Software Requirements Specification

for

Suspicious Movement

**Version 1.0 approved**

**Prepared by Angela Ku, Noah Zbinden,   
Jay Robinson, Jelaiya Jacob, Nick Moreno**

**California Baptist University**

**01.24.2019**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Design Documentation**

6.1 System Architecture

6.2 Key Components

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |

# Introduction

## Purpose

The product is an online web-based game intended to run on mobile-phones. This game is a roleplaying game where users are assigned to a team with different objectives. There are three phases to the game: the investigation phase, the marking phase, and the final battle. Throughout this process players should be able to deduce who is on their team, figure out who they need to eliminate, hopefully gaining enough advantages to shift the game in their favor in the final stage. A user should expect social interaction, physical movement, and strategy talk with other players. All prompts, instructions, and results will be displayed through the web-base on each person’s phone. This SRS encompasses the entire system under design.

## Document Conventions

There are no standards or typographical conventions in this SRS.

## Intended Audience and Reading Suggestions

This document is intended for any software developer, project manager, or website designer to review the specs for this project. The rest of this SRS contains an overall description of the product, external requirements, system features, and other nonfunctional requirements. It is recommended to read this document top-down, as introductory information is posed at the beginning.

## Product Scope

The software being specified is a web-application based game for 3-6 players. This mafia themed game consists of a twist on classic hidden role games and creates an interaction where players are at different times both sit in the same room and seek out or avoid one another in a building. During setup, players will receive a role with an objective they must complete in order to win the game (most players will be teammates). The initial phase of this game is of Investigation--players, already spread out, will look for and attempt to discover the identity of other players. Then, the Marking phase gives an opportunity to tag others to decrease their chances of winning. Lastly, the Final Battle includes a wild card that may change the outcome of the game, and presents results. The completion of this software will result in an interactive game with multiple facets.

## References

There are no references for this SRS.

# Overall Description

## Product Perspective

This game is a self-contained project that will be designed by project members. The system of the game itself will be implemented from an already developed platform (C# in Visual Studios & Additional Web-Based Implementation). The objects, shapes, and characteristics of the game will be designed by members and displayed with C#’s interface to a certain link associated with the game.

## Product Functions

Allows the user to:

* Connect to the game’s website to receive and input information into the server
* Read instructions from their phones
* Hide or open prompts regarding their “identity”
* Interact with the interface (such as tapping an object to “mark” a separate player”)
* Terminate an instance of the game
* Begin an instance of the game

Product must:

* Support multiple phones connected to its server
* Keep track of marks, player count, time passed, and rankings

## User Classes and Characteristics

Users of this product may range anywhere between 13 and older who have access to a mobile device with internet connection. Any user class should be permitted to utilize this product, as the game is about bringing into a community. There is no required security level, technical expertise, or privilege level that one must have in order to participate in this game.

## Operating Environment

The software will operate on Visual Studios on Windows. The language used to program the game will be C# and any additional components required to translate code into a visual interface. This will likely be achieved using C#’s interface qualities. The largest scope of the project includes Geofencing, which is still under further examination and research. Mobile device users must be able to access this web applications on their IOS or Android devices.

## Design and Implementation Constraints

Time is a large constraint on this project, as many details may not be ironed out before the deadline for this game. In addition to this, the project designers on this team have skills in different fields of coding, and not all are proficient in C#. A slight delay may be present while required skills are acquired and practiced. There is no budget for this project, so all components must be free of charge and last for the scope of the project (several months).

## User Documentation

Unless otherwise planned, users of this software will receive no user manuals, online help or tutorials besides the instructions presented during the game. Producers of the software may test the game with several users before release, and during that time will receive advice, comments, and opinions on what is unclear. The goal is for a stand alone and simple game, that users may play with easy access to a known website link.

## Assumptions and Dependencies

So far there are no plans to advertise this product, as it is small-scale and will be underdeveloped compared to its ideal form. This project is dependant on all designers and members showing up on time, and working together to idealize, design, code, and modify the undergoing project. Some dependencies are the language C#, Visual Studios, and a working system that can work as the server to the web-based application. Besides this, there are no further assumptions or dependencies of this project.

# External Interface Requirements

## User Interfaces

Refer to the following Balsamiq Mockup attached\*:

This will be turned in separately.

## Hardware Interfaces

The hardware interface project members will be working on are a Windows laptop computer with internet connection, Visual Studios, and a browser. No other hardware or physical material need be provided for the completion of this project. The software built will be built and run by the CPU, to which Visual Studios is a mediating interface.

## Software Interfaces

Code from Visual Studios will be run and interact with a web-based application via C# code and interfaces. Software on the computer will be working together to create a user-friendly interface on mobile devices hosted by a server on a computer. Ideally, each user will visit the website link associated with the created program and interact with an easy-to-understand screen. The Final Battle stage will include a form a communication between users, where some data (such as text messages) will be transferred between players. More details on this system will be integrated as the project moves forward.

## Communications Interfaces

Communications necessary for this project will include users talking to each other through their phones. Aside from the centralized database that all information will go to, players will need a way to communicate with one another. Players undergo a last stage where they will text each other a brief message as a wild card factor of the game. Programmers will have to set up firstly a friendly interface, and secondly create a function where inter-user communication will be possible.

