Detonating Baby Felines

Software Design Document

[**Introduction**](#_ak9pnorwcqj) **2**

[Purpose](#_kpfav6ecp3w4) 2

[Scope](#_pe8qobbjym96) 2

[**System Overview**](#_37xomaveujmr) **3**

[**System Architecture**](#_ih7yieuxh4v6) **3**

[Web UI](#_bo21q5b0x93y) 3

[Lobby API](#_xsiaboqrsg7) 3

[Game API](#_vpminbjio19b) 4

[Game State](#_kfhigl8ad087) 4

[Decomposition Description](#_st6vg17r6lnt) 4

[Lobby API](#_ysp8kuppach9) 4

[Game API](#_l48i9odxxcp) 4

[**State**](#_l2529clp2mco) **5**

[**User Interface**](#_vcm0po4g90dk) **6**

[Lobby](#_x68lql4n8r10) 6

[Game Board](#_osnkdwbamj2) 8

# 

# 

# 

# Introduction

## Purpose

This software design document describes the architecture and system design of Detonating Baby Felines. This document will also describe the state flow of the game.

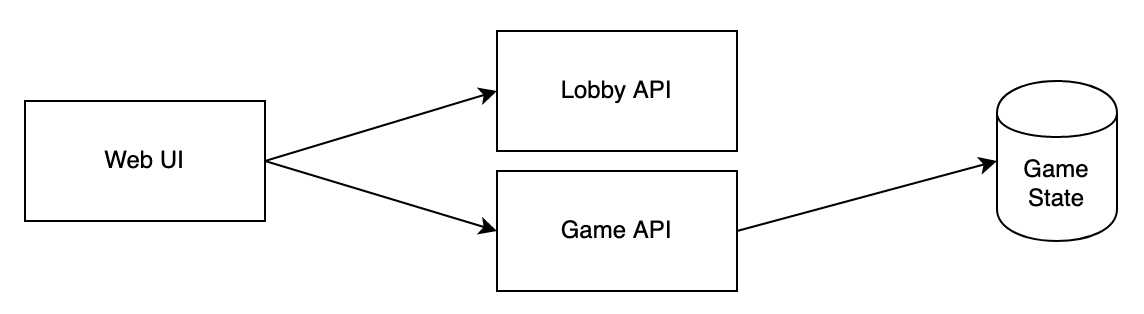
## Scope

The goals of the software are to provide a means to enjoy the Exploding Kittens card game to the computer. The game will provide multiplayer functionality with a lobby that will allow a player to create or join any game in progress.

# System Overview

A user will be able to log into the server, create a game, and join a game. The game will utilize a websocket connection between clients of an instance of the game and the server in order to maintain continuity between players. Each player will be able to play a number of cards that have various effects. A player’s turn ends when they draw a card. If that card is a specific card, that player “dies” and is no longer active.

# System Architecture



### Web UI

All user interaction occurs here. Displayed via static, generated files. Maintains a copy of game rules.

### Lobby API

Basic API that maintains a list of games and the players associated with those games. Defines valid numbers of players.

### Game API

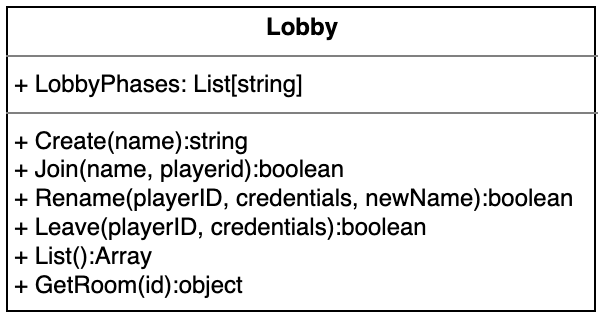
Application that stores the valid moves players are capable of making. Game state is validated here. Gameplay moves are validated here and relayed to other players as events.

