Avalanche

High-Level Design Document

CSCI 3100

Software Engineering @ Spring 2022

Department of Computer Science and Engineering

The Chinese University of Hong Kong

COMPLETED BY:

**GROUP B2**

1155144607 Ho Kam Hung

1155142786 Lam Shi Shing

1155149111 Ng Ching Yi

1155149067 Wong Chi To

1155156680 Wong Shing Lok

Table of Contents

[**1. INTRODUCTION**](#_3y0tb4q8a5bk) **3**

[1.1 Project Overview](#_xwkgyzpwp5wr) 3

[1.2 Background & Objective](#_a37iardyuz1z) 3

[1.3 Highlights](#_xhkox9cs8r3m) 3

[1.3.1 Overview of Game Rule](#_vmr193nj5zcy) 3

[1.3.2 Major Features](#_l6b6z1upu76x) 4

[1.3.2.1 Multiplayer Mode](#_v2dt7z7h66e4) 4

[1.3.3 Elo-based difficulty adjustments:](#_3ha8cm9al4pk) 4

[1.3.3.1 Matching System](#_hpjzt5lff7uc) 4

[1.3.3.2 Word Choosing](#_th1oxqbk7c8w) 5

[1.3.4 Items](#_mhsu3jgs3aqm) 5

[1.3.5 In-game chatroom](#_l6raih53sbl) 5

[1.3.6 User-profile / Leaderboard](#_ndew2tueb07b) 5

[1.4 Other Standard Features](#_kkv7scvnpcfo) 5

[1.4.1 User account management](#_pewvtkqayi6t) 5

[1.4.1.1 User signup](#_qi7zjj2eiqwh) 5

[1.4.1.2 User Login/Logout](#_ly9xd6gmlbh7) 6

[1.4.1.3 Reset Password](#_omxz5uyuv89q) 6

[1.4.2 Admin management portal](#_ybbrscmpcksa) 6

[1.4.2.1 List all users](#_tpjyi1tyjipu) 6

[1.4.2.3 Reset normal user password](#_khbts8b46k45) 6

[1.4.2.4 Ban user](#_efnexiisr2yy) 6

[**2. SPECIFICATION**](#_ybujc7zeqk1m) **6**

[2.1 Data flow diagram](#_y6tb4e6hbdsj) 6

[2.1.1 Login and Reset Password](#_e4k237h2a1v4) 6

[2.2 Activity diagram](#_iydku2ntj6dh) 7

[2.2.1 Game flow](#_tbl8q3j326k5) 7

[2.3 Use Case Diagram](#_rogoucgvpeta) 8

[**3. SYSTEM ARCHITECTURE**](#_jbyrawo40ae3) **8**

[3.1 Architecture Diagram](#_kjlemdcwjbh3) 8

[3.2 System Components](#_m6ffgvajv7ds) 9

[3.2.1 Front-end](#_5jepqznym15n) 9

[3.2.2 Back-end](#_uf8ykivnl19c) 9

[3.2.2.1 Web-server](#_2876agidcpqz) 9

[3.2.2.2 Database-server](#_4gixlfxjpe06) 9

# 1. INTRODUCTION

## 1.1 Project Overview

This project is an online multiplayer word game. This game is similar to the board game Mastermind and the web game Wordle. Players are challenged with their fortune and vocabulary to guess the correct word. Players will be matched to other players on the server and defeated them using special ability items. To be on the top of the leaderboard, players need to enjoy every game round and win.

## 1.2 Background & Objective

The mini-word game Wordle has gained great popularity around the globe after being purchased by the New York Times. While the NYT has been publishing word puzzles since 1942, the minimalist design of wordle has proven its effectiveness in pushing people to learn vocabulary.

Seeing the great potential of this simplistic game design, different ports of Wordle to different languages including Spanish and Cantonese were made since then.

Upon observing people sharing their daily Wordle results, we decided to push it even further. To convert Wordle from a single-player daily word puzzle mini-game to a multiplayer setting.

The goal of this project is to implement a multiplayer Wordle with elo-based features. An Elo (or ranting) system is common across many multiplayer competitive genres of games such as chess and fighting games. It serves as a difficulty adjustment mechanism and is the core part of the rewarding system for the players.

