**Experiment 01- Identifying System Requirements for an Architecture of any specific domain**

**Theory :**

The SRS should contain the following Table of Contents

**Table of Contents [i](#_gjdgxs)**

**Revision History [ii](#_ihv636)**

**1. Introduction [1](#_30j0zll)**

1.1 Purpose [1](#_1fob9te)

1.2 Document Conventions [1](#_2et92p0)

1.3 Intended Audience and Reading Suggestions [1](#_tyjcwt)

1.4 Product Scope [1](#_3dy6vkm)

1.5 References [1](#_1t3h5sf)

**2. Overall Description [2](#_4d34og8)**

2.1 Product Perspective [2](#_2s8eyo1)

2.2 Product Functions [2](#_17dp8vu)

2.3 User Classes and Characteristics [2](#_3rdcrjn)

2.4 Operating Environment [2](#_26in1rg)

2.5 Design and Implementation Constraints [2](#_lnxbz9)

2.6 User Documentation [2](#_35nkun2)

2.7 Assumptions and Dependencies [3](#_1ksv4uv)

**3. External Interface Requirements [3](#_44sinio)**

3.1 User Interfaces [3](#_2jxsxqh)

3.2 Hardware Interfaces [3](#_z337ya)

3.3 Software Interfaces [3](#_3j2qqm3)

3.4 Communications Interfaces [3](#_1y810tw)

**4. System Features [4](#_4i7ojhp)**

4.1 System Feature 1 [4](#_2xcytpi)

4.2 System Feature 2 [4](#_1ci93xb)

**5. Other Nonfunctional Requirements [4](#_3whwml4)**

5.1 Performance Requirements [4](#_2bn6wsx)

5.2 Safety Requirements [5](#_qsh70q)

5.3 Security Requirements [5](#_3as4poj)

5.4 Software Quality Attributes [5](#_1pxezwc)

5.5 Business Rules [5](#_49x2ik5)

**6. Other Requirements [5](#_2p2csry)**

**Appendix A: Glossary [5](#_147n2zr)**

**Appendix B: Analysis Models [5](#_3o7alnk)**

**Appendix C: To Be Determined List [6](#_23ckvvd)**

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to outline the software requirements for "Homicidal Hunch," a hangman game application. This document defines the scope, features, and constraints of the software.

## 1.2 Document Conventions

This document follows standard formatting conventions for software requirements. Priorities for requirements are assumed to be inherited by detailed requirements.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, and testers involved in the development of "Homicidal Hunch." Readers are advised to start with the overview sections and then proceed to the sections relevant to their roles.

## 1.4 Product Scope

"Homicidal Hunch" is a hangman game application designed for PC users. The software aims to provide an entertaining and social gaming experience, allowing users to play hangman with friends in a multiplayer mode.

## 1.5 References

No external references at this stage.

# 2. Overall Description

## 2.1 Product Perspective

"Homicidal Hunch" is a standalone application designed for PC. It offers a unique combination of the classic hangman game and multiplayer feature, providing users with a social gaming experience.

## 2.2 Product Functions

* Hangman Game: Users can play the classic hangman game, guessing words and phrases.
* Multiplayer Mode: Users can play hangman with friends in a multiplayer setting.

## 2.3 User Classes and Characteristics

* Regular Users: Players who enjoy hangman games and social interactions.
* Administrators: Responsible for managing user accounts.

## 2.4 Operating Environment

The application will run on PCs with the 4GB RAM and any other basic features.

## 2.5 Design and Implementation Constraints

* The application will be designed for PC use only.
* The user will require stable internet connection

## 2.6 User Documentation

User documentation, including manuals and tutorials, will be provided with the software.

