**Group 1**

**<Coding Turk System>**

**Software Requirements Specification**

**For Desktop Version**

**Version <1.0>**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/12/2017 | 1.0 | Version 1.0 for Desktop Version | Joseph Schauer  Suman Islam  Qizhi Zhao  Weicheng Huang |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Introduction 4](#_TOC_250012)
   1. [Purpose 4](#_TOC_250011)
   2. [Scope 4](#_TOC_250010)
   3. [Definitions, Acronyms, and Abbreviations 4](#_TOC_250009)
   4. [References 4](#_TOC_250008)
   5. [Overview 4](#_TOC_250007)
2. [Overall Description 4](#_TOC_250006)
   1. [Use-Case Model Survey 5](#_TOC_250005)
   2. [Assumptions and Dependencies 5](#_TOC_250004)
3. [Specific Requirements 5](#_TOC_250003)
   1. [Use-Case Reports 5](#_TOC_250002)
   2. [Supplementary Requirements 5](#_TOC_250001)
4. [Supporting Information 5](#_TOC_250000)

**Software Requirements Specification**

# Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

## Purpose

The purpose of this document is to give a detailed description of the requirements for the “Coding Turk System” (CTS) software. It will illustrate the purpose and complete declaration for the development of system. It contains an overview of the use-case model, description of technical feasibility, specific requirements, functional and nonfunctional requirements, and supplementary requirements.

## Scope

CTS is a desktop application which client can hire develops to implement applications. The application should available in window, mac and Linux system.

This is system where registered users can be either a client who posts system request or a developer who implements the system(s) a client posted. A super-user handles user accounts, money related issues and proctor user activities. The super-user’s id and password can be hardcoded. A client can post a project with the system requirements with the options to bid for the developers. The winning bidder will be chosen by the client and a portion of the fee will be collected by the super user.

Furthermore, this software is just a demo that help student to understand software engineering. Internet is not required for this software. All system information is maintained in a database, which is hard coded in the source code. With the source code, you can run the complete version of this software.

CTS is now a open source software under MIT license.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| Visitor | Software user who are not registered |
| Client | Registered software user, can post a system demand and hire developer |
| Developer | Registered software user, can bid on any demand with promised timeline and money |
| Super-User | System administrator who is given specific permission for managing and controlling the system |
| CTS | Coding Turk System |
| Markdown | Markdown is a lightweight markup language with plain text formatting syntax |

## References

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

## Overview

The rest of this document includes three chapters. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of user and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed term and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter provides supporting information for this document.

# Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of user that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## Use-Case Model Survey

The users we have are the Super-User, Clients, Developers and Visitor. The use case diagram below shows the relations and capabilities of all users. The Super-User is able to manage users such as process/approve application, transaction management and proctor user activity. Visitors are able to see top developers, clients and apply to become either a developer or a client.

