Use Cases

1.

1. Use case: User Registration

1.1. Brief description: Allow a new user to register to the system.

2. Actors: Primary - unknown individual, secondary - game system.

3. Precondition: unknown individual who is not yet registered.

4. Postcondition: unknown individual becomes a user.

5. Flow of Events:

5.1. Basic Flow:

5.1.1. An unknown individual requests to register.

5.1.2. USERNAME ,PASSWORD, E-MAIL AND PROFILE PICTURE:

The unknown individual enters his wanted username, password, e-mail and

profile picture.

5.1.3. CHECK USERNAME ,PASSWORD AND PROFILE PICTURE:

System checks that the username is not occupied and the password is valid,

otherwise flow continues at USERNAME AND PASSWORD once more.

5.1.4. SHOW MAIN MENU WINDOW:

Registration complete window is closing and openning the Main Menu.

5.2. Alternate Flows:

5.2.1. INPUT VALIDATION:

The unknown individual enters some characters that doesn’t meet the policy

of the system or the username or email already exists in the system.

An error message will be shown.

Flow continues to USERNAME AND PASSWORD.

2.1

1. Use case: Login

1.1 Brief description: Allows a user to login the system.

2. Actors: Primary - user, secondary - game system.

3. Precondition: User is registered in the system.

4. Postcondition: User is logged in to the system.

5. Flow of Events:

5.1 Basic Flow:

5.1.1. A user requests to log in.

5.1.2 USERNAME AND PASSWORD:

The user enters his username and password.

5.1.3 CHECK USERNAME AND PASSWORD:

System checks that the username and password match and correct, otherwise the flow continues at USERNAME AND PASSWORD once more.

5.1.4.SHOW MAIN MENU WINDOW:

The user is transferred to the Main Menu window is shown again.

5.2. Alternate Flows:

5.2.1. INVALID INPUT:

The username which was entered does not exists or the username and password do not match. An error message is shown.

Flow continues to USERNAME AND PASSWORD.

2.2

1. Use case: Logout

1.1 Brief description: Allows a logged in user to logout from the system.

2. Actors: Primary - user, secondary - game system.

3. Precondition: User doesn’t participate in an active game in the system.

4. Postcondition: User is logged out from the system.

5. Flow of Events:

5.1 Basic Flow:

5.1.1. A user requested to logout.

5.1.2 SHOW LOGIN WINDOW

The user is logged out and the system transferes to the Login User Case.

3.

1.Use case: Edit user profile

1.1 Brief description: Registered user edits his profile.

2. Actors: Primary - user, secondary - Game system.

3. Precondition: User is logged in to the system.

4. Postcondition: Profile is updated for the user.

5. Flow of Events:

5.1 Basic Flow:

5.1.1. User requests to update he’s profile.

5.1.2. UPDATE FIELDS:

The user updates the wanted fields: username, password, avater, email and money.

5.1.3 CHECK UPDATE:

The system validates the input and updates the profile if it was correct,

otherwise flow continues to UPDATE FIELDS.

5.1.4. SHOW MAIN MENU WINDOW:

The user is transferred to the Main Menu window is shown again.

5.2 Alternative Flow:

5.2.1. USERNAME TAKEN:

User tries to change his name to a username that is already taken. an error will be shown and the flow continues to UPDATE FIELDS.

5.2.2. EMAIL TAKEN:

User tries to change his email to an email that is already taken. an error will be shown and the flow continues to UPDATE FIELDS.

4.

1. Use case: Create Texas Hold’em games

1.1 Brief description: A user creates a Texas Hold’em game.

2. Actors: Primary - User, secondary – Game system, Existing game.

3. Precondition: User is logged in to the system.

4. Postcondition: A Texas Hold’em game is created and the user is joined.

5. Flow of Events:

5.1 Basic Flow:

5.1.1. CREATE GAME:

The user requests from the system to create a game.

5.1.2. SET GAME PREFERENCES:

The user chooses the game’s preferences of his liking. If the preferences not

entered correctly, an error message appears. The preferences to choose are:

-Game type policy(and to limit if needed), -Buy in policy, -Chip policy,

-Minimum bet, -Number of players(min, and max), -Can be spectated

-IsLeague.

5.1.3 UPDATE GAME:

The system creates the game with the preferences the user chose.

5.1.4. UPDATE USER PROFILE:

The System updates the game by joining the user to the game.

5.1.5 START GAME:

A Game window is open and also the Main Menu window is shown again.

5.2 Alternative Flow:

5.2.1 ILLEGAL GAME PREFERNCES:

The user chose illegal preferences to the game. An error message stating

which preference is illegal will appear. Flow continues at CREATE GAME.

5.

1. Use case: Join an existing game

1.1 Brief description: Logged in user joins existing game.

2. Actors: Primary - user, secondary - game system, existing games.

3. Precondition: User is logged in to the system.

4. Postcondition: User is playing in existing game.

5. Flow of Events:

5.1. Basic Flow:

5.1.1. User requests to join an active game.

5.1.2. SELECTING GAME:

The user selects an existing game to join.

5.1.3. VALIDATE GAME AND PLAYER STATS:

The system checks if there are any available seats in the selected game, also the system will check if the user have enough money to join the game.

5.1.4. UPDATE GAME:

The System updates the game by joining the user to the game.

5.1.5 JOIN GAME:

A Game window is open and also the Main Menu window is shown again.

5.2 Alternative Flow:

5.2.1. NOT EXISTING GAMES:

There are no existing games that are available.

A message is shown which describes that there are no existing games.

5.2.2. NOT AVAILABLE SEATS IN THE CHOSEN GAME:

There are no available seats in the selected game. A message is shown

which describes it. otherwise flow continues to SELECTING GAME.

