Software Requirements Specification (SRS)

Project D Simulator

Authors: Shanruo Xu, Yuxiang Lin, Isaac Yang

Instructor: Mustafa Misir

1 Introduction

The following Software Requirements Specification (SRS) document outlines the detailed requirements and specifications for the D (Detective) Simulator software. This document serves as a comprehensive guide to the various aspects of the software, including its functionality, design, and implementation details. The SRS is organized into subsections that provide a structured overview of the software's requirements.

1.1 Purpose

The primary purpose of this SRS document is to establish a clear and complete understanding of the requirements for the D Simulator. It defines the functional and non-functional requirements, ensuring that all stakeholders involved in the software development process have a consistent and comprehensive reference. The intended audience for this document includes the development team, quality assurance personnel, project stakeholders, and anyone participating in the software development lifecycle.

1.2 Scope

The Detective Simulation encapsulates the essence of detective work, aiming to educate and entertain players while testing their analytical and critical thinking abilities. This game immerses players into the challenging role of a chief of police entrusted with the mission of unraveling a series of mysterious murder cases.
In this gripping simulation, a murderer is selected at random and, based on their specific attributes such as occupation, gender, and more, they undertake a distinct modus operandi. For example, a student-turned-killer might strike only after school hours, or risk appearing suspicious when the player investigates the school. Throughout each round of the game, players are equipped with vital information, including police records on all inhabitants, details of murder cases, and accounts from eyewitnesses. They are also presented with the intricate web of relationships between victims, potentially employing advanced query techniques. Armed with this wealth of data, players must strategically deploy

police resources such as patrols and checkpoints to apprehend the elusive culprit while minimizing disruptions to the lives of innocent citizens.

The game concludes when the player successfully captures the murderer or when they reach a point where the murderer remains elusive after a set number of rounds. The Detective Challenge Simulation replicates the tools and techniques used by real investigators, including working with relationship webs of suspects and utilizing a database to access essential residence information.

The game leverages an Entity-Relational model with SQL to effectively manage the substantial volume of data involved. While real-time aspects are not a focus, the game's turn-based nature and intricate relational mechanics make SQL databases a fitting choice. This unique blend of education and entertainment promises a captivating experience for players, making the Detective Challenge Simulation a true test of their detective skills and critical thinking abilities.

1.3 Definitions, acronyms, and abbreviations

Term	Definition
Entry Point	The entry point of the software, user is given two options, “start game” and “load game”
Right View	Panel to the right of the game interface, placeholder for either the QT/R or the IIP
Left View	Panel to the left of the game interface, placeholder for maps of various kinds
QT/R (Query Table/Result)	Query table on the right of the interface. Gives the player freedom to filter out unrelated inhabitants, the result is displayed
IIP (Inhabitant Information Page)	Page that displays information on a selected Inhabitant
Geographic Map	The map that displays the geographic map of the town in the game
QRM (Query Result Map)	displaying query results from QT/R on the Geographic map
RM (Relationship Map)	Mapping the relationship web of the inhabitant displayed on the IIP

1.4 Organization

This Software Requirements Specification document is structured into several subsections. The initial section covers the explanation of the document's Purpose, Scope, and Organization. It also addresses project-specific terminology, acronyms, and abbreviations used in the document. The second section is divided into five distinct parts, each elaborating on specific aspects of system usage and the associated actions. These sections include Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions, Dependencies, and Apportioning of Requirements. The third section provides an itemized list of all the requirements for this system. In the fourth section, you will find Use-case, Sequence, State, and Class diagrams that model the system. The fifth section presents a system Prototype, along with a visual scenario illustrating its usage. The sixth section comprises a list of all the reference materials used in this document. Finally, the seventh and last subsection is dedicated to serving as a point of contact for anyone reviewing this document.

2 Overall Description

2.1 Product Perspective

- User Interfaces
    - Entry Point
        - Options to start a new game and load from a started game
    - Save and Load
        - Player are given a page with slots to save/load a game
    - Left View
        - Geographic Map

- Map of the town, shows the geographic locations of areas in which the inhabitants reside (i.g. The School, The Hospital)
  - QRM
    - After obtaining search results from the QT/R, a QRM will be displayed, indicating the number of inhabitants that satisfy the given query in each area
    - When some area A is clicked on, a table containing inhabitants that match given query and reside in A will be displayed
  - RM
    - Page displaying all relations of selected inhabitant
    - displayed in the form of a relationship web
- Right View
  - QT/R
    - The QT gives options to filter out specific inhabitants based on characteristics (i.g. income > 1000)
    - QR returns a table containing the result of the query from QT
  - IIP
    - Page containing information on selected inhabitant, players are allowed to place selected inhabitants into a suspect list for later inspection
  - Detail Buttons
    - Access information such as a list of murders and testimonies
  - Action Buttons
    - Deploy the police force to set up roadblocks, interview eyewitnesses, and etc.

2.2 Product Functions

- User Interface
  - An interface to access data about the inhabitants in an organized manner
  - A catalog for filtering and viewing the details of the inhabitants
  - A map for viewing the information about the inhabitants on the geographic map and to zoom into the individual buildings
  - An overlay of inhabitant relationships on the map
  - Management of known information to assist deduction of the murderer
  - The option for the user to deploy the police force on the map to gather more information or to apprehend the suspect
- Simulation Mechanics
  - Turn-based simulation

        - Each turn is one day
        - Every turn the player can take actions and the world simulation advances by one day
    - Player
        - May deploy the police force each turn under the limitation of movement points
        - Under a time constraint (number of days left) to catch the murderer or be fired
        - Deploying more police resources could be done at the cost of reducing the number of days left
    - Inhabitants
        - Moves on the map daily according to their daily routine
    - Murderer
        - Randomly selected from inhabitants
        - Murders other inhabitants
        - May have some motives behind their killings and thus act according to certain patterns
        - May be constrained by their daily routine
    - Police
        - Deploy at the player’s command
        - Patrol and report dead bodies and suspicious behaviors
        - Set up checkpoints to intercept the suspect
        - Gather testimonies from eyewitnesses

2.3 User Characteristics

Anyone with a passion for detective novels; no specific limitations are placed on end users; no specific expertise, skills required.