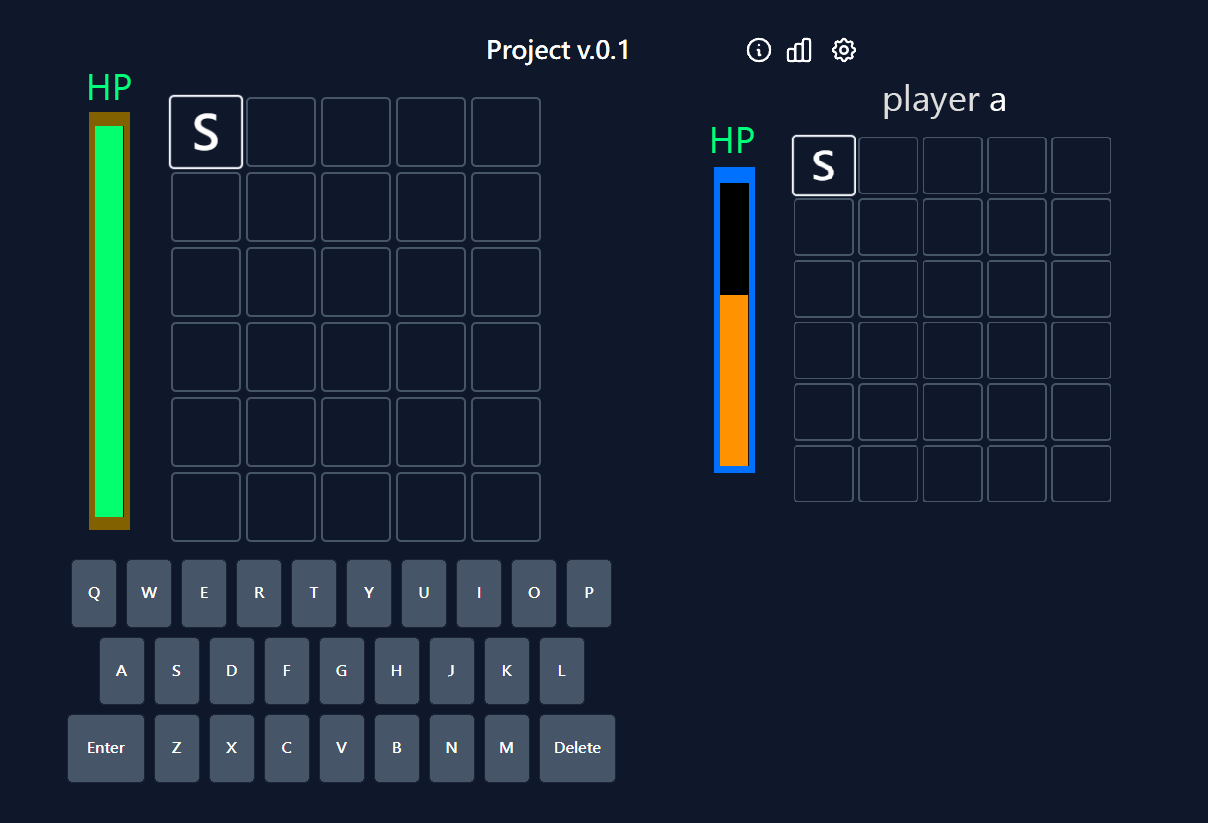
We believe that the introduction of competitive elements would encourage more players to learn English via Wordle, or simply play it for leisure, who belongs to our major target groups.

## 

## 1.3 Highlights

### 1.3.1 Overview of Game Rule

1v1 mode is introduced based on the original Wordle’s game rule. An ever-decreasing health points (HP) gauge is assigned to each player, a successful match of words would recover part of your own HP and decrease the opponent’s HP. While a mismatched attempt of 5 times would hurt yourself. The puzzle would be reset to another word after any of the aforementioned conditions. The game ends immediately when one side has his HP drop to 0.

An early iteration of the game GUI.

### 1.3.2 Major Features

The major features of our iteration include:

#### 1.3.2.1 Multiplayer Mode

1v1 (ranked / custom game) is the core of the improved game mechanism. In ranked matches, players are automatically matched with other players of similar skill levels by their Elo ratings.

Custom games (unranked) are also available to invite friends to play together with a link.

### 1.3.3 Elo-based difficulty adjustments:

The Elo system is designed for difficulty adjustments and to award the players.

#### 1.3.3.1 Matching System

An Elo algorithm is proposed to estimate the skill level of a player:  
The expected win rate of player a against player b and the Elo updating function is defined as:  




Where gamma is a constant to control how much Elo to change per match, LN(.) is the logarithm normalization, I\_result is an indicator function for the result of the match (win = 1, lose = 0).

