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](mailto:v.slavchev@student.fontys.nl)

* Secretary

Name: Yasen Georgiev

Student number: 2837465

E-mail: [y.georgiev@student.fontys.nl](mailto:y.georgiev@student.fontys.nl)

* Quality manager

Name: Ivan Stoilchev

Student number: 2833360

E-mail: [i.stoilchev@student.fontys.nl](mailto:i.stoilchev@student.fontys.nl)

* Chairman

Name: Ivailo Hristov

Student number:

E-mail : [i.hristov@student.fontys.nl](mailto:ivailo.hristov@student.fontys.nl)

# Table of contents:

1. Preface

2. Client

3. Processes

4. Functional requirements

5. User interface

# 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 and example of how the process will follow trough 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 (such as names and emails) and selects the camp spot for the group. He then pays for: his entrance fee and the camp spot (55 + 30 + 3x20). 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 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.

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 the fees are allowed to enter. The attendee who paid partly is allowed to pay, but has to pay an extra 10 euro each (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, but 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.

After getting through the entrance everybody in the group has supplied their personal information and 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:

1. check attendee status;
2. write in db that a specific material is assigned to a certain person;
3. cancel loaning;
4. 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;

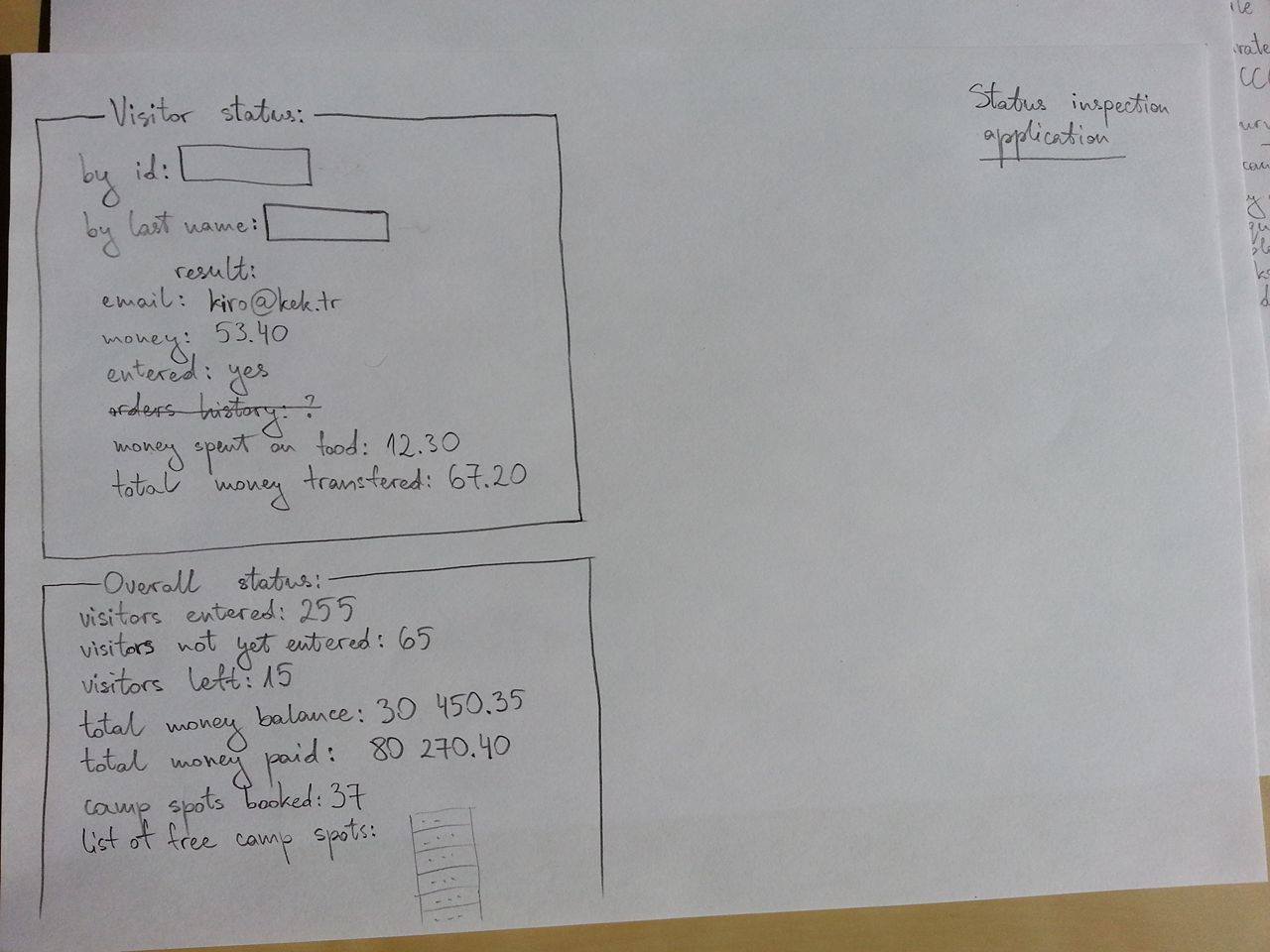
Should have:

