**Software Requirements Specification**

**for**

**Solitaire**

**Version 1.0**

**Prepared by Group 5a**

**(Cloette Owensby, Connie Uribe, Bolun Zhang)**

**DePauw University**

# Introduction

## Purpose

*Solitaire is intended to entertain an audience by having the audience manipulate a layout of cards with the goal of sorting them in some manner and using up all one’s cards. This software will replicate several versions of Solitaire with a playing table layout display, interactive cards that can be manipulated if the game conditions are met and the move is permissible, and a start menu that allows the user to select which type of Solitaire game they would like to play.*

## Intended Audience and Reading Suggestions

*This document was developed both as a guideline for student developers, and also as a reference for users seeking more information about the different types of solitaire and game controls.*

## Project Scope

*The goal of this application is to create multiple types of solitaire games with a functional and practical user interface. The application will manage the shuffling of the deck or decks, the dealing of the cards, the display of the layout, the initial start menu, the movement and accumulation of cards, display hints of additional moves upon request, and display a “Congratulations!” message upon the user’s victory or a “Try Again” message upon their loss (running out of moves).*

*The goal of this application is to facilitate the playing of this game with a digital version. Yes, we are aware that there are multiple free versions of solitaire in most computers and online.*

# Overall Description

## Product Perspective

*This software product is being developed by students of the Object-Oriented Software Development course at DePauw University and is intended for use by student researchers or anyone desiring to play a game of Solitaire. The goal of this project is to develop a feature-rich application which will serve as a functioning prototype for a more comprehensive application which could be developed from extending the codebase.*

## Product Features

*The main features of this product are:*

* *dropdown menu to select different types of solitaire games*
* *manipulating cards, “undo”-ing and “redo”-ing moves*
* *initializing and quitting the program with buttons*

## User Classes and Characteristics

*This application is rated E! for EVERYONE! (including aliens). However, blind users may encounter difficulty using this application. Thus, we suggest future versions feature an audio component. In addition, users with low motor control may encounter difficulty operating the mouse, and this will affect their in-game experience.*

## Operating Environment

*This application is designed to work with a Java Virtual Machine in a desktop environment. Users of this application are expected to be running either a Windows, MacOS, or Linux desktop operating system.*

## Design and Implementation Constraints

*\* We are unsure what to place here. At this time, we do not intend to store any permanent data from these games.*

## User Documentation

*A programmer's guide to working with the software components developed as part of this application will be provided for those who wish to continue development on this application. Additionally, a user's guide will be provided for those who wish for more detail on the rules and controls of the different types of solitaire games.*

# System Features

*The following features, with their associated requirements, will be implemented in the final revision of this software system:*

## Start Menu

3.1.1 Description and Priority

*The Start Menu includes the application icon (which activates the application), the initial “home” interface for the user to select their desired game type, and a “Start” button to confirm the game type and begin loading the desired game layout & interface.*

3.1.2 Functional Requirements

REQ-1: The application must launch upon the user clicking the application icon, and close upon the user clicking the “close” button.

REQ-2: The application will provide an interface for selecting their desired game type (likely in the form of a drop-down menu). The selected game type will be stored and updated when the user re-selects their desired Solitaire type

REQ-3: The application will provide a start button to confirm the selected game type and proceed/replace the current “start” interface with the “game” interface

## Game Types

3.2.1 Description and Priority

*A variety of different Solitaire games are available for the user to play, each with different rules and organization instructions. For this application, we will stick to [3, 5, 6, 7?] game types: (insert names here). This system will allow the user to select, move, undo, and redo their moves. It will also display hints about the next available move upon request, and check to see if the user has won or lost the game. Buttons will also be available to quit, restart, or return to the home menu.*

3.2.2 Functional Requirements (will explain in more detail later)

REQ-1: generate deck, game setup

REQ-2: play (move cards)

REQ-3: hints

REQ-4: points

REQ-5: quit this game (back to start?)

REQ-6: end program

## Display Win/Loss and Number of Moves

3.3.1 Description and Priority

*Displays to the user whether they have won, lost, and if they have won, how many moves it took for them to win. Provides a “New Game” button and a “Quit” button.*

3.3.2 Functional Requirements (will explain in more detail later)

REQ-1: The ap detect win/lose

REQ-2: display number of moves the user took

REQ-3: back to start

REQ-4: end program

# External Interface Requirements

## Hardware Interfaces

*\* Unsure what to put here*

## Software Interfaces

*\* Unsure what to put here as well.*

# Other Nonfunctional Requirements

## Performance Requirements

*Queries performed over the citation data must consume minimal system resources so as to be accessible in real-time by users of the system. This application is intended to be used interactively, so users should not be expected to wait for the completion of any of the operations provided by the application.*

## Security Requirements

*It must be possible for the primary user of the application (the instructor) to keep citation information private. This will be achieved through existing user-based security on the host operating system.*

## Software Quality Attributes

*This application will ship with a suite of tests which insure its proper function, even if third-party updates to the source-code are integrated. Additionally, at run time, this application will verify the correctness of any data files it uses, or of any input provided by the user, and issue appropriate error messages in the cases of unexpected or erroneous input.*

**Appendix B: Analysis Models**

Each group is expected to update this document to include at least one additional feature and should identify its functional requirements, and any other associated requirements, within this document. Requirements should be concise, complete, unambiguous, verifiable, and important. Functional requirements describe the software capabilities which must be in place to carry out the task of the indicated feature, and should be indicated in section 3, along with a description of the feature. Other requirements address issues such as security or system interoperability that may be indicated by the inclusion of a new feature, and should be indicated in sections 4 and 5. Check the existing solitaire software linked on Moodle for some feature ideas, and remove this appendix from your final revision of your requirements document.