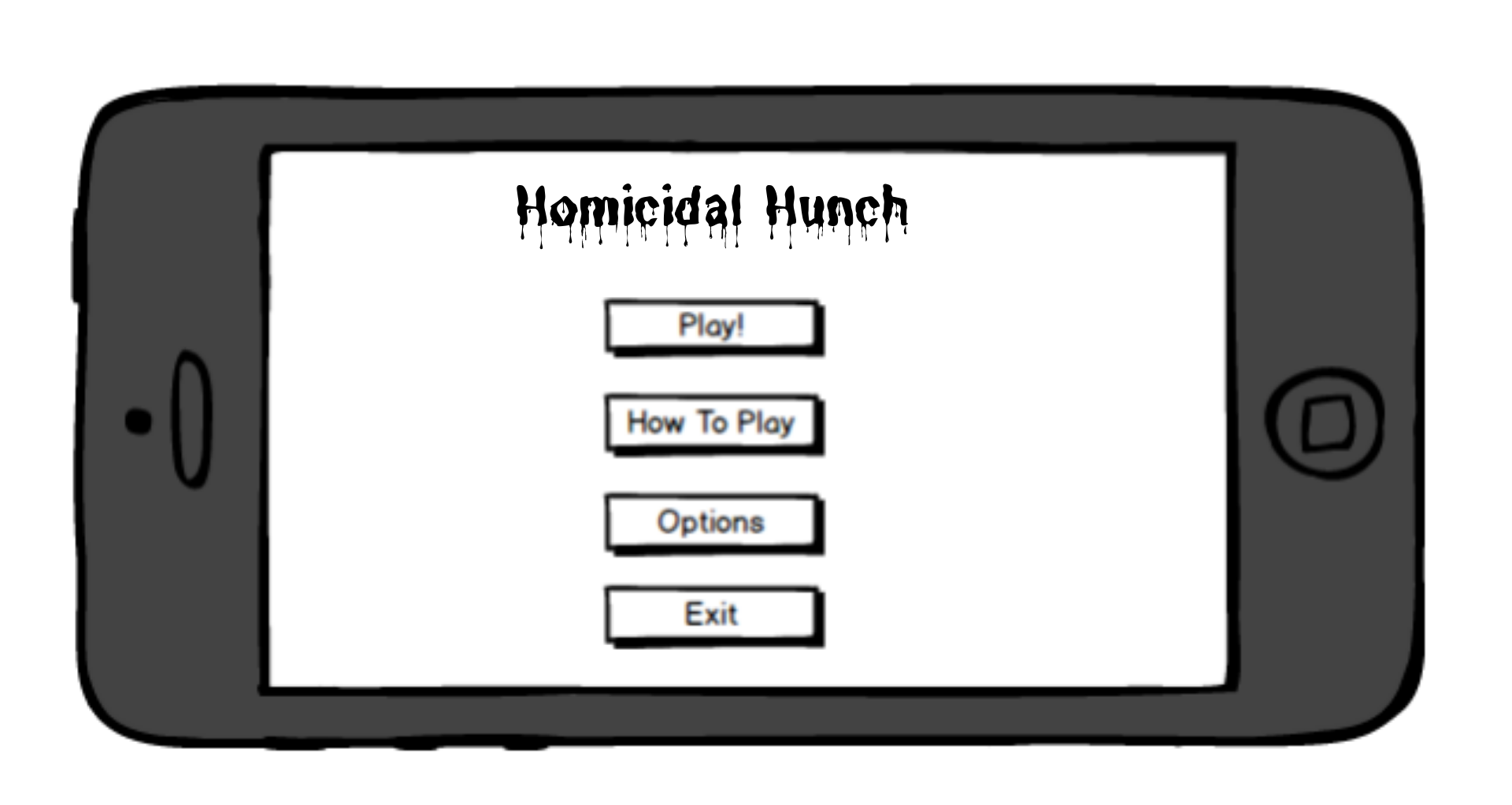
## 2.7 Assumptions and Dependencies

* Assumption: Users have a stable internet connection.
* Dependency: The successful implementation of the multiplayer feature.
* We will use Unity3D for graphics, after the testing phase, we will decide the minimum requirements and the oldest android version to be supported after which we will release it on the market.

# 3. External Interface Requirements

## 3.1 User Interfaces

* The application will have an intuitive and user-friendly interface for playing hangman.



