System Design Document

NeverOvertime

Table of Content

Front End Classes:

Class name: HomePage	
Responsibilities: ● Provides a menu to navigate to other pages ● Show Shop Icon ● Show Play Button ● Create appealing UI ● Show user icon (avatar, name)	Collaborators: ● All other Pages (because HomePage is what the users use to navigate to other pages)

Class name: ShopPage	
Responsibilities: ● Shows the items that can be purchased, and allows the user to buy the items. ● Show bank balance	Collaborators: ● The Shop class at the back end ● Profile class (Profile stores money)

Class name: GamePage	
Responsibilities: ● Show card distributed ● Able to see user card when the round starts ● change bank balance after winning or losing	Collaborators: ● The game class at the back end ● Profile class (Money can be added to Profile when the game is being played.)

Class name: AchievementPage	
Responsibilities: ● Allow user to view finished/unfinished achievement ● Display UI design ● Allow user to claim achievement reward	Collaborators: ● The achievement class ● The GamePage class ● The Game class

Class name: ProfilePage	
Responsibilities: ● Allow users to see details of their account. ● Show Avatar ● Basic Stats ● Able to search friends' profile and their stats ● UI of user account	Collaborators: ● The profile class ● The Game class (you get tokens or money added to Profile, which will be shown on the ProfilePage)

Back End Classes:

Class name: Card	
Responsibilities: - Contains the rank and suit of a card - Key element in the game	Collaborators: The class deck (Deck contains cards)

Class name: Deck	
Responsibilities: - Contains all the cards in it - The dealer can shuffle the deck or deal cards from the deck to players	Collaborators: - Card class (Cards are contained in the deck) - Dealer class - Player class

Class name: Player	
Responsibilities: - Player is self-explanatory. The Player gets cards from the dealer and player the game - When certain conditions are met, the player wins or loses.	Collaborators: - Card class - Dealer class - Deck class

Class name: Dealer	
Responsibilities: - Deals cards to the player - In the poker game blackjack, the player play against the dealer, so when certain conditions are met, the dealer wins or loses too.	Collaborators: - Card class (Cards are contained in the deck) - Deck class - Player class

Class name: Home	
Responsibilities: Allow user to navigate to different page/screen	Collaborators: - Profile class - ProfilePage class - Game class - GamePage class - Achievement class - Shop class

Class name: Shop	
Responsibilities: ● Contains items that can be purchased in the game, and allow the users to buy them.	Collaborators: ● The Profile class (because Profile stores money/tokens) ● ShopPage class

Class name: Game	
Responsibilities: ● Able to distribute cards ● create AI ● record wins or loses ● Keep track of basic game stats and check achievement	Collaborators: ● Achievements class ● GamePage class

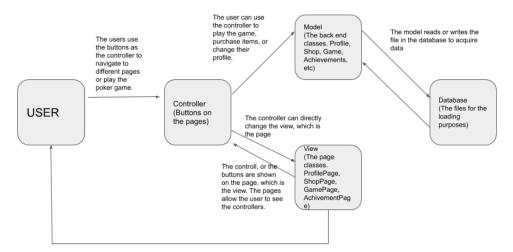| Class name: Achievement	
Responsibilities: ● Give the user the correct reward amount for its completed achievement	Collaborators: ● Game class ● AchievementPage class

Class name: Profile	
Responsibilities: ● Register users account when created ● Stores the details of user accounts. ● Secure password ● Allow users to delete their accounts. ● List of friends ● Search up friends	Collaborators: ● Game class (stats) ● ProfilePage class

Software Architecture Diagram

The users use the buttons as the controller to navigate to different pages or play the poker game.

The user can use the controller to play the game, purchase items, or change their profile.

USER

Controller
(Buttons on the pages)

Model
(The back end classes. Profile, Shop, Game, Achievements, etc)

The model reads or writes the file in the database to acquire data

Database
(The files for the loading purposes)

The controller can directly change the view, which is the page

View
(The page classes. ProfilePage, ShopPage, GamePage, AchivementPag e)

The controll, or the buttons are shown on the page, which is the view. The pages allow the user to see the controllers.

The users can view different pages of the programme to get feedback.