Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 11/03/14 | 1.0 | Write specifications for the TicTacToe Project | Eren Sezener |
| 11/03/14 | 1.1 | Write requirements for the TicTacToe Project | Deniz Sökmen |
| 11/03/14 | 1.2 | Write requirement specifications | Deniz Sökmen |
| 12/03/14 | 1.3 | Update user & system requirements | A. Emre Ünal |
| 12/03/14 | 1.4 | Update system requirements specifications | Deniz Sökmen |

**TABLE OF CONTENTS**

Revision History 1

1 INTRODUCTION 3

1.1 Document overview 3

1.2 Abbreviations 3

1.3 References 3

1.4 Conventions 3

2 REQUIREMENTS 5

2.1 User Requirements Specification 5

2.2 System Architecture 5

2.3 Use Cases and Usage Scenarios 5

2.4 System Requirements Specification 5

# INTRODUCTION

## Document overview

This document presents the software requirements specifications of TicTacToe game software development project.

It describes:

* Overview of software system to be developed
* Conventions to be used during the development of the project
* Requirements of functionalities
* Architecture of the software
* Use cases and usage scenarios,
* The compliance of requirements to user's needs
* The relative importance and precedence of requirements

## System overview

Tic-Tac-Toe game has been Turkey's national sport since Sir Muharrem Tic-Tac-Toe invented the game in 500 BC. Although the game is popular in many Kahvehanes (Turkish Cafe), the game is rarely played on the Internet due to the lack of high quality software. TicTacToe Project aims to fill this gap by enabling users to play Tic-Tac-Toe against each other. The game will have peer-to-peer architecture and users will connect to each using their IP addresses.

## Abbreviations

The TicTacToe game software project: “the game” or “the software”

The TicTacToe game software project’s GitHub repository page: “the repo”

## References

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| [R1] | SDPv1.2 | Software Development Plan of the TicTacToe game, version 1.2. |

## Conventions

Requirements listed in this document are constructed according to the following structure:

Requirement Id

Requirement title

Requirement description

Requirement version

# REQUIREMENTS

## User Requirements Specification

SRS-REQ-001 LOCAL

Playing a local game

The user should be able to play a local game, with 2 players in one computer.

V1.0

SRS-REQ-002 JOIN

Joining a remote game

The user should be able to join a game through the network, to play with a remote opponent.

V1.0

SRS-REQ-003 HOST

Hosting a remote game

The user should be able to host a game through the network, to play with a remote opponent.

V1.0

## System Architecture

The main screen is composed of three GUI elements, a label and two buttons. Clicking the Multiplayer button shall clear the frame and display network related buttons labeled as “Join a Game” and “Host a Game”. In the multiplayer mode, the network modules with TCP/IP protocol shall be created. If the game is created in host mode, the network module shall start listening the port 2593 as a server and the game shall be blocked until there is an accepted connection. After the accepting process, the actual game shall begin. The host program’s actions shall directly be displayed on the host and the associated action shall be sent to the peer through the network as a packet. If the game is created in client mode, the user shall be expected to enter the IP address of the host. If the connection is successful, current frame shall be cleared and the actual game screen shall be displayed. If the client does any action, the action shall be directed to the server through the network, instead of displaying the action, the client should wait for the response from the server. Therefore in the networking mode, every action shall be sent through the network to the other peer, every action is treated as a network packet.

In the local mode, the game shall not be associated with the network module. Therefore the game logic shall behave as if there are two players on the same computer, instead of attempting to send any action through the network, the game shall directly respond to any action. Since there is not any association with the network module, the players shall not have to be aware of the game actions, the system shall directly compute any action.

## Use Cases and Usage Scenarios

Include here the relevant UML use case diagrams and sequence diagrams that depict the usage scenarios of the system.

## System Requirements Specification

**Functional Requirements**

* The user should be able to connect to another player to play.
* The user should be able to host a game to accept other players.
* The user should be able to create a local game without network to play on the same computer.

**Nonfunctional Requirements**

* The user should be able to connect to a game in 5 seconds before timeout.
* The program should check if the game is won after a packet is received
* Every action of the user should be sent over the network as packets in the network mode

SRS-REQ-001.1 LOCAL

Creating a Local Game

When one player plays, the system should check whether the game is finished.

V1.0

SRS-REQ-001.2 LOCAL

Creating a Local Game

Provided the game is not finished, the system should change the turn.

V1.0

SRS-REQ-002.1 JOIN

Connecting a Game

The game should be able to connect to the internet.

V1.0

SRS-REQ-002.2 JOIN

Connecting a Game

The game should be able to register a socket to join a game.

V1.0

SRS-REQ-003.1 HOST

Hosting a game

The game should be able to connect to the internet.

V1.0

SRS-REQ-003.2 HOST

Hosting a game

The game should be able to bind a socket to a port to host a game.

V1.0