# System Features

## Game Instance System

4.1.1 Description and Priority

Users must be able to create or join existing ‘rooms’ or game instances. These instances must maintain connection with and information on each player. This is high priority, as it is critical to giving players control of who they play with.

4.1.2 Stimulus/Response Sequences

When a user presses the button to make a new instance, a new instance is made and they are put in it. When a user inputs the room code and presses the join button, they are added to the corresponding instance. If there is no active instance for the code they input, they will receive an error message.

4.1.3 Functional Requirements

REQ-101: Create New Instance

REQ-102: Add User to Instance

## Assign Player Roles to Users

* + 1. Description and Priority

Each user in an active instance will be assigned a player role at random from a predefined pool based on the number of connected users. This is high priority as it is critical to game function.

* + 1. Stimulus/Response Sequences

When every connected user in an inactive instance has pressed the ready button, transition to a loading screen and assign roles, then transition to a role explanation screen.

* + 1. Functional Requirements

REQ-201: Assign a role to every user

REQ-202: Final roles adhere to game design

## Provide Directions for Players

* + 1. Description and Priority

Display text descriptions of the players role, and the actions they are able to take, as well as their objective. This is high priority as the game is not simple enough to be purely intuitive.

* + 1. Stimulus/Response Sequences

These directions will be popups displayed at the beginning of each ‘phase’ of the game, and users may minimize and reopen them.

* + 1. Functional Requirements

REQ-301: Display unique text to each player

REQ-302: Minimize and reopen dialogue box

## Read and Write Player Stats

* + 1. Description and Priority

In Phase 2 players are ‘marked’ by each other. The system must track who marked who and how many marks each player was given.

* + 1. Stimulus/Response Sequences

Players ‘mark’ each other through real-world interaction. The marked player then inputs a player id to tell the system who marked them.

* + 1. Functional Requirements

REQ-401: Keep a log of who is marked by who

REQ-402: Keep the number of marks each player has received

## Send Messages Between Players

* + 1. Description and Priority

In Phase 3, players may send a message to another role to assist in gathering information. This is a medium priority, as it is an important part of the game, but we could redesign to exclude it if needed.

* + 1. Stimulus/Response Sequences

Players would select a role, write a message and press send. After this, any received messages will be displayed.

* + 1. Functional Requirements

REQ-501: Send user-generated messages to a specific player role

# Other Nonfunctional Requirements

## Performance Requirements

The web-based application should run smoothly and timely on all mobile devices involved. As users are moving about in an inclosed area, the server must be strong enough to hold all the pieces of data together. The connection between players will not matter until the Final Stage. In the initial phase, players need only to communicate by word of mouth with each other (none of this is contingent on software). However, in the tagging phase, each phone must remain connected to a database in order to store how many “marks” each person has acquired, and if they count (based on who has tagged them). The last stage requires for mobile devices to perform well under inter-user communication, and finally synchronize results across all devices to avoid early spoilers or miscommunication.

## Safety Requirements

If geofencing were to be used, location safety features would have to be implemented for user security and privacy. However, if not implemented, there are no features that would cause any users danger or big concern. As the application is purely web-based, there will be no required downloadable content, no external software, and no embedding/saving of information or source files on one’s mobile device.

## Security Requirements

Users will be prompted for their first name (last name is optional) that everyone in the lobby may identify one another. This form of identification will be used to track scores, marks, and winning factors. Thus, a security feature may need to be added regarding what outside sources can access any instance of a game off the website. Or, users will be warned at the beginning that they should limit the amount of specific information mentioned in the game (such as to use aliases that everyone would recognize, or to go by initials instead).

## Software Quality Attributes

Quality characteristics of this application should be: reliability, speed, availability, and correctness. It is essential for this web-page to run smoothly and simultaneously on all devices for the sake of game clarity and strategy. The web-page should load within several seconds, and there should be less than a 0.5 second delay if there is to be a difference in display in mobile devices. Players should receive reliable information for the sake of gameplay and all users being on the same page. No false information should ever be provided or stored. Lastly, the web-page should be available and running for any party of people to use - depending on how it is organized, the web-page should specifically state available hours or always have an active server.

## Business Rules

N/A

# Design Documentation

## System Architecture

A game instance with a unique ID contains 3-6 players and references a Visual Studios compatible database or external server (to be determined). A player includes a character role and description, a unique 2-character ID (one letter, one number), a counter for marks, a counter for being marked, and character statistics (attack, defense, and health). The game interface displays information gathered from the instance depending on interactions between the players and the game, and updates players on the current game status. The messaging system is connected to the game instance (and database) and conveys messages between different players.

## Key Components

The game instance is instantiated by a player creating a lobby in the web-based application. The instance will have a unique ID by which other players can join the instance by entering the code on their devices. The instance will draw information from the database and display how many players are participating in the game. The database, containing a set of roles depending on present players, will distribute those roles using a shuffle method. Players will be “stored” into the database table via their unique 2-character ID, and their counter for marks, counter for being marked, and character statistics will be associated with that ID. All but the 2-character ID must be variables that will change depending on marks received or given and additional gameplay. The game interface should be available on various mobile devices provided internet connection, and display information regarding the 3 phases depending on the stage of the game and information gathered. The messaging system will either be connected by a server or via the database that players send information to.

**Appendix A: Glossary**

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**

Geofencing is a feature that may or may not be implemented, highly based on time restraint and implementation practicality and ability.