

Software Requirements Specification

for

Chess

Version 1.0

|  |  |  |
| --- | --- | --- |
| Daniel Maida | 11141203 | [daniel.maida@wsu.edu](mailto:daniel.maida@wsu.edu) |
| Nazar Stelmakh | 11441299 | [nazar.stelmakh@wsu.edu](mailto:nazar.stelmakh@wsu.edu) |
| Steven Call | 11451145 | [steven.call@wsu.edu](mailto:steven.call@wsu.edu) |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | Date: Octover 7, 2016 |
|  |  |
|  |  |

Contents

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 4

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

3 Specific Requirements 6

3.1 External Interface Requirements 6

3.2 Functional Requirements 7

3.3 Behaviour Requirements 7

4 Other Non-functional Requirements 8

4.1 Performance Requirements 8

4.2 Safety and Security Requirements 8

4.3 Software Quality Attributes 9

5 Other Requirements 9

Appendix A – Data Dictionary 10

Appendix B - Group Log 11

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Daniel Maida  Nazar Stelmakh  Steven Call | Initial draft with all sections coplete | 10/06/16 |

# 

# Introduction

## Document Purpose

This document specifies the requirements for implementing a first version of a Computer Chess game implementing GUI and Artificial Inteligence features. The main purpose of this document is to ensure the development team has an in depth understanding of the project requirment. This SRS describes the implementation of GUI components and the underlaying Artificial Inteligence that will allow the user to engage in an epic battle between man vs. machine. The user will have a choice between playing against another user or playing agaisnt the A.I. In addition, this document will describe the expected users of the game, use cases and the interaction between the GUI and the user.

## Product Scope

This software will provide an interactive interface for a Computer simulated Chess game. The project goal is to allow the user to play against Artificial Intelligence at variying diffuculties as well as playing other users in the same playing interface. The GUI will provide a clean interface that allows the user to have an immersive playing experience.

## Intended Audience and Document Overview

This document is intended to be read by the developer team, client, and Professor Xinghui Zhao of WSUV who is advising and grading the project.

## Definitions, Acronyms and Abbreviations

IEEE – Institute of Electrical and Electrons Engineers.

AI – Artificial Inteligence

JDK – Java Development Kit

Java – Programming language

## Document Conventions

This document conforms to IEEE formatting standards for a SRS. No other special formatting which differs from the IEEE standard has been used in this document.

## References and Acknowledgments

No outside works are referenced in this document.

# Overall Description

## Product Perspective

This Chess interface will be a self contained system that is being developed for a 300 level software engineering project. This system will be developed with the intent of allowing users to select from multiple game options, as well as select who their opponent will be. This system will have multiple modes of gameplay allowing different features to be enable or disabled based upon preferance. No outside devices will be needed to play or interact with this system, outside of the normall components of a computer.

## Product Functionality

* Chess Game
  + AI Mode
    - Multiple levels of difficulty
    - Next move assistance
    - Undo last move
    - Allow user to choose who makes initial move or randomly select first player
    - Modify environment
      * Choose Chess board style
      * Choose Chess piece style
  + Player vs. Player
    - Modify environment
      * Choose Chess board style
      * Choose Chess piece style
    - Undo last move
    - Allow user(s) to choose who makes initial move or randomly select first player

## Users and Characteristics

The game should be useable by players with the basic understanding of the rules of chess. These users will be the most important users for this system. Our system will be functional to all users from amatuers to very skilled chess players.

## Operating Environment

The Computer Chess game will be developed using JavaFX. The game will be able to run on any platform such as Windows, Mac OS X, and Linux that meet these minimum system requirments:

Software Requirments:

Windows XP SP+

Mac OS X 10.8+

Most Linux distributions

JDK 1.8.x +

## Design and Implementation Constraints

The Computer Chess game is a simple application designed to give the user the experience of playing chess on a computer. The game does not require any specialized hardware. The application will self contained and does not require any type of database or external software.

## User Documentation

There will be no external user manuals or tutorials delivered along the software. The only requirements for this software is to have basic knowledge of chess and being able to navigate through a simple GUI.

