Requirements Specification for the PROJECT of course TDDC32

Laurent POUYET

Rémi GUILLARD

remgu983@student.liu.se

laupo171@student.liu.se

[TDDC32] – [Project] Laundry booking system

Group A6

|  |  |  |
| --- | --- | --- |
| Revision | Date | Comment |
| 1.0 | 2012/02/16 | First version of subject. |
| 1.1 | 2012/02/20 | Add of the module responsibilities |
| 1.2 | 2012/02/21 | Add of the requirements section. |
| 1.3 | 2012/02/24 | Modification of the document structure, addition of use-cases, summary, document conventions, non-functional requirements and optional requirements. |
| 1.4 | 2012/03/25 | Update of Work repartition because of architecture modification |
|  |  |  |
|  |  |  |
|  |  |  |

# Summary

This document consists in the requirements specification for the project “Laundry Booking System” in the context of the Project part for the Course [TDDC32] – “Design and implementation of a software module in Java”.

This project is a booking system for a laundry; it will consist in 2 parts: a client and a server. The server will handle a database and answer to the client requests. The client will handle the booking part and the calendar display. The user will be able to book a machine with the help of the GUI client, and then manage and administrate all his booking, when he is authenticated.

Contenu

[Summary 2](#_Toc317872535)

[Introduction 4](#_Toc317872536)

[Project Description 5](#_Toc317872537)

[Description of subject 5](#_Toc317872538)

[The server 5](#_Toc317872539)

[The client 5](#_Toc317872540)

[Task repartition 6](#_Toc317872541)

[Document Conventions 7](#_Toc317872542)

[System Description 8](#_Toc317872543)

[User Interface 8](#_Toc317872544)

[System Functions 9](#_Toc317872545)

[Non-functional Requirements 9](#_Toc317872546)

[Storage of permanent data 10](#_Toc317872547)

[Use-cases of the System 11](#_Toc317872548)

# Introduction

The project “Laundry Booking System” is a software which consists in booking a machine in a laundry. This software will permits to the customer to manage his booking for the laundry really easily and will be divided in 2 parts for more efficiency.

A server part, which will connects to the database and handle the requests of booking and availability from the client.

A client part, which will be the interface between the booking system and the customer.

This way, the customer will be able to make some reservation or to cancel or delay some of his booking anywhere he is.

# Project Description

## Description of subject

We would like to make a Laundry booking system.

This booking system will consist in some actions:

* To authenticate himself
* To book a machine, a day in the calendar
* The possibility to book a dryer just after a machine
* The possibility to book a cloth line to make your clothes drying
* The possibility to book regularly, each day on a week, etc …

This booking system will be composed of 2 softwares:

### The server

The server will handle these parts:

* The database connection, which will make possible to save booking
* Socket management to accept the new connection from clients.
* Thread management to make easier the handling of the clients

### The client

The client will handle these parts:

* The calendar module which will display a calendar to make easy to book a machine, etc…
* The GUI in which the booking system will be displayed
* The booking system itself, which will make the booking, to cancel or delay a reservation, etc …
* The Newtork module where is handled the protocol and the socket to the server

We think that points are the mainline of the project, but they will be exposed to some modification during the development part.

## Task repartition

The work is divided as sort of we are obliged to work together. Each module will be independent and get a highly cohesion for the integration.

We will need to display correctly the calendar in the GUI and the Database should be connected with the booking system. The exchange of data between each module will be discussed and defined well to be sure of the behavior the program will get.

The tasks are divided this way:

* Laurent will be in charge of the socket management, the GUI and the booking system client-side.
* Rémi will be in charge of the Socket, Thread and database management Server side and the calendar module client-side.

# Document Conventions

DB represents the database.

GUI represents the graphical interface.

“user” will be used to describe the customer.

“client” will be used to describe the client part of the software.

“server” will be used to describe the server part, with the database connection.

“Booking” will represent a booking for a washing machine.

“BS” represents the booking system.

# System Description

A user will authenticate himself on the client part. The client is connecting to the database and search for the credentials of the user. When the user is authenticated, the client is requesting the data about the booking of the user, and the availability of the washing machines. Then when a booking is done, the server will receive the information of the booking and will check if the machine is free. If the machine is free the booking is done and the server returns a confirmation message or an error message in the case that the booking is not possible.

# User Interface

The user will only use the client. He will authenticate himself on the authentication screen. When authenticated, he can see his booking if he has, or the calendar with the availability of the machines. He can click on a slot on the calendar to make a booking or he can enter the wanted data in a specific field. He can also cancel or delay a booking if he wants, on an existing booking. He can also make a regularly booking by checking a box and then specify on how many weeks he wants.

# System Functions

Here is a list of mandatory requirements:

1. The user shall be able to create an account.
2. The user shall be able to authenticate on the client.
3. The user shall be able to log out.
4. The user shall be able to book a machine at a given time for 1 hour.
5. The user shall be able to cancel a booking (single day booking or regularly booking).
6. The user shall be able to delay a booking (single day or regularly booking).
7. The user shall be able to select a slot on the calendar.
8. The user shall be able to see the availability of a machine on the calendar.

Here is the list of optional requirements:

1. The user shall be able to book a dryer (30 minutes).
2. The user shall be able to book a machine and a dryer if available just at the end of the machine (1h + 30minutes).
3. The user shall be able to book a machine regularly each day of a week, for a given period.
4. The user shall be able to use different type of powder and softener.

# Non-functional Requirements

1. The server will be accessible by the network.
2. The passwords will be encrypted.
3. The user should be able to book a machine in 2 clicks.
4. The calendar should display a complete week.

# Storage of permanent data

The database will store the booking system by datetime values. There will be two tables:

* LaundryUsers: will contain the encrypted password, an Id and a username. Eventually an email and a phone number.
* LaundryBooking: will contain the booking of users. A booking contains the id of the user, the datetime, the number of the machine. It will may also contain a Boolean which will say if this is a regularly booking or not.

# Use-cases of the System

Use case 1: creating an account and log out.

A user creates an account by clicking on the button “Create an account” on the identification screen. Then he types his login, the password and the verification of the password. When he clicks on “Create”, the account is created or an error is displayed if the login already exists. If there is an error, the user has to change the login. When the account is created, he enters his login and his password in the correct fields. Then, when the user is connected, he can click on the button “Log out” to log out.

Use case 2: making a booking.

The user connects himself on the client by seizing his credentials. Then he clicks on the button “Book a machine”. He has then to enter the wanted day and time. When he has finished, he clicks on “Confirm the booking”, and if the booking is possible, a window display that everything is okay. If not, there is an error message.

Extension:

The user authenticates. The calendar displays the availability of the machines. Then the user can select an available slot by clicking on the wanted slot on the calendar, a window opens and display the data of the slot. Then the user can click on “Confirm the booking” to make a booking.

Use case 3: Cancelling or delaying a booking.

The user authenticates. He click on “See my booking”, then the client displays the booking the user did. When a he click on a booking, he can see the data of the booking. He has clicks on “Delay” and has to enter the new data for the booking. The client displays a confirmation message or an error if it’s not possible to change to the new time. Then the user can cancel a booking by clicking on it and then clicking on “Cancel a booking”. He has to confirm his choice to cancel the booking. If not, the booking isn’t modified.