### Game State

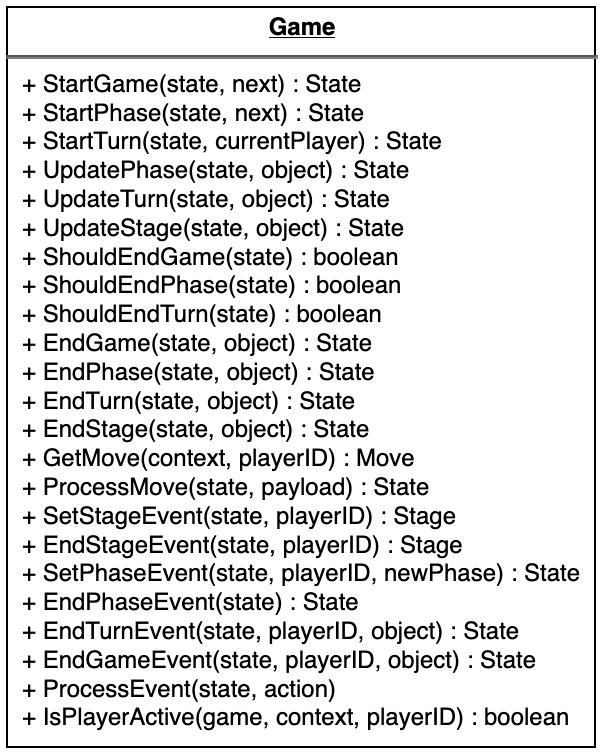
The game state is stored in a datastore along with logging information.

