
THE



TEAM

Software Design Document

Team Member

Jason Acevedo

Vincent Agriesti

Thomas Campus

Ming Lei

Sally Ng

Table of Contents

1. Introduction	4
1.1 Purpose.....	4
1.2 Scope.....	4
1.3 Definitions, acronyms, abbreviations	4
1.4 Reference	4
1.5 Overview	4
2. System Overview	4
3. System Design	5
3.1 Architectural Design	5
3.2 Class Designs	6
3.2.1 Server Class Designs	6
3.2.2 Client Class Designs.....	7
3.3 Dynamic Models	8
3.3.1 Config	8
3.3.2 Start	8
3.3.3 Move	9
3.3.4 Suggest/ Disprove	9
3.3.5 Accuse	10
3.3.6 Chat	10
4. Component Design.....	11
4.1 User Interface	11
4.2 Client.....	11
4.3 Player.....	11
4.4 Card	11
4.5 SuspectCard	11
4.6 WeaponCard.....	11
4.7 RoomCard.....	12
4.8 Envelope	12
4.9 Location	12
4.10 Hallway	13

4.11 Room	13
4.12 GameBoard.....	14
4.13 Server	14
4.14 Suspect	14
4.15 Weapon.....	14
4.16 ClientCommand	14
4.17 Event Handler	14
4.18 ClientState	14
4.19 PlayerManger.....	14
4.20 Notebook.....	14
4.21 Dealer	14
4.22 Game	14
4.23 Message	15
4.24 MessagesEnum	15
5. Human Interface Design.....	15
5.1 Overview of User Interface.....	15
5.2 Screen Image.....	16
5.3 Screen Objects and Actions.....	16
6. Document History.....	17

1. Introduction

1.1 Purpose

The purpose of this document is specify the software design of the Clue-Less system developed by the A-Team. This document will describe the architecture and system design of the target Clue-Less system, as described in the A-Team Clue-Less Project Plan.

1.2 Scope

Clue-Less is a computerized version of the popular board game Clue. Clue-Less is being developed to allow players to enjoy the mechanics of the board game Clue in the digital world. See the A-Team Clue-Less Vision Document for more information about Clue-Less.

1.3 Definitions, acronyms, abbreviations

- SDD - Software Design Document
- SRS - Software Requirements Specification
- SSD - Supplementary Specification Document
- GUI - Graphical User Interface

1.4 Reference

This subsection provides a complete list of all documents referenced elsewhere in the Supplementary Specification.

A-Team. (2018). Clue-Less Software Requirements Specification. Unpublished document.

A-Team. (2018). Clue-Less Supplementary Specification Document. Unpublished document.

A-Team. (2018). Clue-Less Project Plan. Unpublished document.

A-Team. (2018). Clue-Less Vision Document. Unpublished document.

1.5 Overview

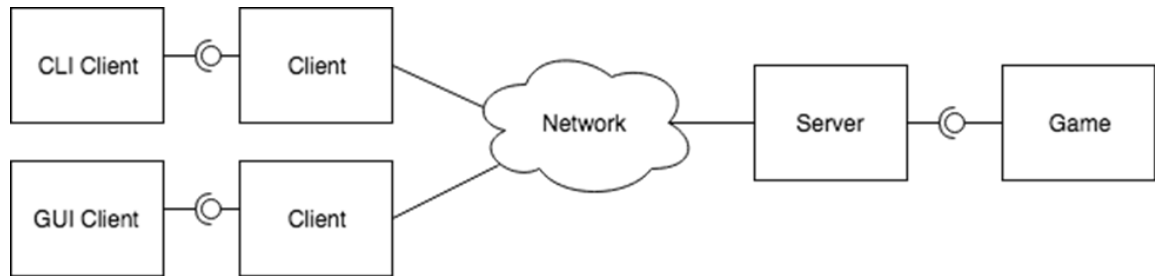
This SDD contains an overview of the system, system architecture, component design, human interface design and the requirements.

2. System Overview

Clue-Less is built using the Java programming language and incorporates a client-server model to meet the requirements as put forth in the A-Team SRS and SSD.

