Software Requirements Specification

for

Trivia Maze

Version 1.0 approved

Prepared by Daylyn Hoxie

Triple Threat

Daylyn Hoxie, Spencer Curley, & Raymond Shiner

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Daylyn Hoxie	11/6/19	Initial draft	1.0 draft 1
Daylyn Hoxie	12/9/19	Baseline following changes after inspection	1.0 approved

1. Introduction

1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the Trivia Maze. This document is intended to be used by the members of the Triple Threat project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

1.2 Project Scope and Product Features

The Trivia Maze will permit a user to navigate a maze by correctly answering trivia questions. A detailed project description is available in the *Class Project* assignment in the CSCD 350 course [1].

1.3 Intended Audience and Reading Suggestions

This document is intended for use by developers, users, client, testers, and documentation writers.

1.4 References

[1] Capaul, Tom. *Software Development Principles,* http://penguin.ewu.edu/cscd350/Spring15/index.html

2. Overall Description

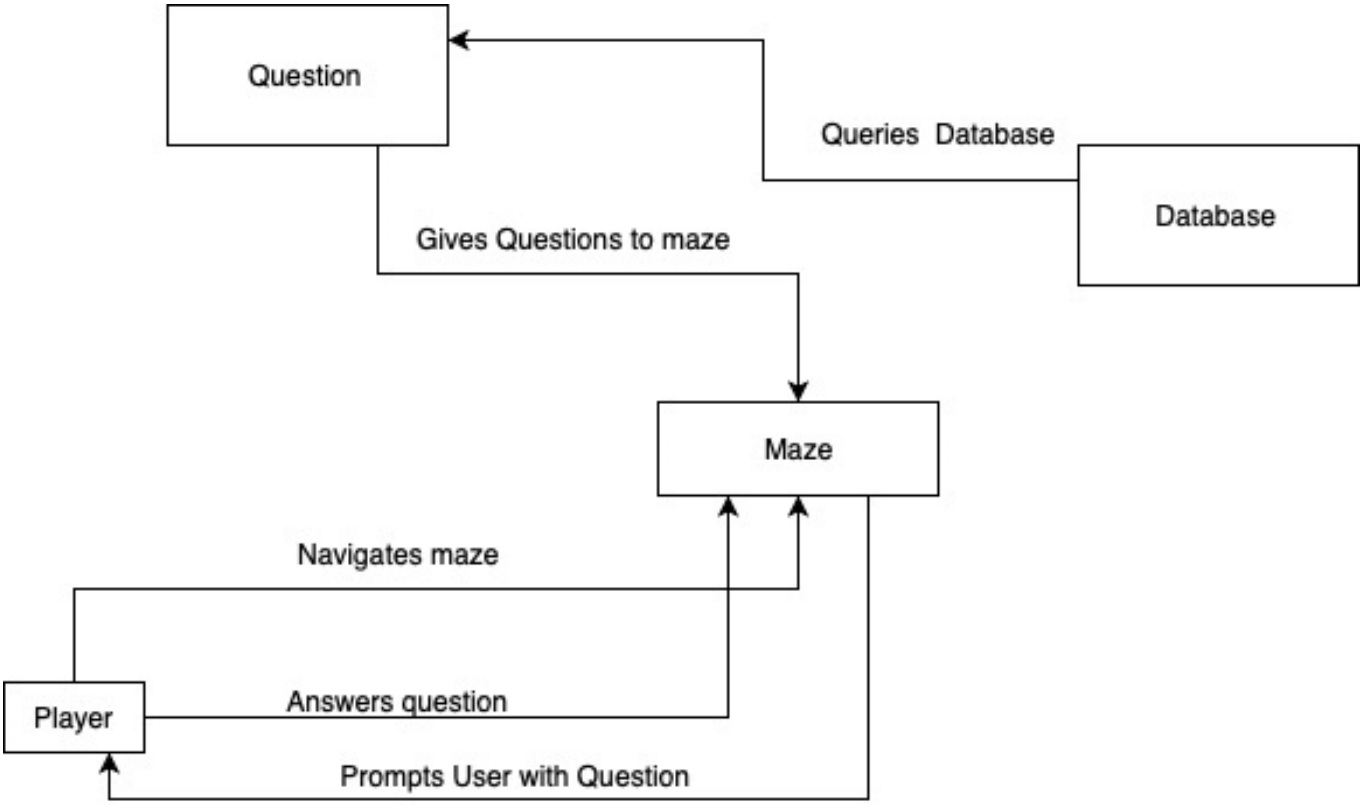
2.1 Product Perspective

The purpose of the Trivia Maze is to create a fun interactive game for players and trivia fans of all ages to enjoy. The simplified UML diagram in *Figure 1* illustrates the system interfaces for release 1.0. The system is expected to evolve over several releases, ultimately connecting to the Trivia Maze to a database containing questions. The database schema in *Figure 2* provides more details of the database and question formats.

2.2 User Classes and Characteristics

Player	A Player is a user of the Game.
Question	A Question is a trivia question that the Player must answer in order to navigate the Maze.
Database	A Database contains the questions and answers for use by the Maze.
Maze	A Maze is the space that the Player must navigate in order to win the Game.

Figure 1:



2.3 Operating Environment

OE-1: The Trivia Maze will operate on any Operating Systems that can run Java programs.

2.4 Design and Implementation Constraints

CO-1: The software will be written using IntelliJ IDEA ULTIMATE 2019.1 and will conform to new updates of IntelliJ as released.
CO-2: The software will be written using JUnit 5 for testing.
CO-3: The software will be written using the SDK Java version 13.0.1.
CO-4: The software will be written to interact with SQLite version 3.30.1.

