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 |
|  |  |  |  |

**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 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

## References

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| [STD1] |  | Add your documents references.  One line per document |

## Conventions

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

Requirement Id

Requirement title

Requirement description

Requirement version

Example:

SRS-XXX-000

Title of XXX-000 requirement

Description of XXX-000 requirement

Version of XXX-000

Every system requirement id includes the id of the corresponding user requirement as a prefix.

Example:

SRS-XXX-000.1

Title of XXX-000.1 system requirement, which is related to the user requirement XXX-000

Description of XXX-000 requirement

Version of XXX-000

# REQUIREMENTS

## User Requirements Specification

The user shall first see the main screen captioned with “Welcome” and shall be provided with two buttons: Local Game and Multiplayer. If the user chooses Local Game, he/she shall be redirected to the game screen. If the user chooses Multiplayer, he/she shall be redirected to a screen with two buttons again: Host a Game and Join a Game. In this screen, if the Host a Game button was chosen, a new screen captioned with “Waiting for the opponent” shall be displayed until another user chooses Join a Game and enters the IP and port addresses of the host. After a host is matched with an opponent, they shall be redirected to the game screen. Game screen consists of 9 buttons and a text panel on top of them indicating whose turn it is. The buttons are used for putting X and O marks. The game has classical TicTacToe rule set so whenever the user clicks a button, the game shall check these rules and if the game is over, the players shall get a pop-up indicating the game is over and who the winner is and they will be redirected to the main screen.

SRS-XXX-010 SAMPLE

Hosting a Game

The user can host a game through the network to find an opponent.

V1.0

SRS-XXX-020 SAMPLE

Local Game

When the user plays local game, the system shall host a game as if there are two players on the same computer, clicking a button will end the turn for the player and it will be the other player’s turn on the same computer.

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 labelled 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-XXX-030.2 SAMPLE

Network Packets

TicTacToe uses TCP/IP protocol, therefore ensures that every game action shall be delivered to the other player as a packet completely and shall drop the connection in case of delivery problems. These packets&actions are:

* Clicked on a button,
* Joined the game,
* Left the game

V1.0

SRS-XXX-040 SAMPLE

Connecting a Game

In the network mode, attempting to connect a game shall create a TCP/IP socket and shall connect to the specified IP address on port 2593 and shall wait for 5 seconds until timeout, If there is no response in 5 seconds, the socket shall be closed.

V1.0