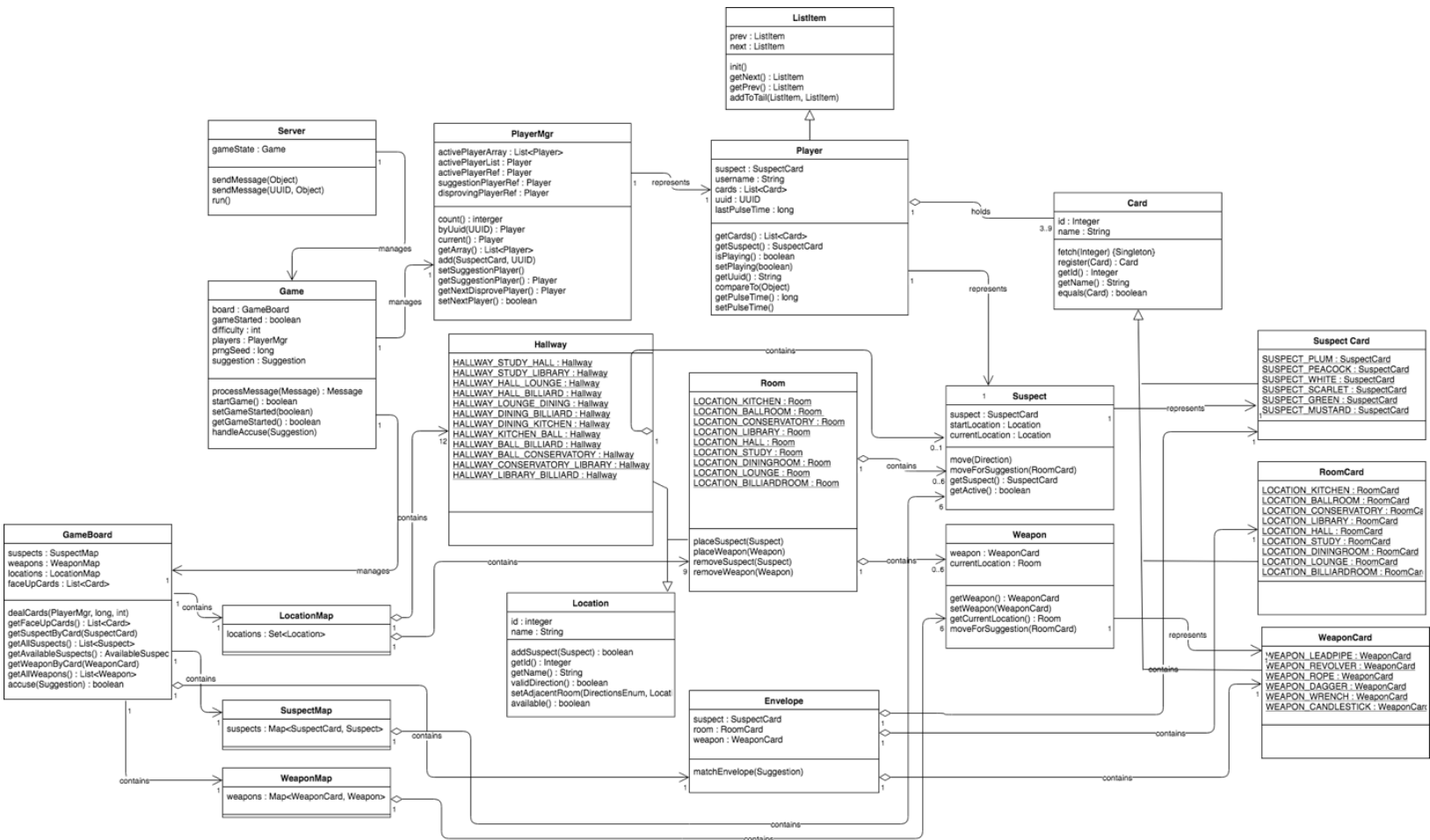
3. System Design

3.1 Architectural Design

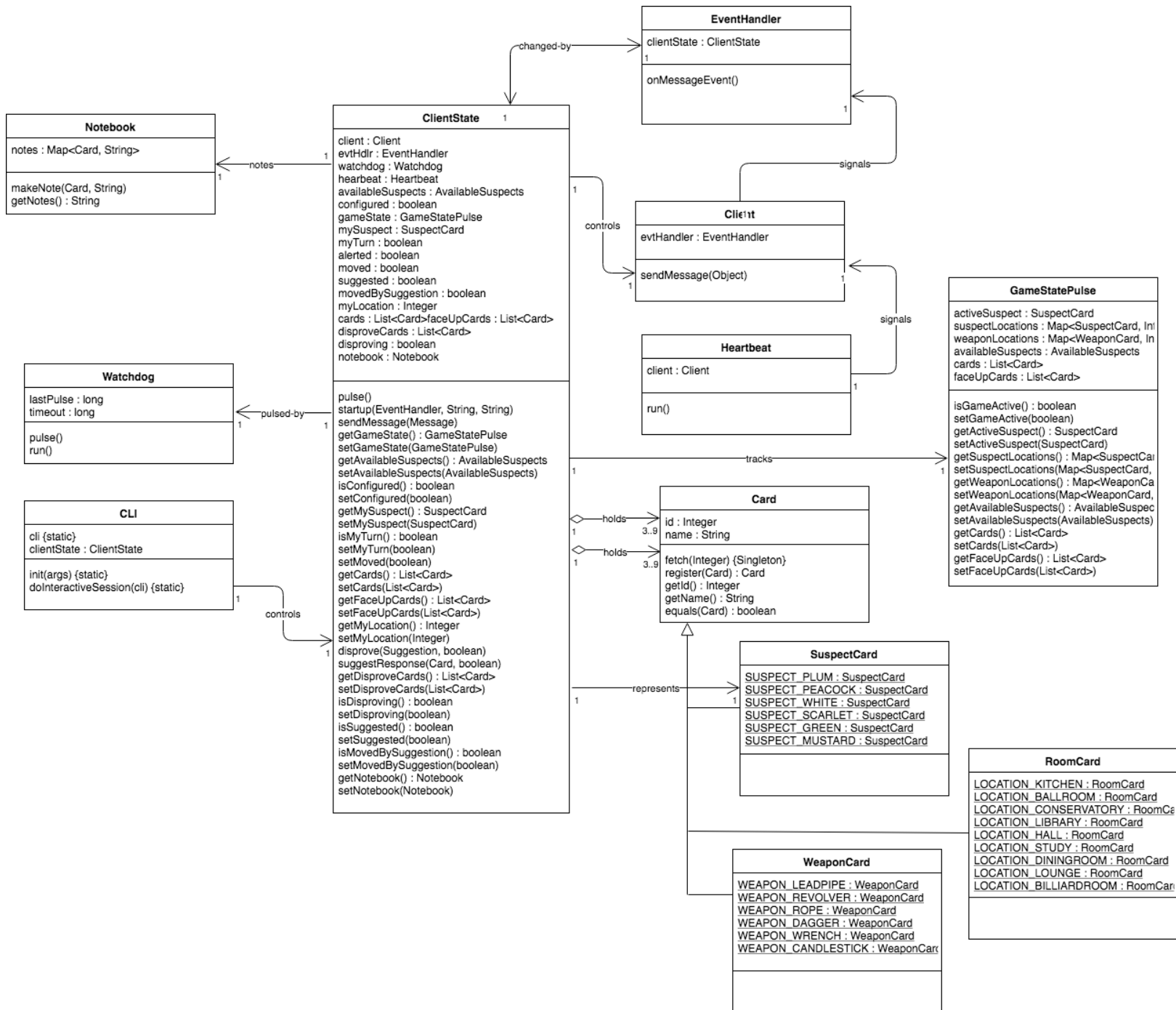


3.2 Class Designs

3.2.1 Server Class Designs

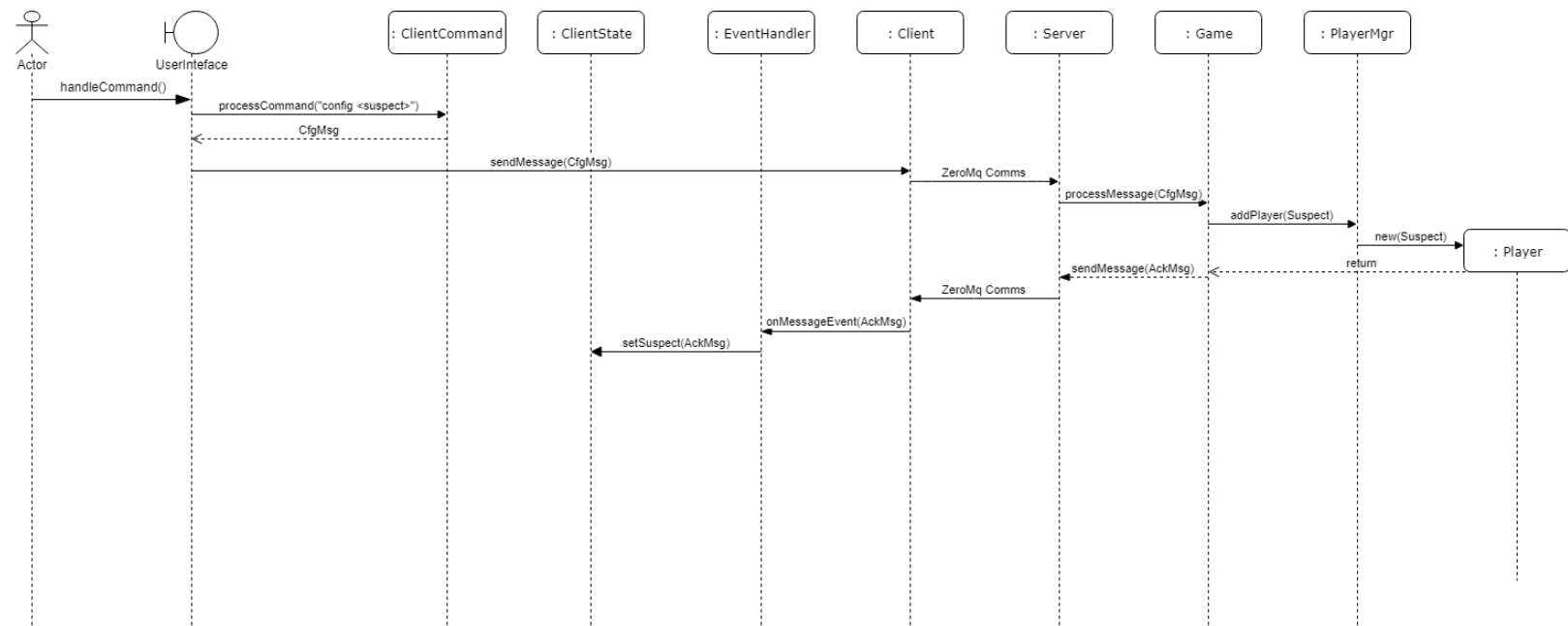