## Assumptions and Dependencies

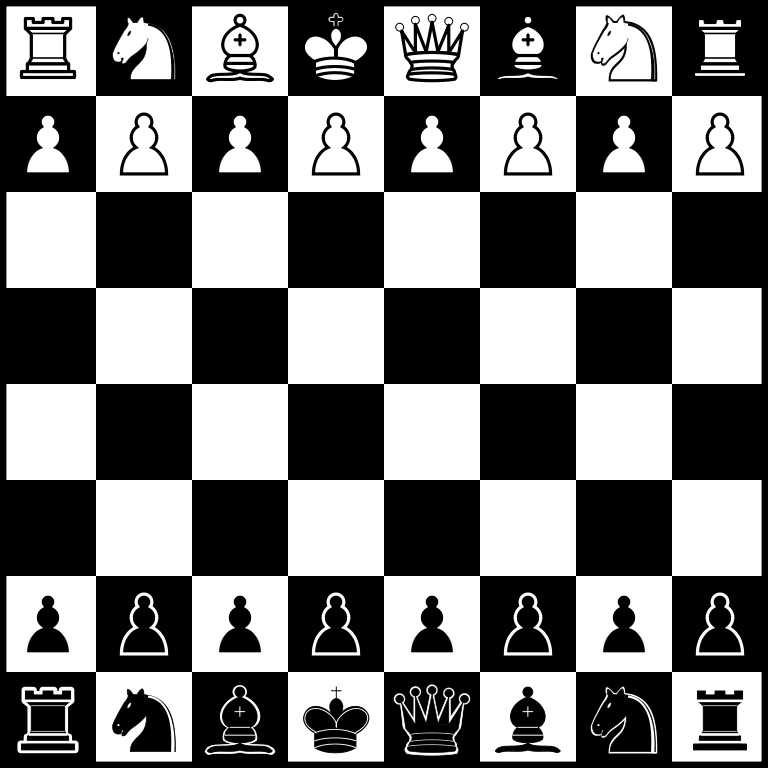
The only dependency for the Computer Chess game is to have the latest version of the JDK.

# Specific Requirements

## External Interface Requirements

### User Interfaces

The main window of game will have the chess board front and center. When the game is first first initialized a basic settings window will pop up. In that settings window the user will be able to choose their color and what color starts first. The user will also have the option to play agaisnt another player or play agaisnt the A.I. engine. There will be a menu in the main window where the user can change the theme of the board. The following is a rough outline of what the main window and the default theme will look like.



### Hardware Interfaces

The chess pieces on the board will be controlled using only a mouse. Any mouse that is compatible with the operating system will work. The mouse needs to have a primary click setting. No other external hardware is required to interact with the program aside from initializing it with the command line.

### Software Interfaces

The Chess game implementation wll be built on top of a javafx framework.

### Communications Interfaces

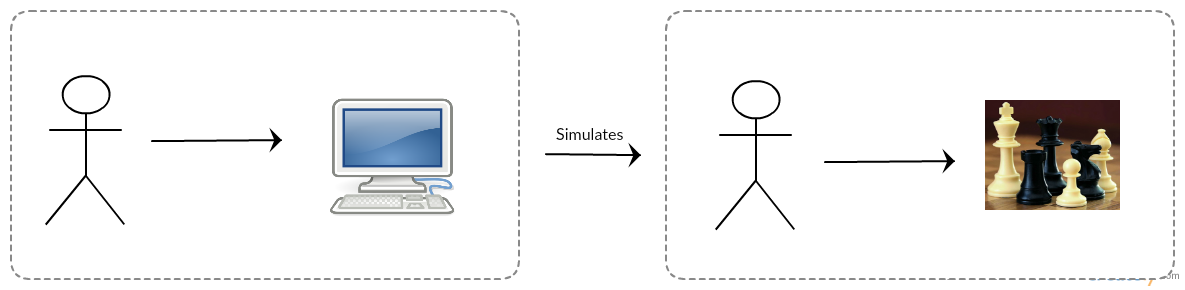
The Chess game implementation does not require any network communications.