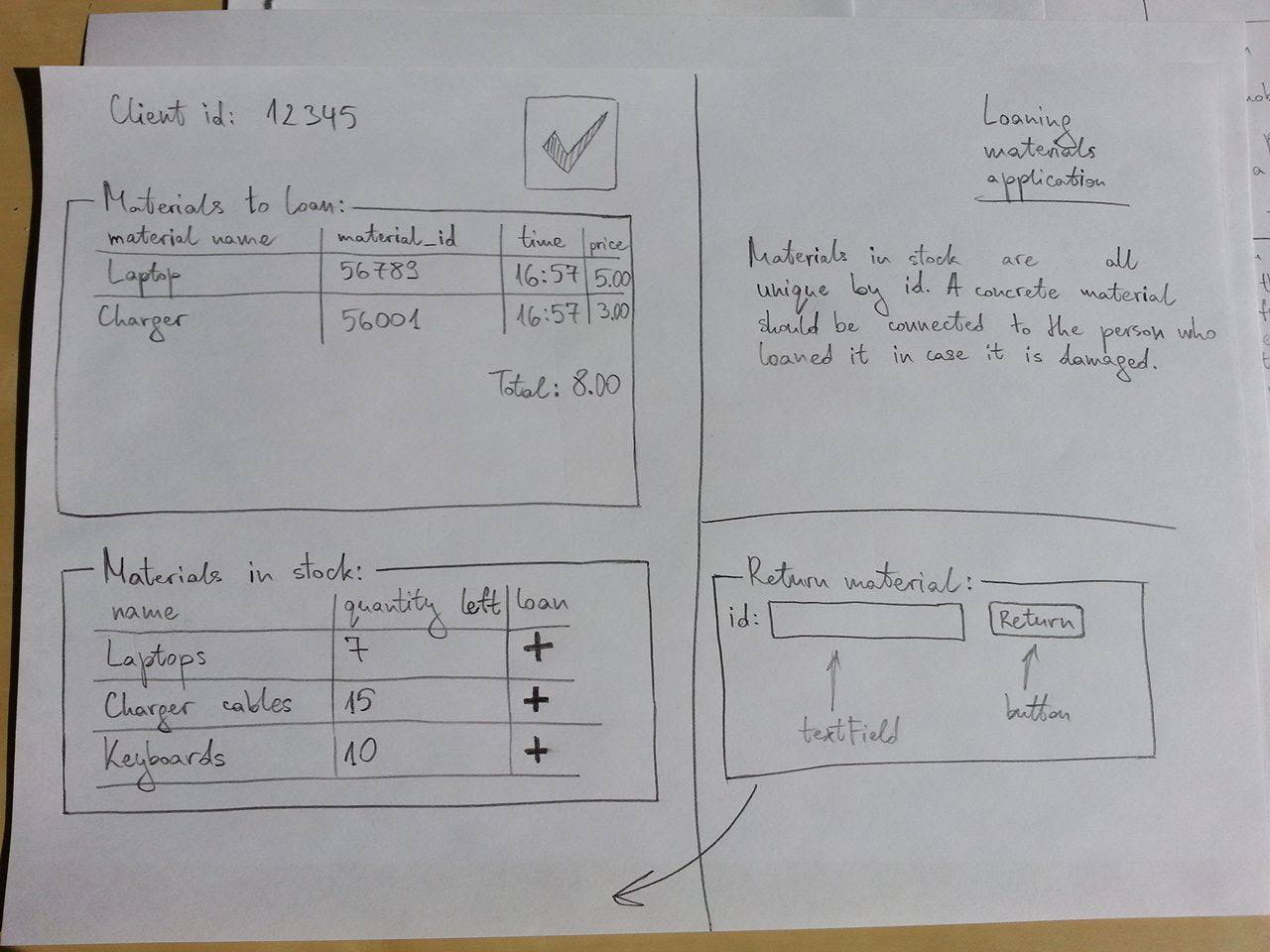
Could have:

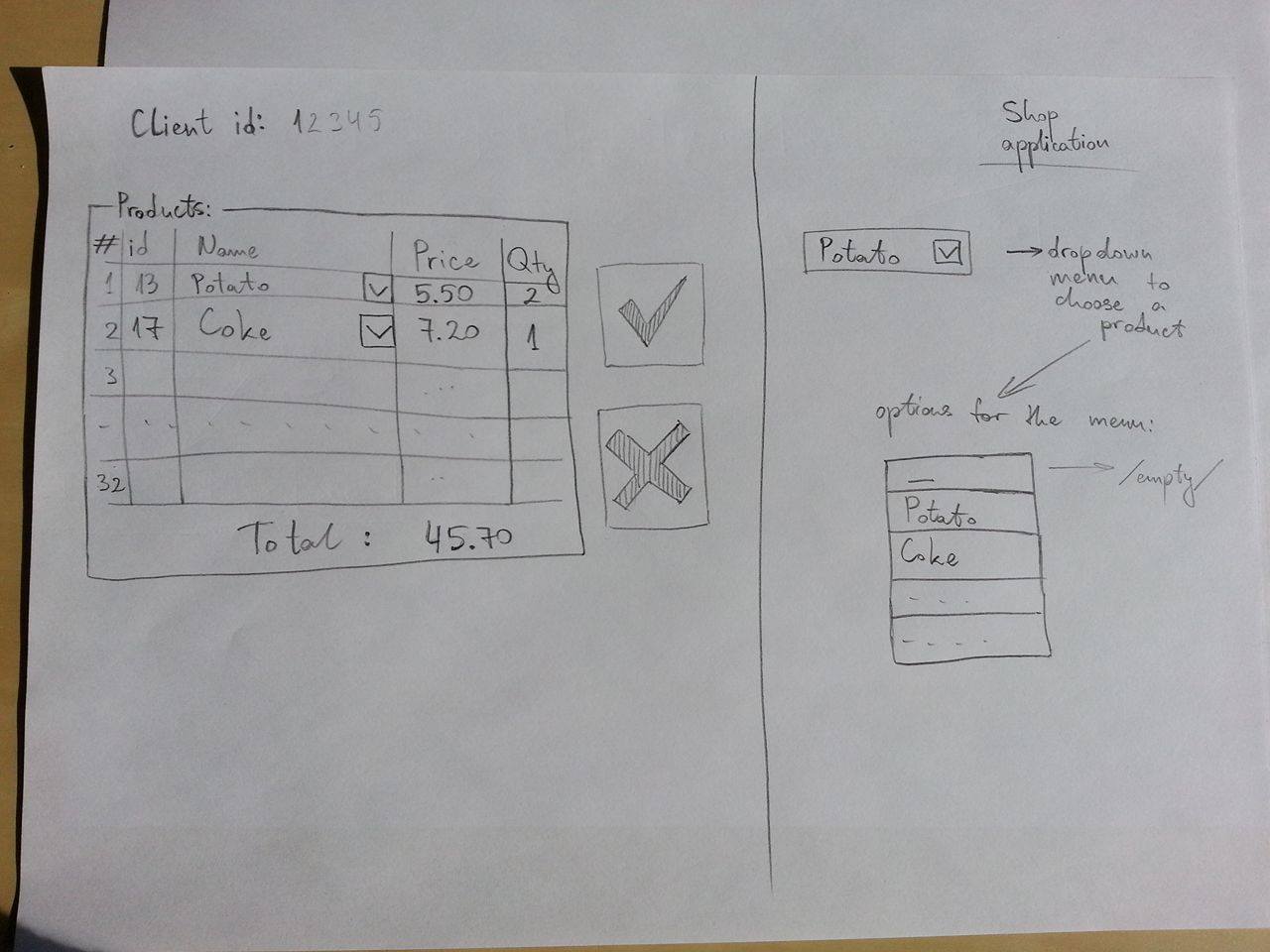
Would have:

