

# **Software Requirements Specification**

**For**

**Team 3**

**October 28, 2019**

**Version 1**

**Prepared by:**

**Bachir Karim**

**Montresor Riccardo**

## **Table of Contents**

- 1 INTRODUCTION
  - 1.1 Overview
  - 1.2 Goals and Objectives
  - 1.3 Scope
  - 1.4 Definitions
- 2 GENERAL DESIGN CONSTRAINTS
  - 2.1 Chat\_Protocol\_Client Environment
  - 2.2 User Characteristics

- 2.3 Mandated Constraints
- 3 NONFUNCTIONAL REQUIREMENTS
- 3.1 Operational Requirements
- 3.2 Performance Requirements
- 3.3 Security Requirements
- 3.4 Documentation and Training
- 3.5 External Interface
  - 3.5.1 User Interface
  - 3.5.2 Software Interface
- 4 FUNCTIONAL REQUIREMENTS
- 4.1 Required Features
  - 4.1.1 Use Case:
  - 4.1.2 Use Case:
- 4.2 Optional Features
  - 4.2.1 Use Case:
  - 4.2.2 Use Case:

# 1 INTRODUCTION

## 1.1 Overview

The Chat\_Protocol\_Client application will be a Computer app. The application will provide the possibility to users to connect to a Chat\_Server that allows the user to chat with others that are connected too.. This document provides information on the requirements for the Chat\_Protocol\_Client software application. Project goals, scope and definitions are given in the introduction. Design constraints and application environment are described in the following section. Non-functional requirements are outlined for later verification. Functional requirements are given to show the system features and expected user interaction. Project constraints will be included in separate documentation. The Software Project Management Plan will give specifics on project budget and schedule..

## 1.2 Goals and Objectives

The main objective of this project is to allow students a way to chat to each other. The Chat\_Protocol\_Client is expected to:

1. Provide an interface where the students can log in and chat in it

2.The GUI have to allow the students to chat in a public or private room

## 1.3 Scope

The Chat\_Protocol\_Client application will provide users with the ability to chat with other people from a pc. Users will be able to chat with other people by logging in to the server specifying the username and the Topic(the public room). Once logged in the user has the possibility to chat in private with the online users or to chat with all of them in the public room.

## 1.4 Definitions

**Chat\_Protocol\_Client** – the product that is being described here; the software system specified in this document. **Project** – activities that will lead to the production of the Chat\_Protocol\_Client. **Client** – the person or organization for which this Chat\_Protocol\_Client is being built. **User** – the person or persons who will actually interact with the Chat\_Protocol\_Client. **Use case** – describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.

**Scenario** – one path through a user case **Actor** – user or other software system that receives value from a user case. **Developer** – the person or organization developing the system, also sometimes called the supplier. **Stakeholder** – anyone with an interest in the project and its outcomes. This includes clients, customers, users, developers, testers, managers and executives.

## 2 GENERAL DESIGN CONSTRAINTS

### 2.1 Chat\_Protocol\_Client Application Environment

The Roo Balance product will include a mobile app designed to work on a windows Platform. This application will interface with a server to complete the registration and then to start chatting. This server will be the intermediary device that allows the communication between the users.

### 2.2 Mandated Constraints

The application will run on a computer. This application was built with netbeans and the language used to build it was java

## **3 NONFUNCTIONAL REQUIREMENTS**

### **3.1 Operational Requirements**

Usability: 95% of users will not need to read the user manual to be able to use the application.

### **3.2 Performance Requirements**

Maintainability: Changes made to the Developer can be adopted but they can altering the computer application. tanti messaggi in arrivo non devono bloccare l'applicazione

### **3.3 Security Requirements**

The Chat\_Protocol\_Client app has one features. Infact when a user start the application the first GUI that will appear is an interface where he have to insert is nickname and Ip of the Server or the other user machine, but every data is in clear.

### **3.4 Documentation and Training**

The Roo Balance application will be delivered to users as a download without documentation or training. A user guide and system documentation will be provided to project stakeholders.

### **3.5 External Interface**

#### **3.5.1 User Interface**

The user interface will be eye-catching and visually appealing. When users access their Roo Bucks accounts, the interface will provide a smooth transition with the Manage My ID website which has a straightforward, understated look and feel. The interface will be intuitive. As a mobile app it will be streamlined and simple to use. No training will be provided and it is expected that 95% of users will be able to use the app without any training.

#### **3.5.2 Software Interface**

The Roo Balance proxy server will serve as an interface between the iOS application and Manage My ID website. It will enable interaction between the user and the remote website.

# 4 Functional Requirements

## 4.1 Required Features

### 4.1.1 Use Case: 1

**Description: User Login / Check Roo Bucks Balance** Actors: student or any Roo Bucks user  
Value = high Cost = high

Basic Path

1. User clicks icon for Roo Bucks application.
2. System prompts user to select an option: View Account or View Locations.
3. User clicks View Account option.
4. System prompts user to enter user e-mail and password.
5. User enters correct user e-mail and password and clicks Login.
6. System displays Account Summary with Roo Bucks balance with options to logoff or view transactions.
7. User clicks Logoff.

8. System exits. Alternate Path

9. User clicks icon for Roo Bucks application.
10. System prompts user to select an option: View Account or View Locations.
11. User clicks View Account option.
12. System prompts user to enter user e-mail and password.
13. User enters incorrect user e-mail and/or password and clicks Login.
14. System displays error message: "Invalid Email Address and / or Password for [user@wrongaddress.com](#) ..... Or you may have exceeded the number of consecutive attempts allowed. Please try again later."
15. User may choose to login again, returning to step 1, or exit.
16. System exits.

### 4.1.1 Use Case:

**Description: Find Where to Spend Roo Bucks** Actors: student or any Roo Bucks user Value = high Cost = medium Basic Path

1. User clicks icon for Roo Bucks application.
2. System prompts user to select an option: View Account or View Locations.
3. User clicks View Locations option.
4. System displays Roo Bucks locations, possibly on more than one screen, with an exit option.
5. User clicks exit.
6. System exits.

## 4.2 Optional Features

### 4.1.2 Use Case:

**Description: Check Roo Bucks Recent Transactions**

Actors: student or any Roo Bucks user Value = medium Cost = high Basic Path

2. Following Login [Use Case 1 Step 6]: System displays Account Summary with Roo Bucks balance with options to logoff or view transactions.
3. User clicks View Transactions.
4. System displays recent transactions with option to logoff.
5. User clicks Logoff.
6. System exits.