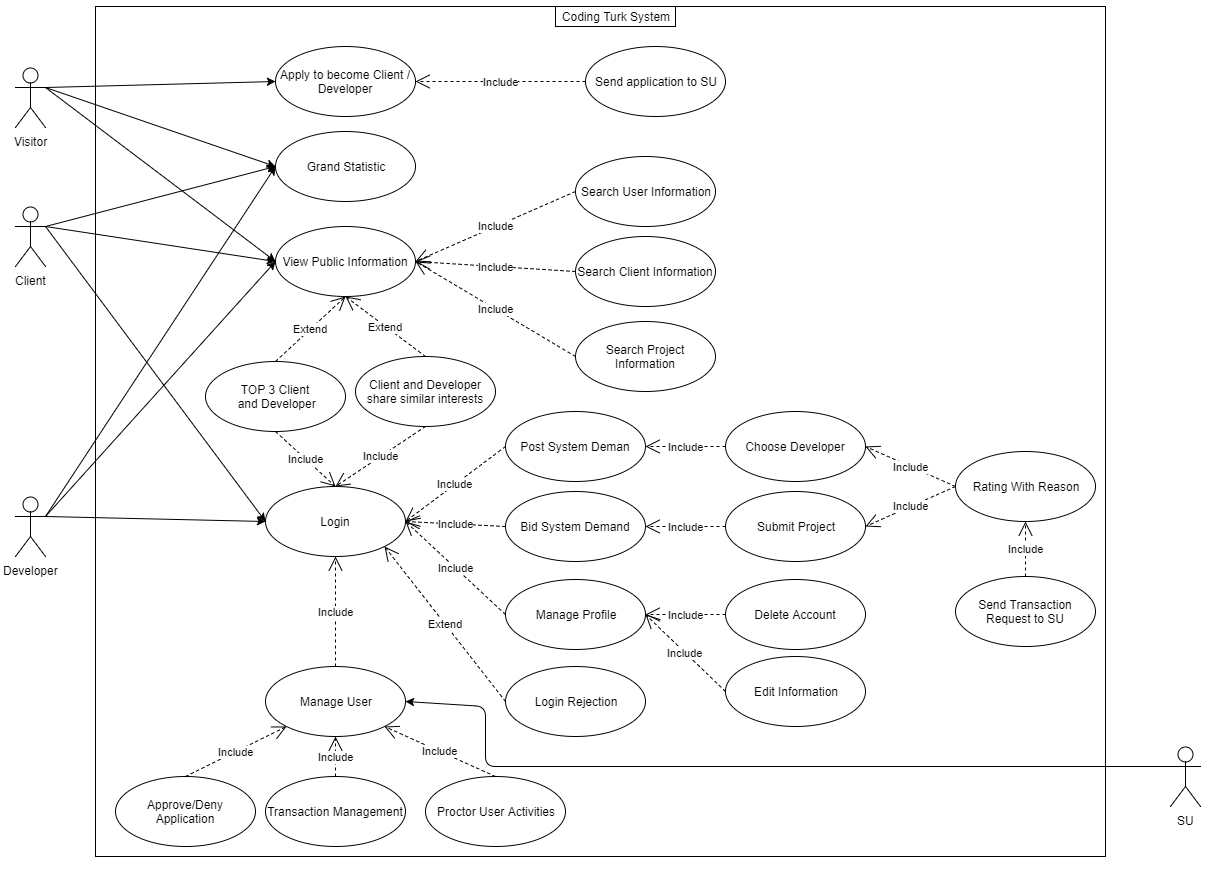


Figure 1. The Use-Case Diagram for the System

## Assumptions and Dependencies

CTS depend on the MariaDB database system.

CTS is an open source software for study purpose, we do not deal with security problem, all user id and password is stored in MariaDB database, super user’s id and password is hardcode in the source code.

CTS is a local desktop application, all information will download and stored in the machine. We assumed that there is enough hard drive space to store all information’s.

# Specific Requirements

The system will be using local hosting desktop application. There is no need for internet and our platform will rely on python and PYQT5. Our database will be built using MariaDB database system. MYSQL might be required as well. There is no other requirements for testers.

## Use-Case Reports

Use Case: Apply for an account

Diagram:

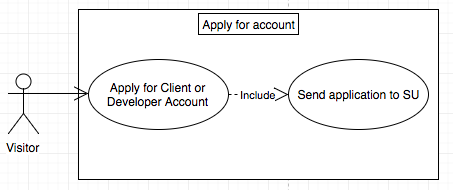
****

Figure 2. Apply for an account Use-Case

Brief Description: Only a visitor can apply to be a client or developer.

Initial Step-By-Step Description:

1. The visitor must choose to apply as a client or developer.
2. The visitor must enter basic information, choose a unique user id and a password, and deposit money.
3. If the super-user denies the application, the visitor will be blocked from logging in with a reason why he/she was rejected.
4. If the super-user accepts the application, the visitor will be shown a welcome screen and prompted to add more information about him/herself. This information will be visible to all visitors, clients, and developers.