2.5 Assumptions and Dependencies

AS-1: The user will have Java installed on their machine.

3. System Features

3.1 Play Game

3.1.1 Description

A user will be able to start the game and answer trivia questions to navigate through the maze. Upon reaching the exit, a user will win the game.

3.1.2 Stimulus/Response Sequences

Stimulus:	User requests to start game.
Response:	Software creates maze and starts Player in the entrance.
Stimulus:	Player requests to open door.
Response:	Software provides Player a Question to answer (if door has not already been opened).
Stimulus:	Player answers Question.
Response:	Software determines if the answer was correct and changes door status accordingly.
Stimulus:	Player reaches the exit.
Response:	Software congratulates Player and ends the game.

3.2 Save Game

3.2.1 Description

A user will be able to save the game, quit the game, and reload a saved game.

3.2.2 Stimulus/Response Sequences

Stimulus:	User requests to save game.
Response:	Software saves the game in the save slot.
Stimulus:	Player requests to quit game.
Response:	Software ends the game.
Stimulus:	Player requests to load game.
Response:	Software determines if a game was saved. If so, the saved game is loaded.

3.3 Add Questions

3.3.1 Description

An admin will be able to add new questions to the database.

3.3.2 Stimulus/Response Sequences

Stimulus:	User requests to add a question.
Response:	Software navigates User to Admin tool.
Stimulus:	Admin requests to add a short answer, true-false, or multiple-choice question.

Response: Software prompts Admin to provide information for the question and answer and adds the information into database.

4. External Interface Requirements

4.1 User Interfaces

UI-1: The Trivia Maze will provide standard keyboard functions that will be displayed on every screen.

UI-2: Error message displays will be standardized to return the Player to the last place they were before the error occurred.

4.2 Software Interfaces

SI-1: Question Database System

SI-1.1: The Maze shall retrieve the questions and answers through a programmatic interface.

SI-2: Player System

The Program will communicate with the Player System through programmatic interfaces for the following operations:

SI-2.1: To allow a Player to start the Game.

SI-2.2: To allow a Player to navigate the Maze.

SI-2.3: To allow a Player to answer trivia questions.

