

# Requirements for the Kakuro project

## Iteration 2 COMP354

### Team PK-A

15 March 2020

Table 1: Team Members

Name	Role	ID Number
Tiffany Ah King	Quality Control	40082976
Isabelle Charette	Documenter	40008121
Brian Gamboc-Javiniar	Coder	40033124
Vsevolod Ivanov	Organizer	40004286
Chang Liu	Quality Control	40056360
Nolan Mckay	Coder	27873557
Nalveer Moocheet	Quality Control	40072605
Hoang Thuan Pham	Coder	40022992
Audrey-Laure St-Louis	Documenter	27558783
Jia Ming Wei	Documenter	40078192

# Table of Contents

<b>1</b>	<b>System</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Context . . . . .	4
1.3	Business Goals . . . . .	5
<b>2</b>	<b>Domain Concepts</b>	<b>6</b>
<b>3</b>	<b>Actors</b>	<b>7</b>
<b>4</b>	<b>Use Cases</b>	<b>8</b>
4.1	Overview . . . . .	8
4.2	Use Case 1 - Start game . . . . .	9
4.3	Use Case 2 - Submit answer . . . . .	11
4.4	Use Case 3 - Restart game . . . . .	13
4.5	Use Case 4 - Choose difficulty level . . . . .	14
4.6	Use Case 5 - Input number . . . . .	15
4.7	Use Case 6 - Pause game . . . . .	18
4.8	Use Case 7 - Resume game . . . . .	19
4.9	Use Case 8 - Login . . . . .	20
4.10	Use Case 9 - Register . . . . .	21
4.11	Use Case 10 - Save game . . . . .	22
4.12	Use Case 11 - Load game . . . . .	23
4.13	Use Case 12 - View best score . . . . .	24
<b>5</b>	<b>Non-Functional Constraints</b>	<b>25</b>
5.1	Business Constraints - Game Mechanism . . . . .	25
5.2	Software Requirements . . . . .	25
5.3	Non-Functional Requirements . . . . .	25
<b>6</b>	<b>User Interface</b>	<b>26</b>
<b>7</b>	<b>Glossary</b>	<b>30</b>
<b>8</b>	<b>References</b>	<b>32</b>

<b>9</b>	<b>Description of File Format</b>	<b>33</b>
9.1	.java . . . . .	33
9.2	.tex . . . . .	33
9.3	.jar . . . . .	33

# 1 Introduction

The introduction of the document provides an overview of the entire document, briefly introducing what are its goals, and what information is to be found in it.

## 2 Architectural Design

This section must give a high-level description of the system in terms of its modules and their respective purpose and exact interfaces.

### 2.1 Architectural Diagram

A UML class diagram or package diagram depicting the high-level structure of the system, accompanied by a one-paragraph text describing the rationale of this design. It is mandatory that the system be divided into at least two subsystems, and that the purpose of each of these subsystems be exposed here.

### 2.2 Subsystem Interface Specifications

Specification of the software interfaces between the subsystems, i.e. specific messages (or function calls) that are exchanged by the subsystems. These are also often called “Module Interface Specifications”. Description of the parameters to be passed into these function calls in order to have a service fulfilled, including valid and invalid ranges of values. Each subsystem interface must be presented in a separate subsection.

## 3 Detailed Design

Complete description of the system design, describing one subsystem separately in respective subsection. UML class diagrams are to be used, as well as a short textual description describing the purpose of each class.

### 3.1 Subsystem X

#### Detailed Design Diagram

UML class diagram depicting the internal structure of the subsystem, accompanied by a paragraph of text describing the rationale of this design.

## Units Description

List each class in this subsystem and write a short description of its purpose, as well as notes or reminders useful for the programmers who will implement them. List all attributes and functions of the class.

## 4 Dynamic Design Scenarios

Describe some (at least two) important execution scenarios of the system using UML sequence diagrams. These scenarios must demonstrate how the various subsystems and units are interacting to achieve a system-level service. Units and subsystems depicted here must be compatible with the descriptions provided in section ?? and ??.