3.2.2 Client Class Designs

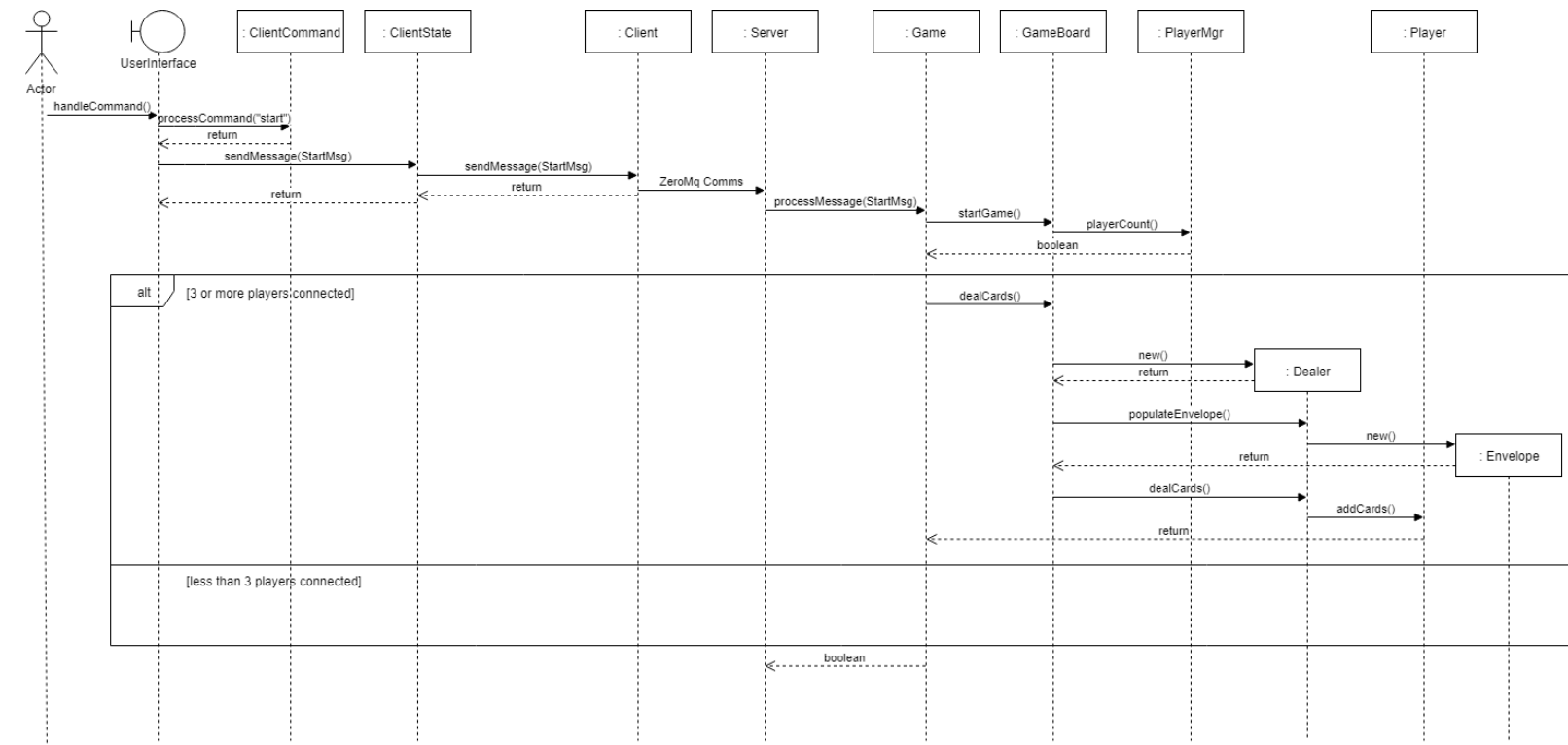


3.3 Dynamic Models

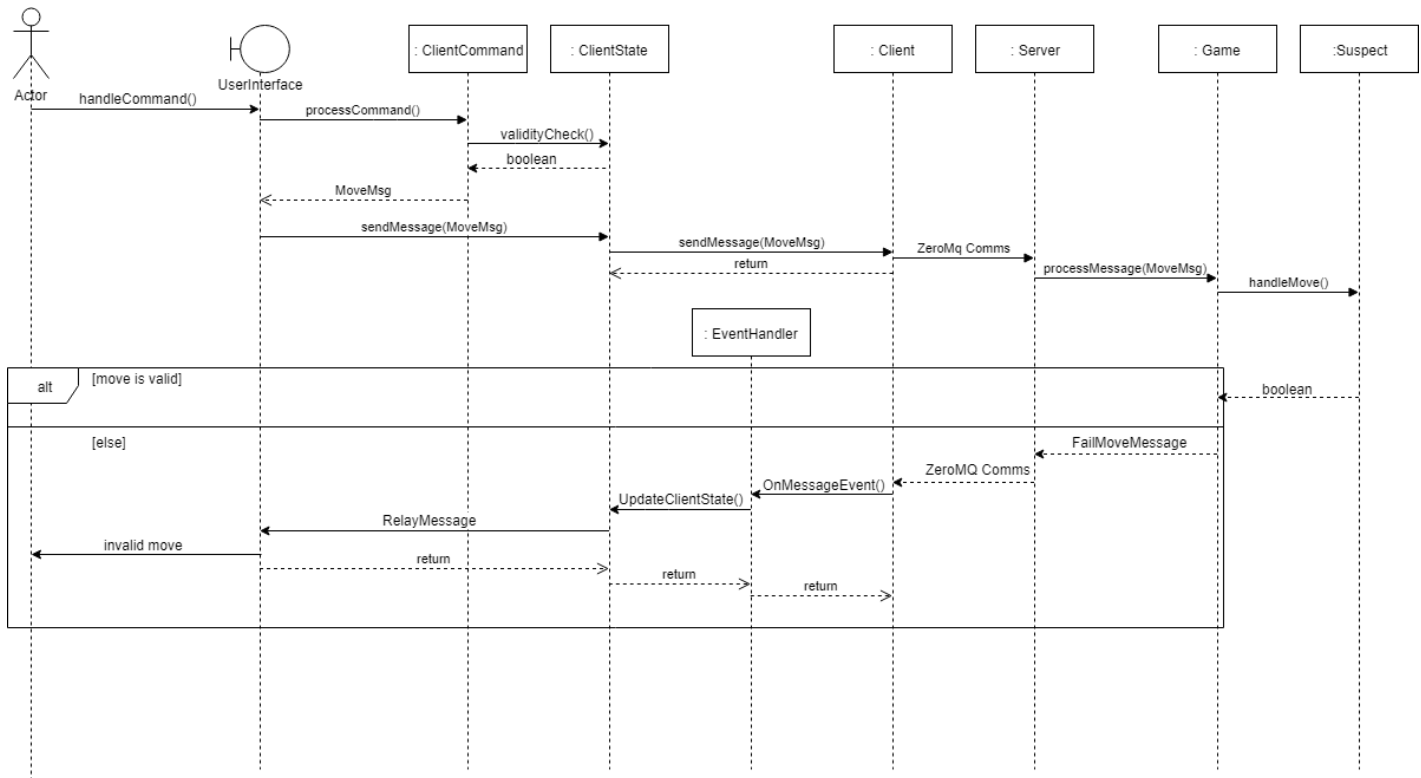
3.3.1 Config



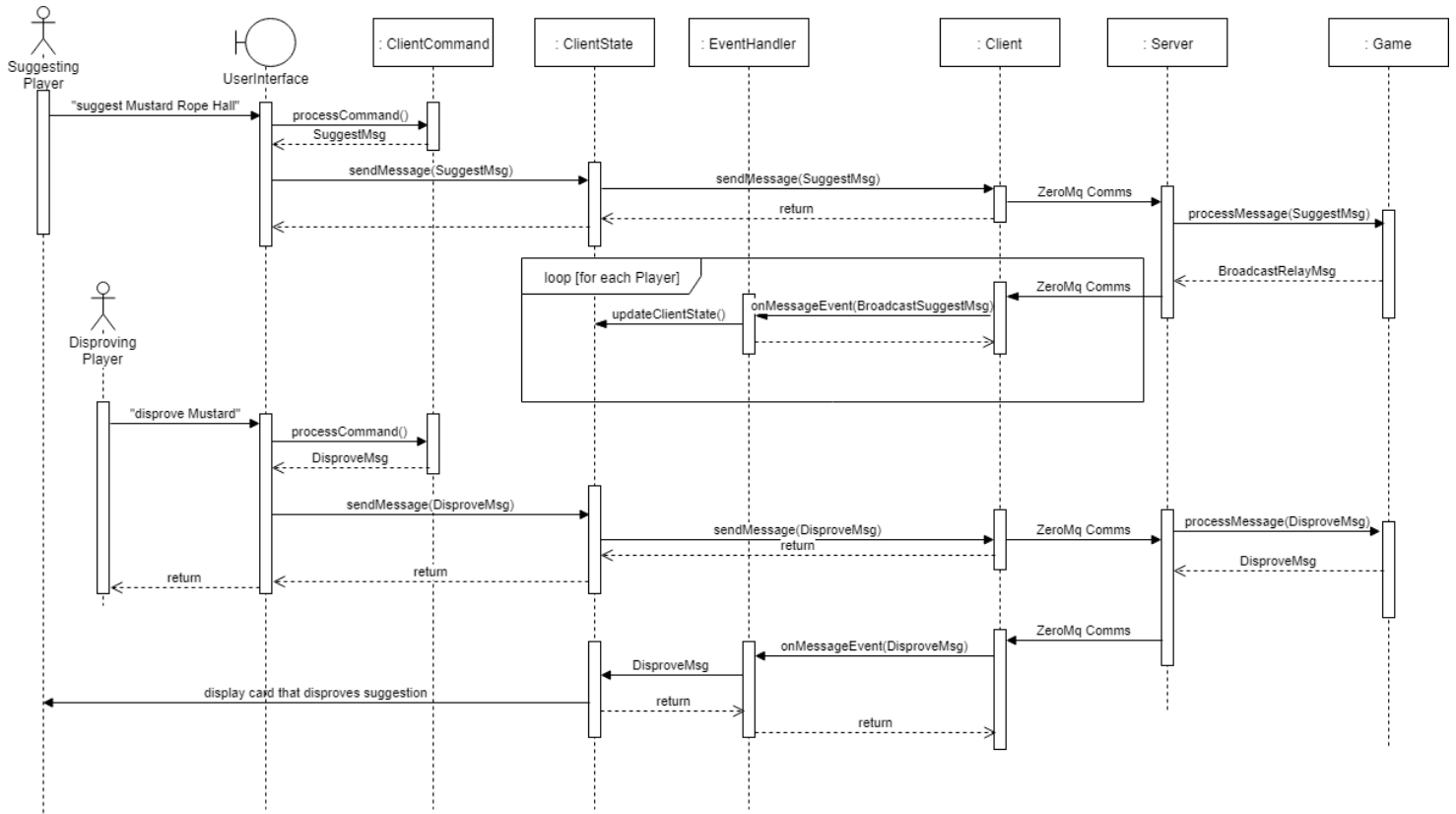
3.3.2 Start