#### 1.3.3.2 Word Choosing

Statistical methods are used for adjusting difficulty for players of different skill levels. K-means is used to produce 10 clusters for each 10% of the Elo group with respect to the difficulty (average win rate) and the rate of appearance (It is expected that the Elo follows a normal distribution, it is to avoid some words having an excessive frequency in the main group). To formalize it, we have the following objective function:

.   
Source: [Wikipedia](https://en.wikipedia.org/wiki/K-means_clustering)

A word is randomly chosen from the cluster of the average between 2 players’ ratings.

### 1.3.4 Items

Inspired by Mario Kart, items against the opponent are available to the losing side as a game balancer. Examples include screen blurring, reversing keyboard inputs or healing potion.

### 1.3.5 In-game chatroom

A live chatroom is available to enhance interaction. To prevent cheating, we may ban users to leak the answer if we detect any user trying to speak any sentence containing answers in the chatroom.

### 1.3.6 User-profile / Leaderboard

A leaderboard is available to see who is the top-performer and also your friends’ skill level. On clicking into each users’ name, ones’ username, profile-pic, and Elo are displayed on a personal profile page. These act as catalysts for community exposure.

## 1.4 Other Standard Features

Some minor standard features are also available.

### 1.4.1 User account management

#### 1.4.1.1 User signup

Allowing anyone to signup as long as their user id is unique and password is long enough. A verification email would be sent.

#### 1.4.1.2 User Login/Logout

One could login using their id and password. All passwords are hashed in the database using a private key.

#### 1.4.1.3 Reset Password

One could reset his password using a verification email.

## 1.4.2 Admin management portal

#### 1.4.2.1 List all users

An admin can list all the normal users registered, or search by names.

#### 1.4.2.3 Reset normal user password

In case a user can’t access his email but has forgotten his account, the admin could help him to manually reset his password.

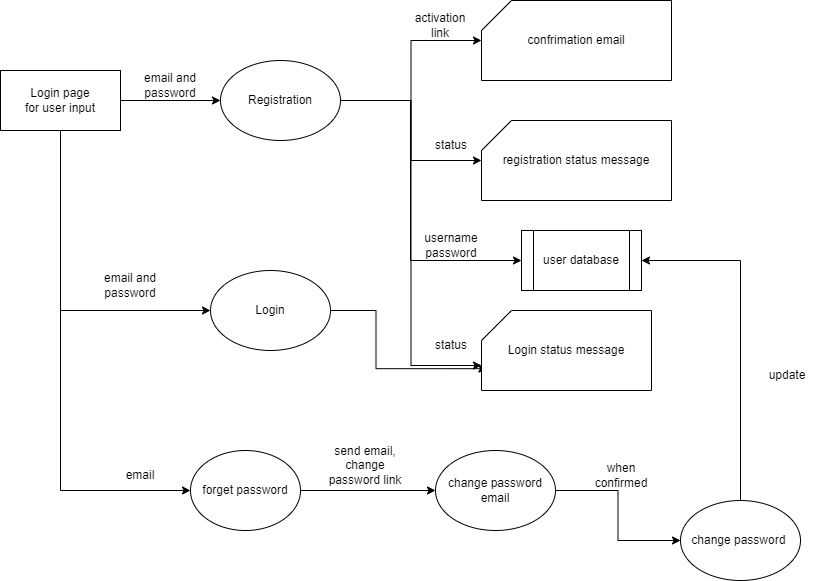
#### 1.4.2.4 Ban user

Admin can ban normal users who cheat.