## 3.2 Hardware Interfaces

The application will require standard PC hardware components.

It should be compatible with common input devices such as keyboard and mouse or touch.

## 3.3 Software Interfaces

Database: The application will interact with a user database for account management.

## 3.4 Communications Interfaces

The application will use standard internet communication protocols for online gameplay.

# 4. System Features

## 4.1 Hangman Game

4.1.1 Description and Priority

The hangman game allows users to guess words and phrases. Priority: High.

4.1.2 Stimulus/Response Sequences

User selects a new game → System generates a word → User makes guesses → System updates display and checks for win/loss.

4.1.3 Functional Requirements

REQ-1: The system must generate random words for the hangman game.

REQ-2: Users should be able to input guesses using the keyboard.

## 4.2 Multiplayer Mode

4.2.1 Description and Priority

Multiplayer feature for playing hangman with friends. Priority: High.

4.2.2 Stimulus/Response Sequences

* Creating a Room:

User selects the option to create a room → System generates a unique room code → System displays the room code to the user → User shares the room code with friends → Friends join the room using the code → Users play hangman together → System updates display and tracks each player's progress.

* Joining a Room:

User selects the option to join a room → User enters the room code → System verifies the room code → User joins the room → Users play hangman together → System updates display and tracks each player's progress.

4.2.3 Functional Requirements

REQ-1: The system must generate random words for the hangman game.

REQ-2: Users should be able to input guesses using the keyboard.

REQ-3: The system must support creating and joining multiplayer games.

REQ-4: Users should be able to create a room with a unique room code.

REQ-5: Users should be able to join a room using a room code.

REQ-6: The system must track and display each player's guesses and progress in real-time.

# 

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

The application should have low latency for real-time competitive gameplay..

Fast loading times for hangman games.

## 5.2 Safety Requirements

No specific safety requirements identified at this stage.

## 5.3 Security Requirements

User authentication is required to access competitive gameplay features.

Communication should be encrypted to ensure user privacy.

## 5.4 Software Quality Attributes

Usability: The interface should be intuitive for users.

Reliability: The game should function without frequent disruptions.

## 5.5 Business Rules

Users must create accounts to access competitive gameplay.

# Other Requirements

No additional requirements at this stage.

**Appendix A: Glossary**

**Hangman Game:** A word-guessing game where players attempt to guess a word by suggesting letters within a limited number of guesses.

**PC:** Personal Computer, a multi-purpose computing device designed for individual use.

**User Interface:** The graphical or textual representation of the application that allows users to interact with it.

**User Database:** A structured collection of data used to store and manage user information such as login credentials and preferences.

**Latency:** The time delay between the stimulation and response in a system, often measured in milliseconds.

**Encryption:** The process of encoding information in such a way that only authorized parties can access it.

**Usability:** The ease with which users can interact with the software to achieve their goals.

**Reliability:** The ability of the software to perform consistently and predictably under various conditions.

**Appendix B: Analysis Models**

1. DFD
2. UML
3. Class Diagram
4. Statechart Diagram

**Appendix C: To Be Determined List**

**In-Game Rewards System:** Planning and designing an in-game rewards system to incentivize user engagement and progression, including rewards for achievements or participation.

**Voice Chat Integration:** Exploring the feasibility and desirability of integrating voice chat functionality alongside text-based chat, and outlining the technical requirements for such integration.

**Cross-Platform Compatibility:** Investigating the possibility of making the application compatible across multiple platforms (e.g., PC, mobile devices) and determining the technical challenges and requirements associated with cross-platform development.

**Reference:**

1. https://www.geeksforgeeks.org/software-requirement-specification-srs-format/

**For Faculty Use**https://docs.google.com/drawings/u/0/d/sMPeBm1ly10pGVKD5V9izNg/image?w=1&h=1&rev=1&ac=1&parent=1HkCPOwvDZE1mKjYdnZTzU5QwTfuFOVtgz2SaqQNfS-k

| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** |  |
| --- | --- | --- | --- | --- |
| **Marks Obtained** |  |  |  |