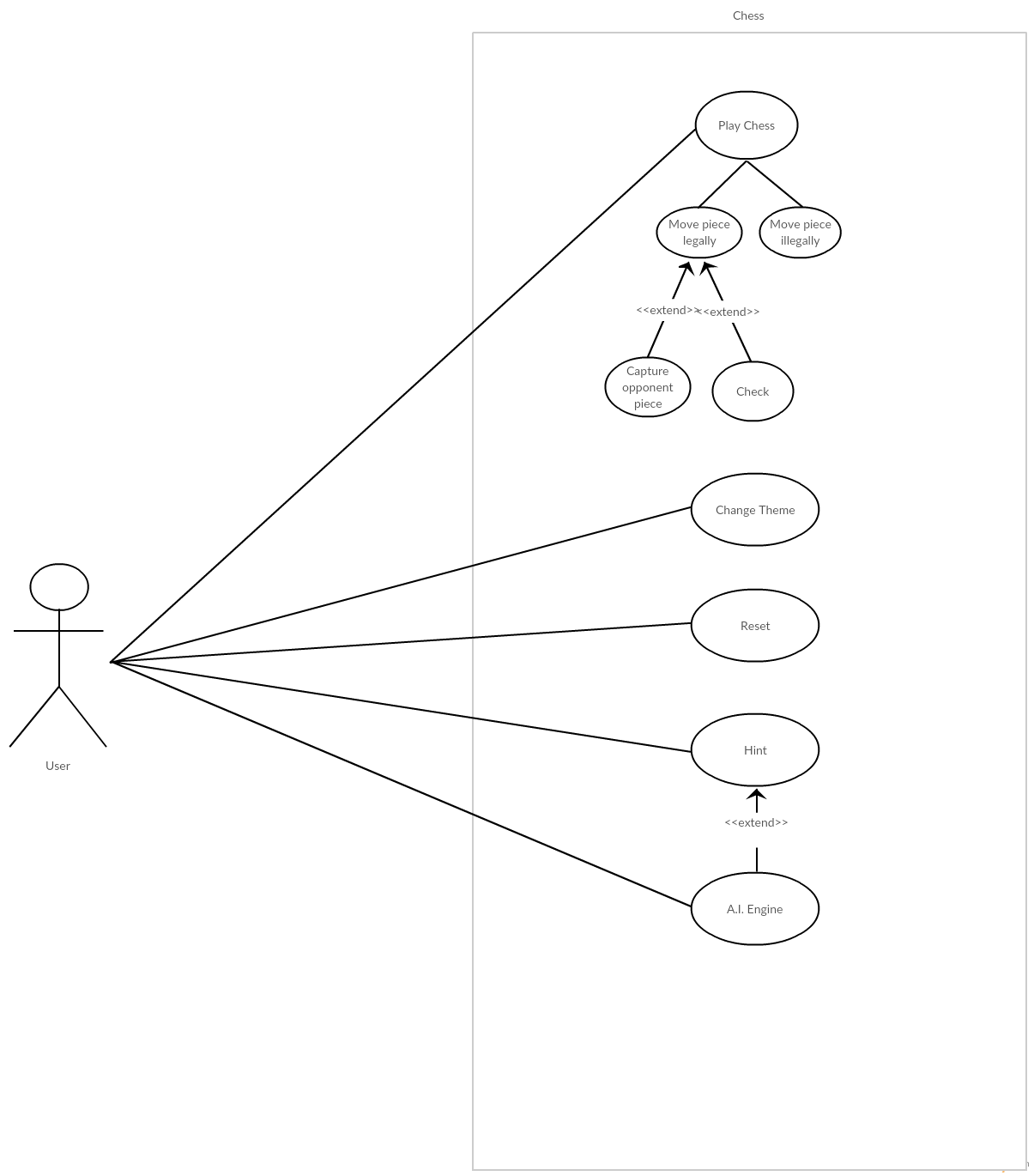
## Functional Requirements

|  |  |
| --- | --- |
| Requirement | Description |
| REQ 1 | Software start-up |
| REQ 1.0 | The software will allow the user to select between AI and PvP mode. |
| REQ 1.1 | In AI mode, the user will compete against the AI |
| REQ 1.1.1 | The user will be allowed to change the difficulty of the AI |
| REQ 1.1.2 | The user will be able to ask for a hint in which the AI will provide a best strategy move |
| REQ 1.1.3 | The user will be able to undo the last move |
| REQ 1.1.4 | Allows the user to select who makes the first move or allows the computer to randomly choose |
| REQ 1.1.5 | The user will be able to adjust the style |
| REQ 1.1.5.1 | The user will be able to choose between different board styles |
| REQ 1.1.5.2 | The user will be able to choose between different piece styles |
| REQ 1.2 | In PvP mode, the user will be able to compete against another user |
| REQ 1.2.1 | The user will be able to adjust the style |
| REQ 1.2.1 | The user will be able to choose between different board styles |
| REQ 1.2.2 | The user will be able to choose between different piece styles |
| REQ 1.2.2 | The user will be able to undo the last move |
| REQ 1.2.3 | Allows the user to select who makes the first move or allows the computer to randomly choose |

## Behaviour Requirements

### Use Case View





**Play Chess**

The user will be allowed to move chess pieces across the board given that it is a legal move.

**Change Theme**

The user will be able to change the style of the board.

**Reset**

The player will able to restart the game.

**Hint**

The user can option to see a suggested next move given by the A.I. engine.

**A.I. Engine**

A.I. will calculate possible future moves and choose the move with best outcome.

# Other Non-functional Requirements

## Performance Requirements

|  |  |
| --- | --- |
| Requirement | Description |
| REQ 1.0 | The A.I. engine should process a move in a reasonable amount of time. |
| REQ 2.0 | Every move made by the user will be processed instantanously. |
| REQ 3.0 | The game should load in less than three seconds. |
| REQ 4.0 | Changing themes will be instantaneous. |

## Safety and Security Requirements

The solfware will not contain any sensitive information. Thus, minimum security is needed.

## Software Quality Attributes

**4.3.1 Usability**

The interface will be designed to be user-friendl and should require minimum instruction to navigate/use. The chess board will function as defined by the client. Instructions and explanations will be provided in the help menu.

**4.3.2 Maintainability**

The The software’s code will be written to allow for future updates and documented properly. The code will be commented to allow easier maintenance. The code will follow the coding conventions provided by professor Xinghui Zhao.

**4.3.3 Portability**

The software will be developed using javafx and should function properly on any hardware that supports javafx.

# Other Requirements

Not applicable to this system.

Appendix A – Data Dictionary

Not applicable to this system.

Appendix B - Group Log

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Meeting/Collaborative  Work Time (hrs) | Individual Work Time (hrs) | Total Time (hrs) |
| Daniel Maida | 6 | 1 | 7 |
| Nazar Stelamkh | 6 | 1 | 7 |
| Steven Call | 6 | 1 | 7 |