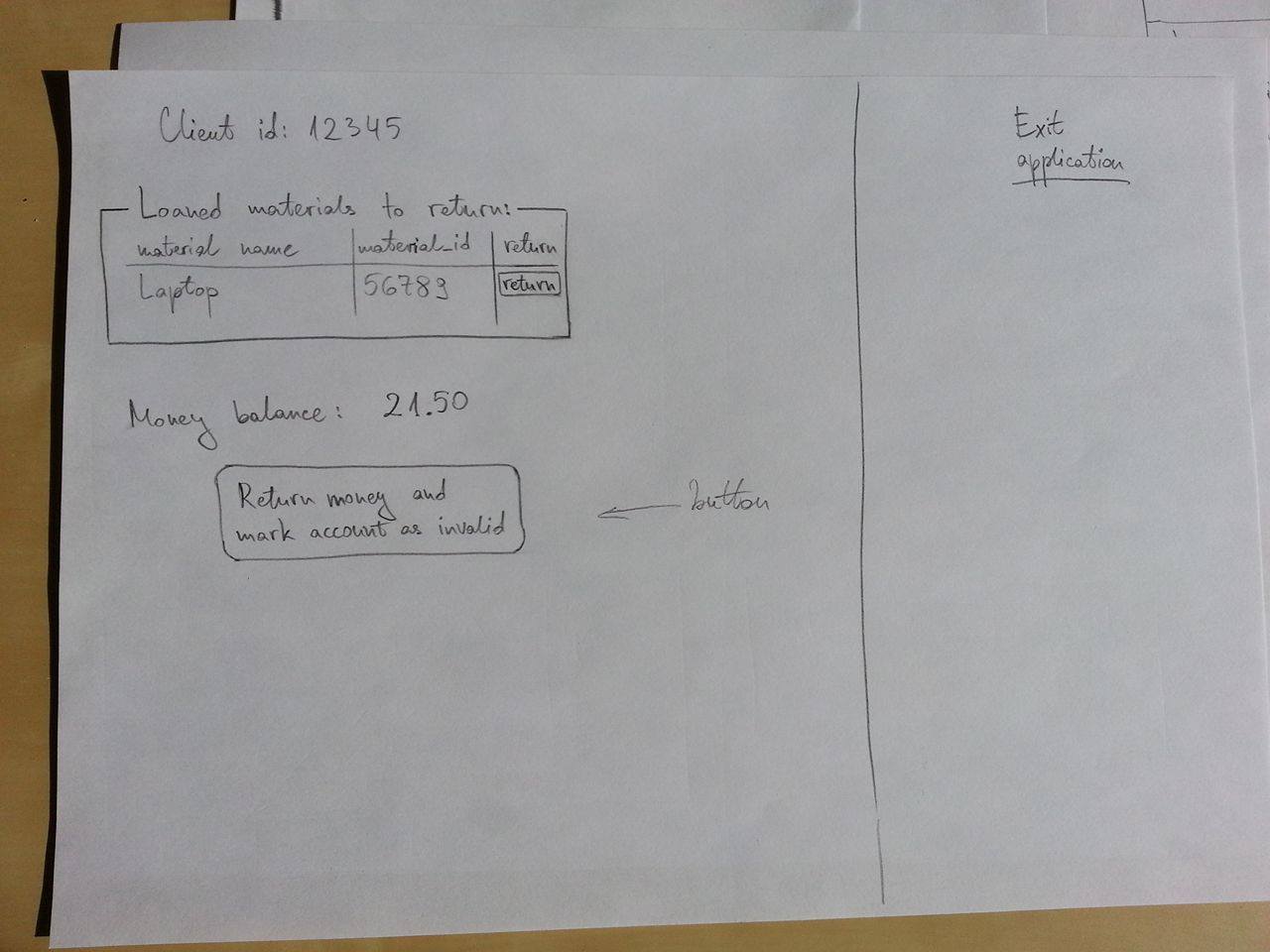
# 5. User interface

Status inspection application



Loaning materials application

Shop application

Exit event application