## Decomposition Description

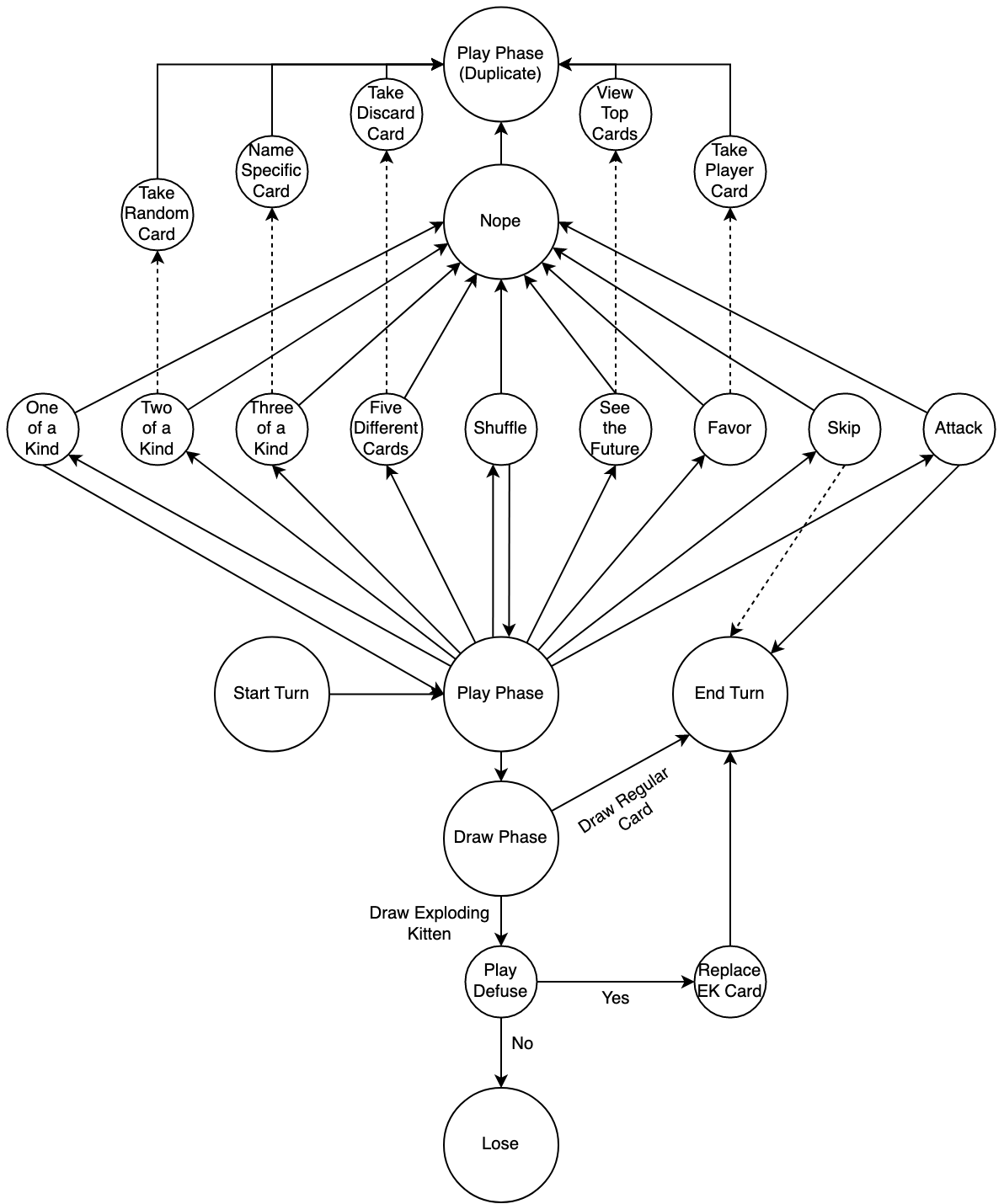
### Lobby API



### Game API