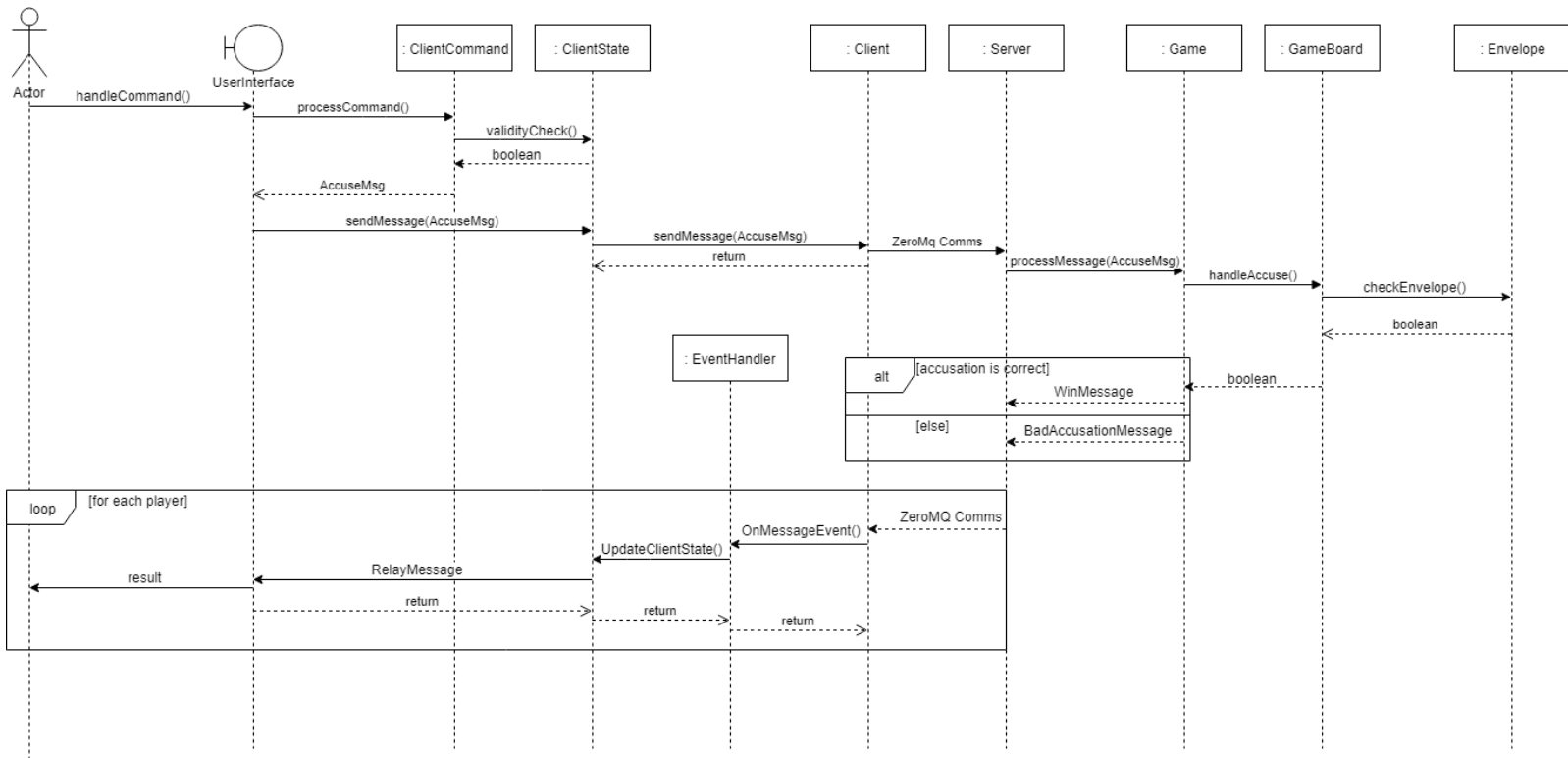
3.3.3 Move



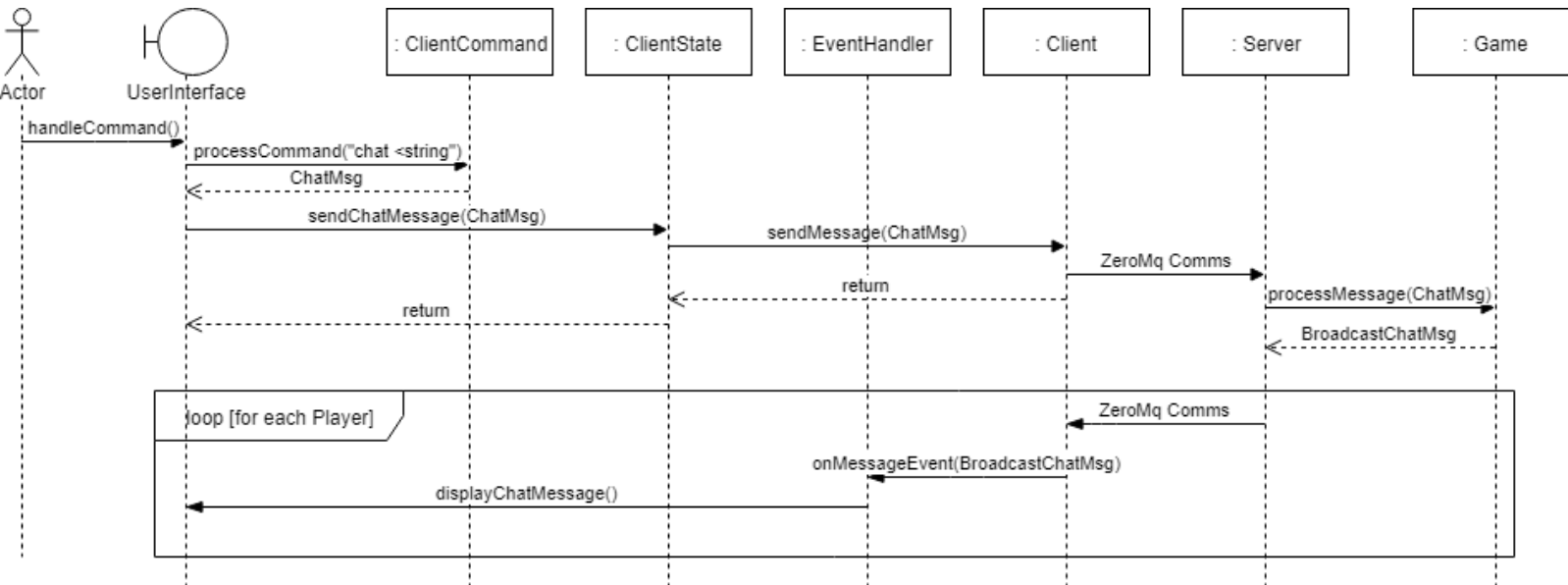
3.3.4 Suggest/ Disprove



3.3.5 Accuse



3.3.6 Chat



4. Component Design

The following classes have been identified during design.

4.1 User Interface

This class is the boundary object between the system and the player. It handles input and output from the player and to the player. It interacts with the ClientCommand and ClientState classes.

4.2 Client

This class handles the socket connection to the server, sending and receiving messages. It passes received messages to the EventHandler class. It sends messages through a zeromq socket to the server.