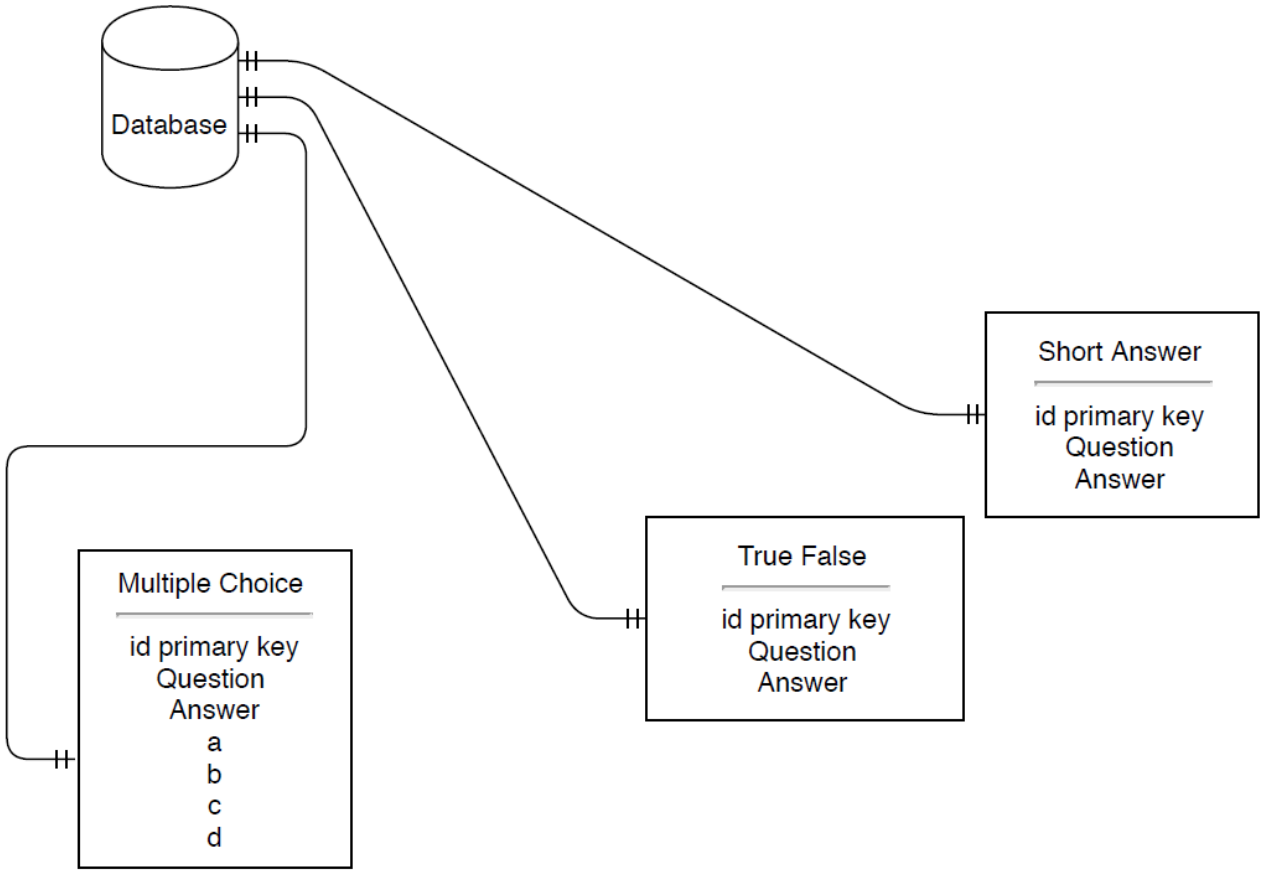
SI-2.4: To allow a Player to quit the Game.

SI-2.5: To allow a Player to save the Game.

SI-2.6: To allow a Player to load a saved Game.

SI-2.7: To allow a Player to win or lose the Game.

Figure 2:


Database
Short Answer
id primary key
Question
Answer
True False
id primary key
Question
Answer
Multiple Choice
id primary key
Question
Answer
a
b
c
d