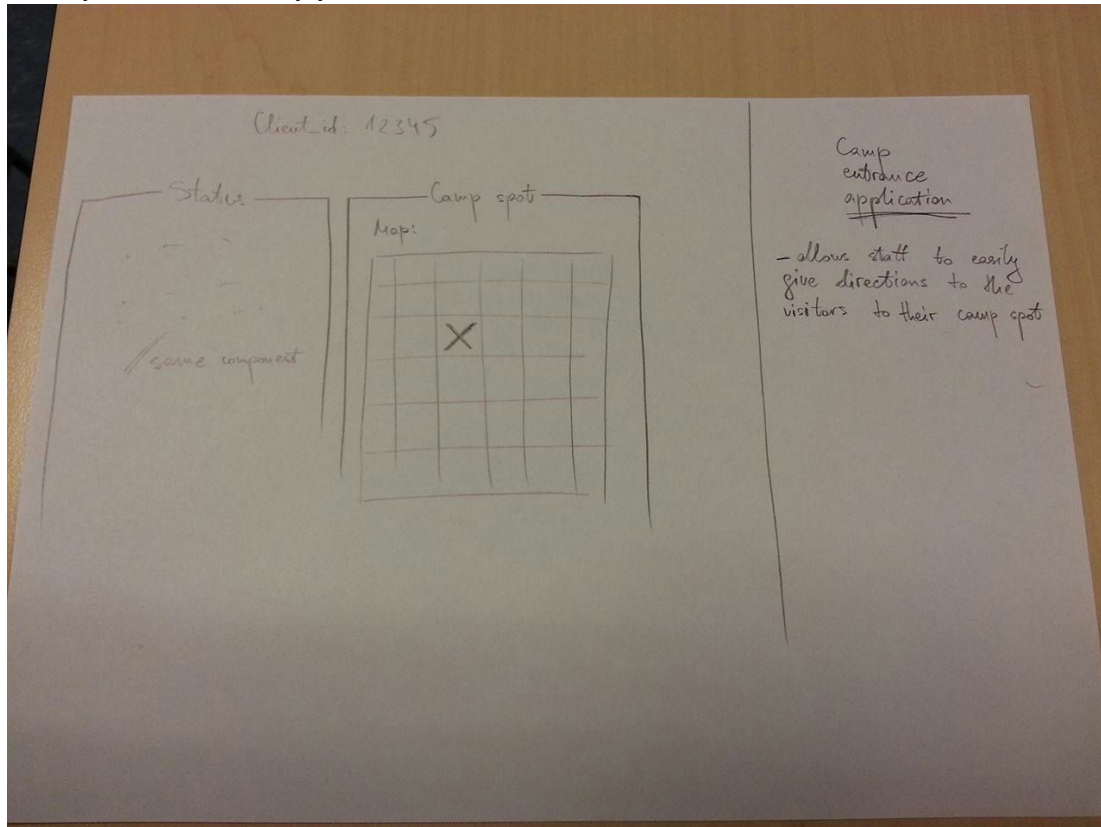
Should have:

Could have:

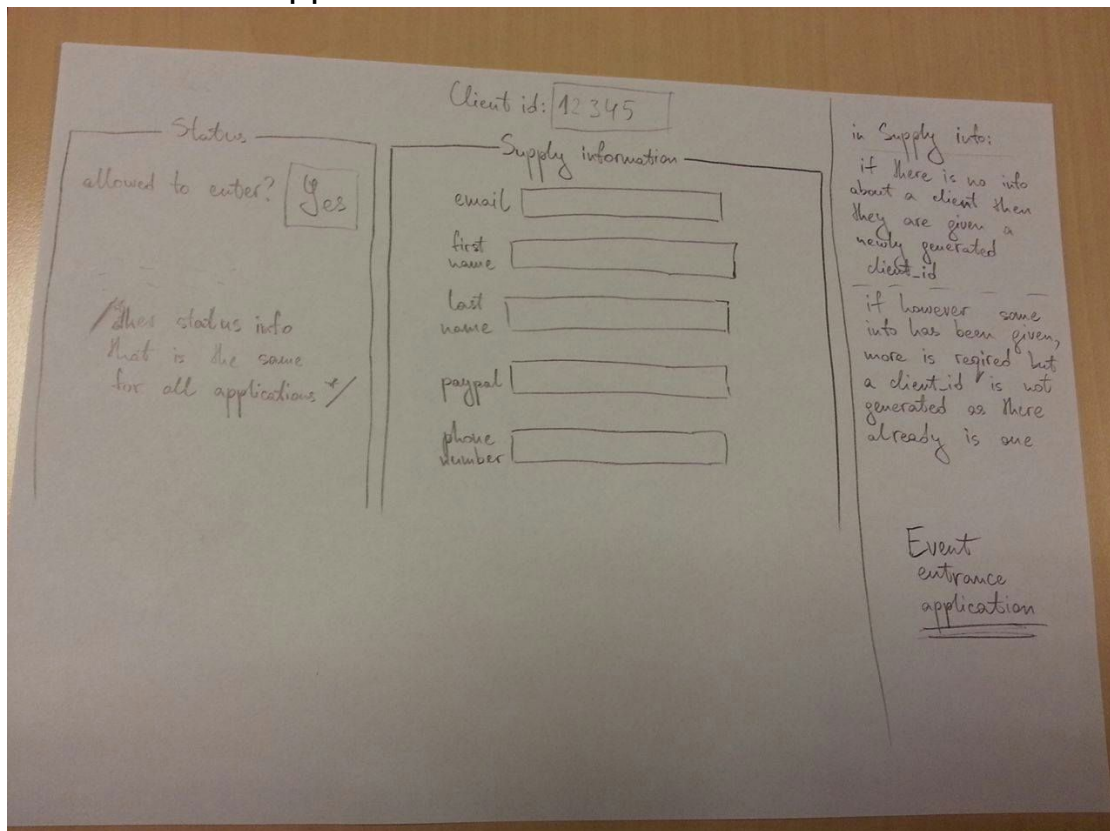
Would have:

5. User interface

Camp entrance application



Event entrance application



Status inspection application

Status inspection application

Visitor status:

by id:

by last name:

result:

email: kiro@kek.tr

money: 53.40

entered: yes

~~orders history~~: ?

money spent on food: 12.30

total money transferred: 67.20

Overall status:

visitors entered: 255

visitors not yet entered: 65

visitors left: 15

total money balance: 30 450.35

total money paid: 80 270.40


camp spots booked: 37

list of free camp spots:

...
...
...
...
...

Loaning materials application

Loaning materials application

Client id: 12345 

Materials to loan:

material name	material_id	time	price
Laptop	56783	16:57	5.00
Charger	56001	16:57	3.00


Total: 8.00


Materials in stock:

name	quantity left	loan
Laptops	7	+
Charger cables	15	+
Keyboards	10	+

Return material:

id:

 textField

 button

Materials in stock are all unique by id. A concrete material should be connected to the person who loaned it in case it is damaged.

Shop application

Client id: 12345

Products:

#	id	Name	Price	Qty
1	13	Potato	5.50	2
2	17	Coke	7.20	1
3				
...				
32				

Total: 45.70

Shop application

Potato ☒ → dropdown menu to choose a product

options for the menu:

—
Potato
Coke
...
...

→ /empty

Exit event application

Client id: 12345

Loaned materials to return:

material name	material id	return
Laptop	56783	<input type="button" value="return"/>

Money balance: 21.50

← button

Exit application

6. Database design