4.3 Player

This class maps a connected client to their suspect and cards.

4.4 Card

This class is an abstract class to represent any kind of card. It is implemented by SuspectCard, WeaponCard, and RoomCard classes.

4.5 SuspectCard

This class is an implementation of the Card class and represents the suspect cards. Its designed to behave as an enumeration class. This creates in effect a limited set of Singletons from a single class. There are 6 possible Singletons for SuspectCard:

- SUSPECT_PLUM
- SUSPECT_PEACOCK
- SUSPECT_WHITE
- SUSPECT_SCARLET
- SUSPECT_GREEN
- SUSPECT_MUSTARD

4.6 WeaponCard

This class is an implementation of the Card class and represents the weapon cards. Its designed to behave as an enumeration class. This creates in effect a limited set of Singletons from a single class.

There are 6 possible Singletons for WeaponCard:

- WEAPON_LEADPIPE
- WEAPON_REVOLVER
- WEAPON_ROPE
- WEAPON_DAGGER
- WEAPON_WRENCH
- WEAPON_CANDLESTICK

4.7 RoomCard

This class is an implementation of the Card class and represents the room cards. Its designed to behave as an enumeration class. This creates in effect a limited set of Singletons from a single class.

It should be noted that the RoomCard enums are identical to the Room enums only in name. They are different objects with different purposes. RoomCards are meant to represent a Card and be serializable. See Room description for more information.

There are 9 possible Singletons for RoomCard:

- LOCATION_KITCHEN
- LOCATION_BALLROOM
- LOCATION_CONSERVATORY
- LOCATION_LIBRARY
- LOCATION_HALL
- LOCATION_STUDY
- LOCATION_DININGROOM
- LOCATION_LOUNGE
- LOCATION_BILLIARDROOM

4.8 Envelope

This class represents the envelope which contains a RoomCard reference, a SuspectCard reference, and a WeaponCard reference. It can be populated with the three cards, and be queried as to whether a certain combination of cards completely match it.

4.9 Location

This class is an abstract class to represent any kind of location. It is implemented by Hallway and Room classes.

4.10 Hallway

This class is an implementation of the Location class and represents hallway objects. Its designed to behave as an enumeration class. This creates in effect a limited set of Singletons from a single class.

There are 12 possible Hallway Singletons:

- HALLWAY_STUDY_HALL
- HALLWAY_STUDY_LIBRARY
- HALLWAY_HALL_LOUNGE
- HALLWAY_HALL_BILLIARD
- HALLWAY_LOUNGE_DINING
- HALLWAY_DINING_BILLIARD
- HALLWAY_DINING_KITCHEN
- HALLWAY_KITCHEN_BALL
- HALLWAY_BALL_BILLIARD
- HALLWAY_BALL_CONSERVATORY
- HALLWAY_CONSERVATORY_LIBRARY
- HALLWAY_LIBRARY_BILLIARD

4.11 Room

This class is an implementation of the Location class and represents room objects. Its designed to behave as an enumeration class. This creates in effect a limited set of Singletons from a single class.

It should be noted that the RoomCard enums are identical to the Room enums only in name. They are different objects with different purposes. Rooms are meant to store adjacent Location references and what weapons and suspects are contained within. See RoomCard description for more information.

There are 9 possible Room Singletons:

- LOCATION_KITCHEN
- LOCATION_BALLROOM
- LOCATION_CONSERVATORY
- LOCATION_LIBRARY
- LOCATION_HALL
- LOCATION_STUDY
- LOCATION_DININGROOM
- LOCATION_LOUNGE
- LOCATION_BILLIARDROOM

4.12 GameBoard

This class contains all of the suspects, weapons, and locations.

4.13 Server

This class handles the socket connection to the clients, sending and receiving messages. It passes received messages to the Game class. It sends messages through a zeromq socket to one or more clients.

4.14 Suspect

This class represents a suspect and provides functionality to move the suspect between locations.

4.15 Weapon

This class represents a weapon and provides functionality to move the weapon between locations.

4.16 ClientCommand

This class parses user input into messages to send to the server.

4.17 Event Handler

This is the abstract class that the user interface implements, and provides functionality for how to handle incoming messages from the Client object that were received from the Server.

4.18 ClientState

This class manages and maintains all of the information that a client needs.

4.19 PlayerManger

This class maintains a list of all players and the play order of the players..

4.20 Notebook

This class represents the notebook that a player takes notes in.

4.21 Dealer

This class performs the action of dealing all of the cards to the players in the game, sets the necessary face up cards, and populates the envelope.

4.22 Game

This class handles messages received from the Server, performs any necessary condition checking, and provides response messages.

4.23 Message

Primary message container class.

4.24 MessagesEnum

All possible Message types that is recognized by all Client and Server implementations.