# 2. SPECIFICATION

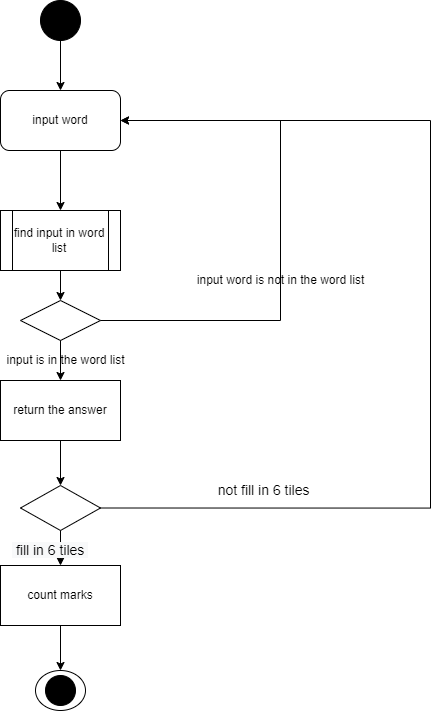
## 2.1 Data flow diagram

### 2.1.1 Login and Reset Password

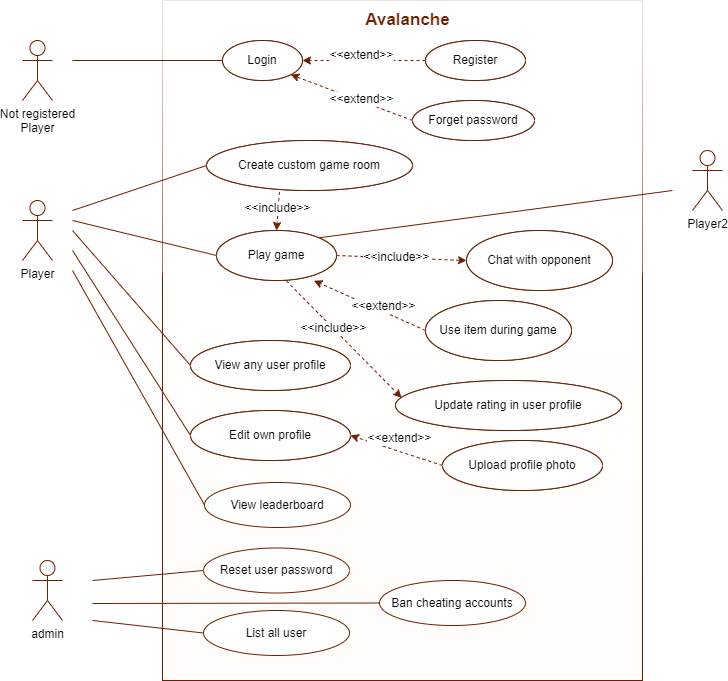


## 2.2 Activity diagram

### 2.2.1 Standard Game flow



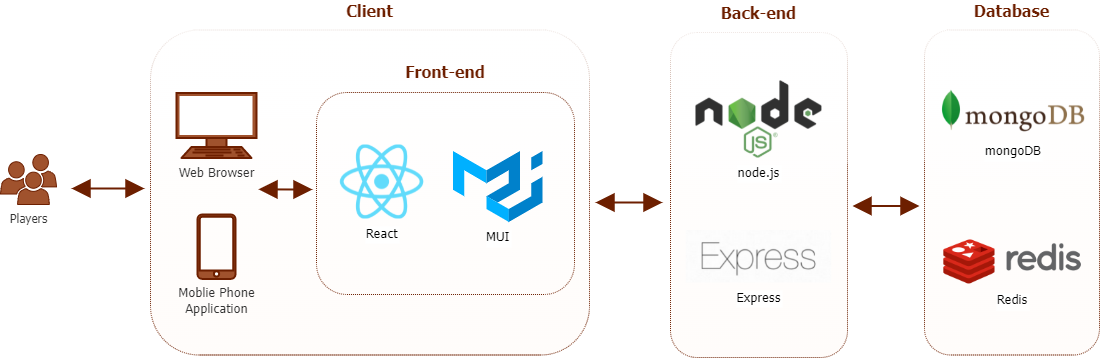
## 2.3 Use Case Diagram



# 

# 3. SYSTEM ARCHITECTURE

## 3.1 Architecture Diagram



## 3.2 System Components

### 3.2.1 Front-end

For the front-end development, we plan to use React with React UI library MUI to develop a Client-side web application. We are going to provide a responsive layout for every device and build an Android / IOS version application to provide more convenience to mobile phone users. We hope the application can provide a user-friendly, amazing, intuitive GUI to our users. The interface will also be minimalist, which focuses on creating compelling designs with less clutter.

### 3.2.2 Back-end

#### 3.2.2.1 Web-server

Express is used for HTTP request handling as well as game-controls. It's minimalist and flexible design philosophy enables fast prototyping.

#### 3.2.2.2 Database-server

NoSQL databases, are to be deployed on AWS EC2 instances.

A MongoDB instance is used for persistent data storage, namely account info, leaderboard, and so on.

While a Redis cluster, an in-memory database, is used as a cache to accelerate real-time functions. Data storage includes access tokens for current login sessions, states of ongoing matches, and buffer the chat message queue. Offset and dirty bit variables for each message along with a handshaking mechanism help ensure the synchronization of chat and game states.