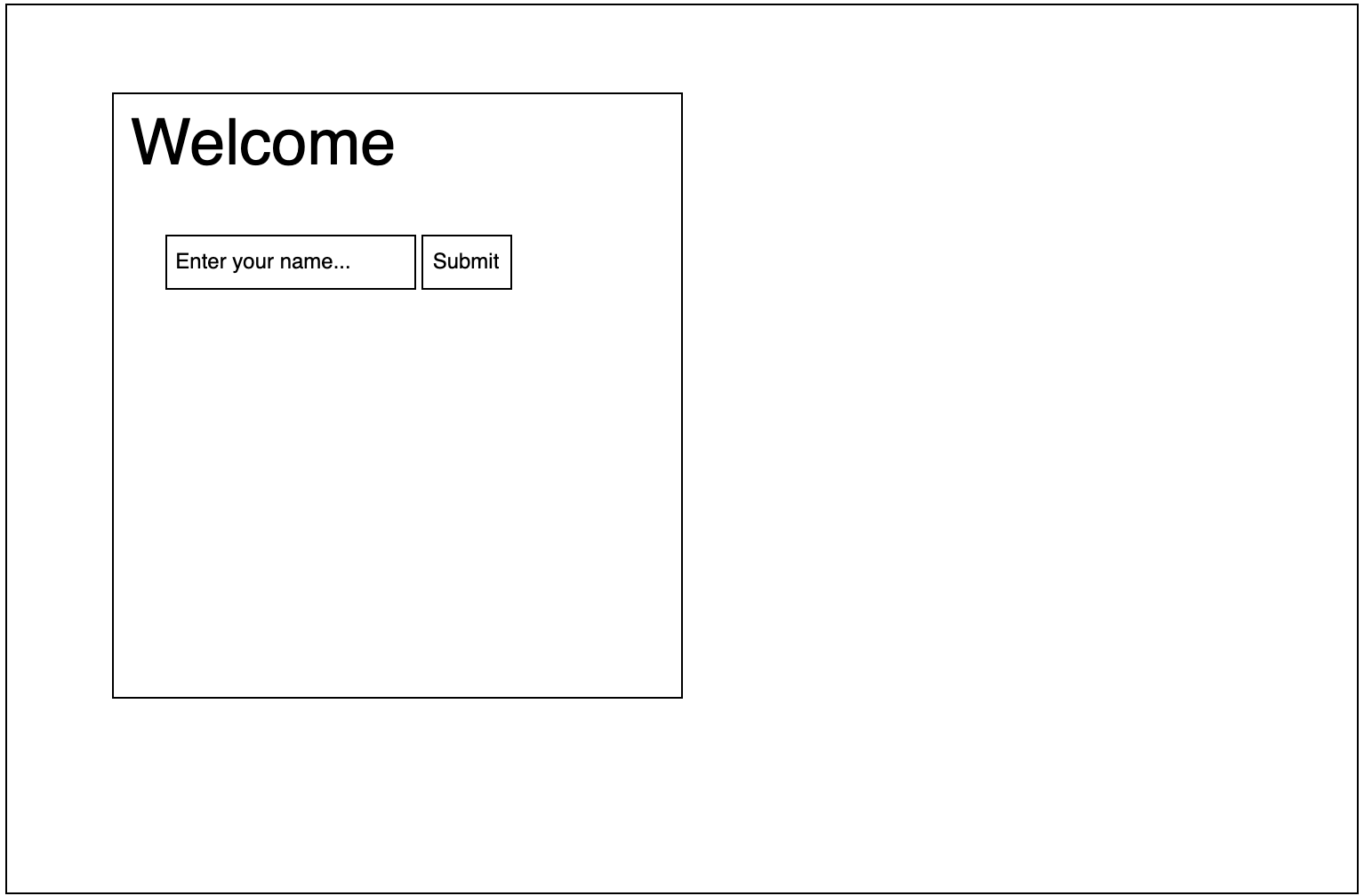


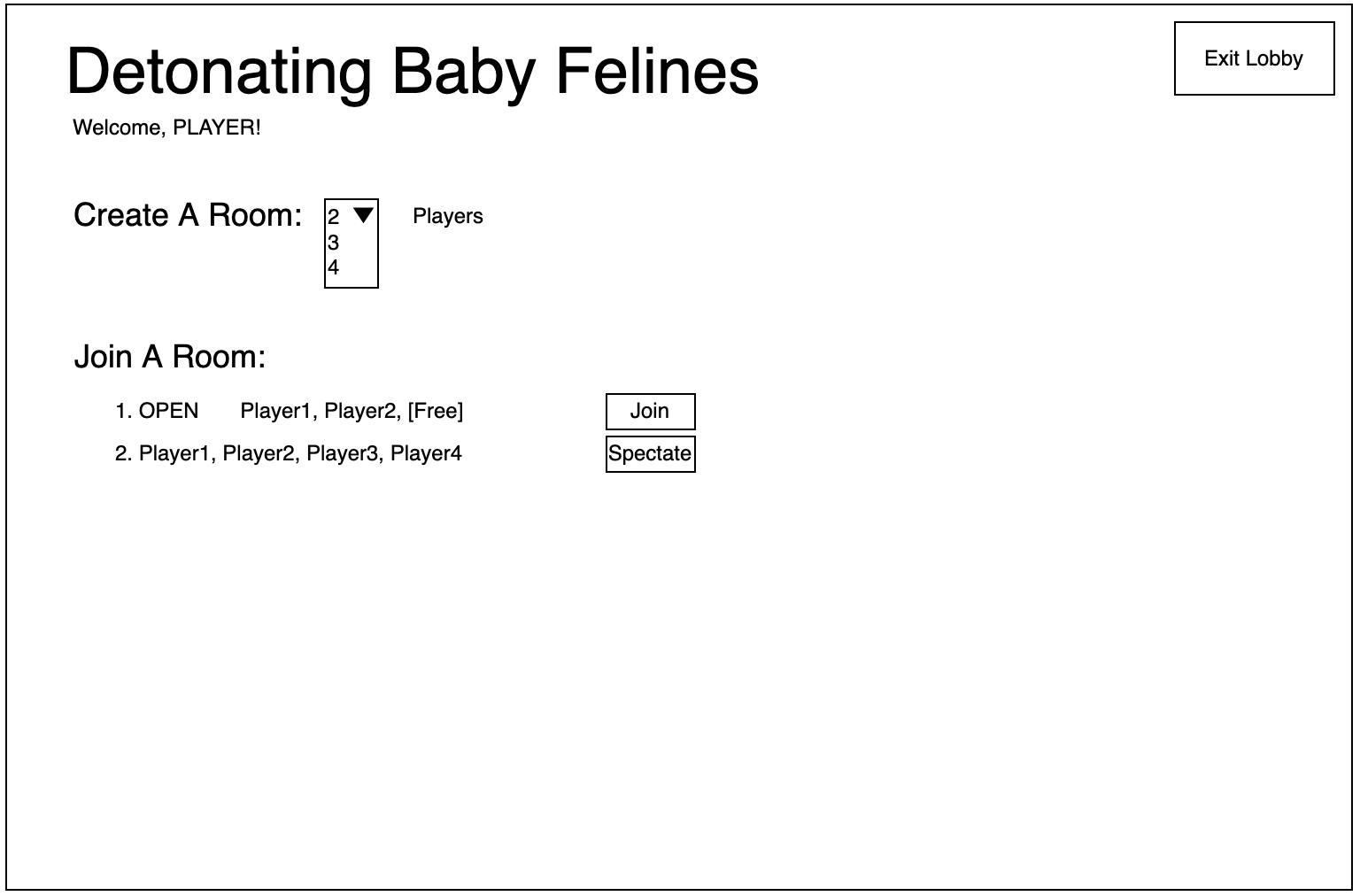
# State



# User Interface

## Lobby





## Game Board