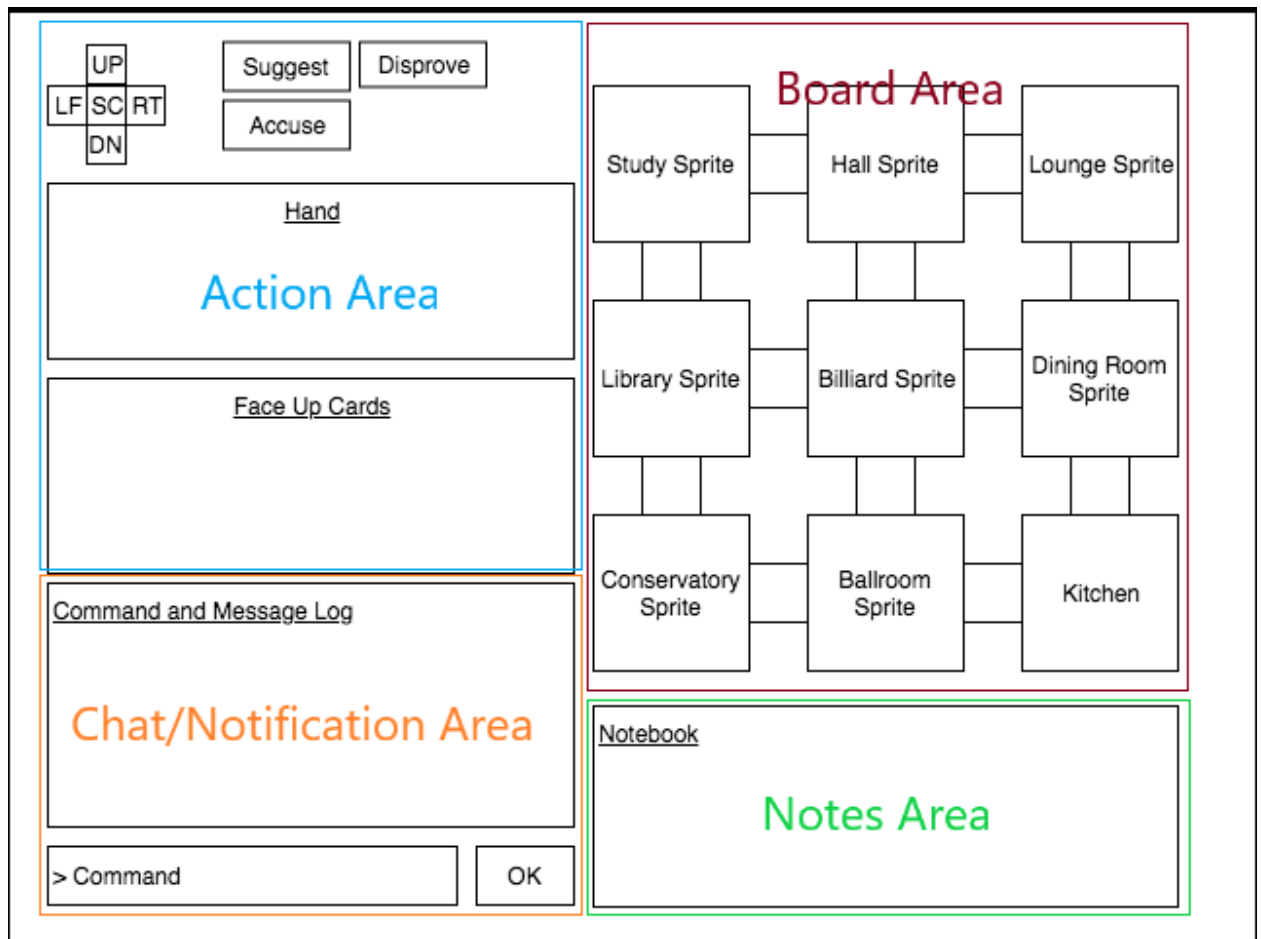
- MESSAGE_PULSE(0x50)
- MESSAGE_CLIENT_START_GAME(0x51)
- MESSAGE_CHAT_FROM_CLIENT(0x52)
- MESSAGE_CHAT_FROM_SERVER(0x53)
- MESSAGE_CARD_FROM_SERVER(0x54)
- MESSAGE_CLIENT_CONFIG(0x55)
- MESSAGE_SERVER_HEARTBEAT(0x56)
- MESSAGE_SERVER_AVAILABLE_SUSPECTS(0x57)
- MESSAGE_CLIENT_MOVE(0x58)
- MESSAGE_CLIENT_CONNECTED(0x59)
- MESSAGE_SERVER_START_GAME(0x5A)
- MESSAGE_CLIENT_SUGGEST(0x5B)
- MESSAGE_CLIENT_ACCUSE(0x5C)
- MESSAGE_SERVER_FAIL_CONFIG(0x5D)
- MESSAGE_CLIENT_END_TURN(0x5E)
- MESSAGE_SERVER_FAIL_MOVE(0x5F)
- MESSAGE_SERVER_RELAY_SUGGEST(0x60)
- MESSAGE_CLIENT_RESPONSE_SUGGEST(0x61)
- MESSAGE_SERVER_RESPONSE_SUGGEST(0x62)
- MESSAGE_INFO(0x6E)
- MESSAGE_ERROR(0x6F)
- MESSAGE_INTERNAL_SERVER_END_GAME(0x63)
- MESSAGE_INTERNAL_SERVER_REMOVE_PLAYER(0x64)

5. Human Interface Design

5.1 Overview of User Interface

The Clue-Less GUI will be written in Java and utilize gooey and JavaFX frameworks. The GUI will provide all of the functionality required to play a game of Clue-Less, and will be split in to four main sections: the chat/notification area, the board area, the action area, and the notes area.

5.2 Screen Image



5.3 Screen Objects and Actions

Chat/Notification Area - This area provides the interface to chat to other players, read other player's messages, and read any messages from the system.

Board Area - This area provides a graphical depiction of the game board, including all of the rooms and hallways, and shows the locations of suspects and weapons.

Action Area - This area provides a way for the player to perform actions such as making a suggestion, an accusation, moving their suspect, and disproving a suggestion. This area also shows the player which cards they were dealt, and which cards are shown face up to all players.

Notes Area - This area provides the interface for the player to make any notes about which cards they have been shown as a result of a suggestion, other player's suggestions, or any other information the player wishes to make note of.

6. Document History

The Design Document may be updated during the project process.

Version	Date	Notes
1	4/15/2018	Initial Release