Use Case: View/Search Client/Developer Information

Diagram:

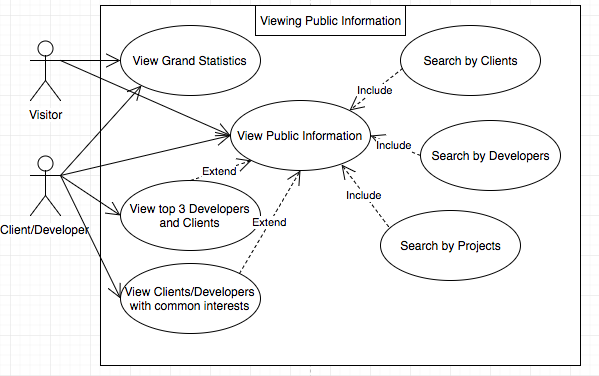


Figure 3. The View/Search User-Case

Brief Description: Visitors, Clients, and Developers can view public information of other clients and developers and search by client, developer, or project.

Step-By-Step Description:

1. A visitor, client, or developer can view the system’s statistics. This includes the number of clients and developers, clients with the most projects, and developers making the most money.
2. A visitor, client, or developer can view public information. A client or developer is shown the top 3 developers and clients by default if their account is new, or clients and developers with common interests if they have a project history.
3. A visitor, client, or developer can search for client information, developer information, or project information by selecting one of the three and entering a keyword such as name to search for.

Use Case: Completing a Project

Diagram:

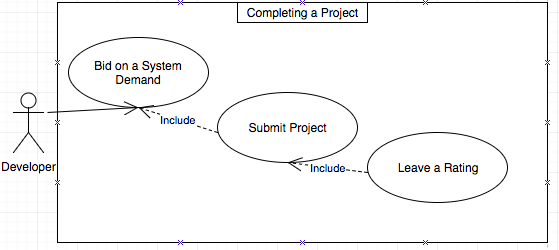
****

Figure 4. The Completing a Project User-Case

Brief Description: A developer can bid to complete a project, and get paid in return for completing it.

Step-By-Step Description:

1. A developer can search for projects to bid on from the search page.
2. A developer can bid on any project. The developer must bid a price.
3. If the developer is selected, then the developer is paid half of the bid from the client up front. The developer should complete and submit the project by the deadline specified by the client.
4. After submitting the project, the developer may rate the client from 1 to 5.
5. The developer will be paid the remainder of the bid as specified in the client’s use case.

Use case: Manage User

Diagram:

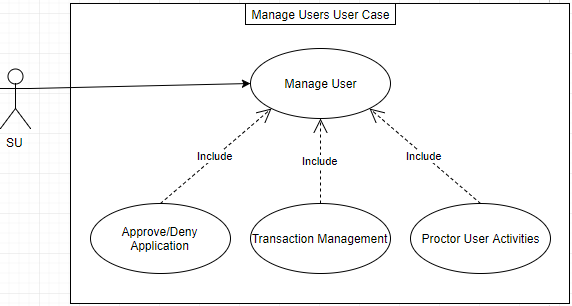
****

Figure 5. The Manage Users User-Case

Brief Description: The Super-User manages the clients’ and developers’ accounts.

Step-By-Step Description:

1. The super-user must go through a different log-in procedure.
2. The super-user can approve or deny visitors’ applications to become clients or developers.
3. The super-user decides how much money the client will pay the developer when the developer gets a bad rating, and can change that rating.
4. The super-user can warn clients and developers if they give too many ratings of 1 or 5. Continued abuse of the rating system can get a user placed on the blacklist for a year.

## Supplementary Requirements

[Supplementary Specifications capture requirements that are not included in the use cases. The specific requirements from the Supplementary Specifications, which are applicable to this subsystem or feature, should be included here and refined to the necessary level of detail to describe this subsystem or feature. These may be captured directly in this document or referred to as separate Supplementary Specifications, which may be used as an enclosure at this point. Make sure that each requirement is clearly labeled.]

# Supporting Information

*[The supporting information makes the* ***Software Requirements Specification*** *easier to use. It includes:*

* *Table of Contents*
* *Index*
* *Appendices*

These may include use-case storyboards or user-interface prototypes. When appendices are included, the **Software Requirements Specification** should explicitly state whether or not the appendices are to be considered part of the requirements.]