2.4 Assumptions and Dependencies

We make the assumption that the players' laptops are generally in good working order. For Linux users, SQLite is considered a necessary dependency, while Windows and Mac users who have not installed SQLite on their laptops will find it included in the downloadable package. This inclusion ensures that the game operates smoothly with a properly functioning database system.

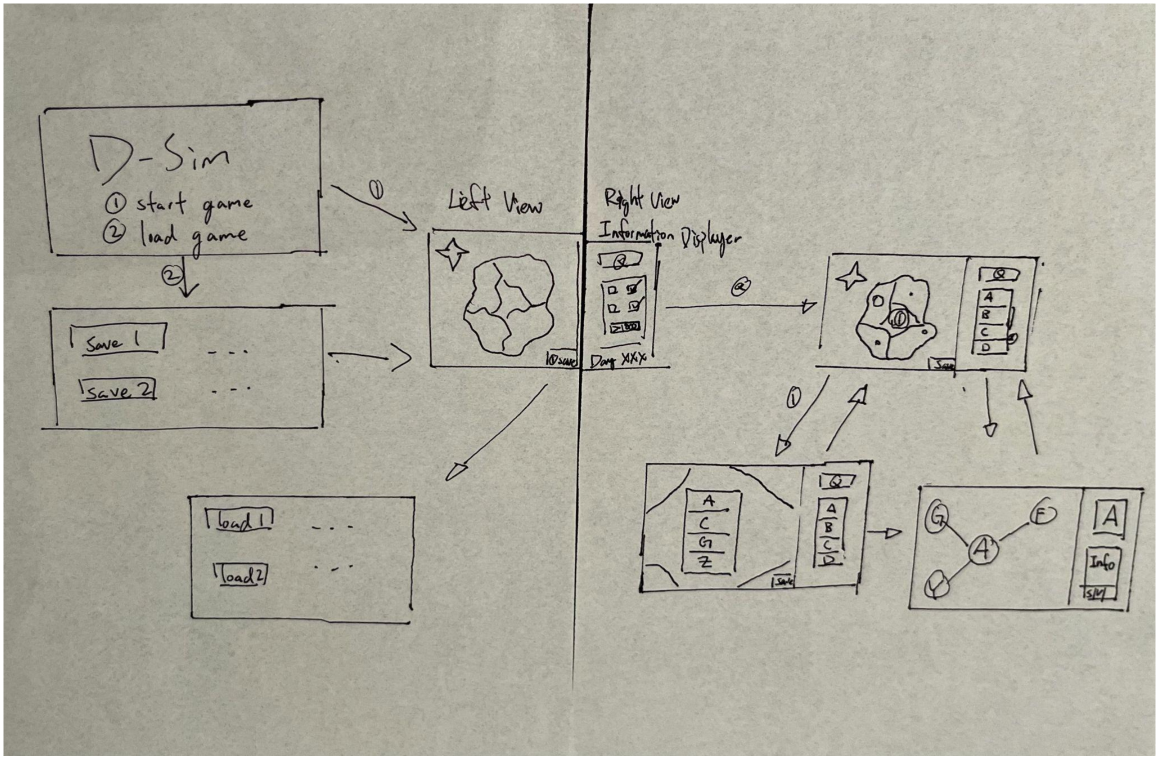
Additionally, we'd like to clarify some key assumptions: Firstly, players are unable to edit their character's information or influence their character's actions. These constraints are integral to maintaining the integrity of the game's narrative and fairness among all players.

2.5 Apportioning of Requirements

One focus of the project would be the creation of a viable database schema that can accurately depict relationships between different inhabitant entities. The other focus is to be able to come up with complex queries that present useful information in aiding the player during gameplay. Both focus, however, lies not on effective gameplay experience. This is to say that the user interface may look crude and not aesthetic, and rules may seem difficult to comprehend, all of which contributes to a poor gameplay experience. These issues are beyond the scope of the current project and could be addressed in future editions.

3 Modeling Requirements

1. **Use Case and Interface Diagram**



2. **Use Cases**

2.1. Start Game

A new game is created and the game interface is shown with the left view containing the geographical map, and the right view containing the query widget and buttons to view known information details and take actions.

2.2. Load Game

A list of saved games is shown for the player to load from.

2.3. Save Game

A list of game save slots is shown for the player to save into.

2.4. Filter Inhabitants

The play enters a list of query conditions. After running the query, the QT/R is shown on the right view and the QRM is shown over the geographic map.

2.5. View Building

A list of inhabitants in that building is shown on the left view. If inhabitants have been filtered, the inhabitants matching the filter conditions will be placed on the top of the list.

2.6. View Inhabitant

The IIP is shown on the right view and the RM originating from that inhabitant is shown on the left view.

4 References

[1] Park, J., & He, C. (n.d.). *SQL Murder mystery*. Knight Lab's SQL Murder Mystery. https://mystery.knightlab.com/

[2] Ohba, T. *Death Note*. Volume 1. Shueisha. 2003