5.2.2. NOT ENOUGH MONEY:

The user don’t have enough money to join the selected game.

A message is shown which describes it. otherwise flow continues to

SELECTING GAME.

6.

1. Use case: Spectate active game

1.1 Brief description: Logged in user spectates an existing game.

1. Actors: Primary - user, secondary - existing game, game system.
2. Precondition: User is logged in to the system.
3. Postcondition: The user is spectating the game.
4. Flow of Events:

5.1 Basic Flow:

5.1.1. User requests to spectate an active game from the system.

5.1.2. SELECTING GAME:

The user selects an existing game to spectate and joins the selected game.

5.1.3. VALIDATE GAME AND USER STATE:

The system will check that the player is not already playing or spectating the game and that the preferences of the game allows to spectate.

5.1.4. UPDATE USER PROFILE:

The System updates the game by joining the user to the game.

5.1.5 SPECTATE GAME:

A Game window is open and also the Main Menu window is shown again.

5.2 Alternate Flows:

5.2.1. NOT EXISTING GAMES:

There are no existing games that are available.

A message is shown which describe that there are no existing games.

5.2.2 CAN’T SPECTATE:

The existing game is not allowed to spectate the game. An error message is

shown and the use case jumps to SELECT GAME

7.

1. Use case: Leave a game

1.1 Brief description: A user leaves an existing game.

1. Actors: Primary - user, secondary - existing game.
2. Precondition: User is logged in to the system and participates in a game.
3. Postcondition: The user left the game.
4. Flow of Events:

5.1 Basic Flow:

5.1.1. The user requests from the game he is participating in to exit.

5.1.2. UPDATE USER PROFILE:

The game updates the system, so it would update the user profile.

5.1.3. UPDATE GAME:

The game updates the seats accordingly and release the user from the

exisiting game.

8.

1. Use case: Replay games that are no longer active

1.1 Brief description: A user asks to watch a finished game replay.

1. Actors: Primary - user, secondary - game system.
2. Precondition: User is logged in to the system, the game is over and recorded.
3. Postcondition: The user watched the replay.
4. Flow of Events:

5.1 Basic Flow:

5.1.1. The user requests to watch a replay of a finished game.

5.1.2. LOADING AND DISPLAY:

The system will fetch the replay and the user will watch it.

5.2 Alternate Flows:

5.2.1. NON FINISHED GAME:

There are no finished games.

A message is shown which describes it.

10.

1. Use case: Find all active games which the user can join to.

1.1 Brief description: A user will search for games which has at least one available

seat.

2. Actors: Primary - user. secondary - Game system.

3. Precondition: User is logged in to the system.

4. Postcondition: A list of all active games that the user can join to is shown to the user.

5. Flow of events:

5.1 Basic Flow:

5.1.1. The user requests for a list of all active games which he can join.

5.1.1. DISPLAY:

A list of all available active games is shown to the user.

5.2 Alternative Flow:

5.2.1. ZERO GAMES FOUND:

There is not even a single available game which the user can join,

a message shown and notify the user that there are no games

found.

11.

1. Use case: Filter active games by criteria

1.1 Brief description: A user can filter for active games by criteria (player name,

pot size or game preferences).

2. Actors: Primary - user. secondary - Game system.

3. Precondition: User is logged in.

4. Postcondition: A list of all active games that matches the criteria is shown to the user.

5. Flow of events:

5.1 Basic Flow:

5.1.1. SEARCH CRITERIA

The user requests for a list of all active games by criteria which he can join.

5.1.1. DISPLAY:

A list of all available active games is shown to the user and he can choose to

join or spectate a game from that list.

5.2 Alternative Flow:

5.2.1. ZERO GAMES FOUND:

There is not even a single available game which matches the criteria,

a message shown and notify the user that there are no games found and the user case jumps back to SEARCH CRITERIA.

12.

1.Use case: Maintain leagues:

1.1 Brief description: managing which users are in which league at any given

moment.

2. Actors: Primary - Server, secondary - system.

3.Preconcition: The last update was last week.

4. Postcondition: All the are users in leagues related to their experience.

5. Flow of Events:

5.1 Basic Flow:

5.1.1. DIVIDE TO LEAGUES:

The system will compute the number of leagues as a factor of the number of

users, and will divide the users according their rank in a each league will

be in the same size +-1.

A new user is not related to any given league for his first 10 games.

5.2 Alternative Flow:

5.2.1 NOT ENOUGH USERS:

The system modify less than 2 users in a league.

13.

1. Use case: Bet.

1.1 Brief description: A player bet in a game

2. Actors: Primary - player , Secondery - existing game.

3. Precondition: The game is an existing game, it’s the player turn, he has the minimum

chips to bet and no one placed a bet before.

4. Postcondition: The player placed a bet.

5. Flow of Events:

5.1. Basic Flow:

5.1.1. The player request to place a bet.

5.1.2. CHOOSE AMOUNT

The player chooses the amount he wants to bet.

5.1.3. POT UPDATE

The existing game updates the player state and the pot so other

players will know the new bet.

5.2 Alternative Flow:

5.2.1 NO CHIPS

The player chooses to place a bet higher than the chips he have.

A message is shown which describes it.

Flow continues to CHOOSE AMOUNT.

5.3.1 ALREADY BET

The player already placed a bet in this round.

A message is shown which describes it.

14.

1. Use case: Call.

1.1 Brief description: A player calls the bet in a game

2. Actors: Primary - player , Secondery - existing game.

3. Precondition: The game is an existing game, it’s the player turn and the player need to

equalize the bet or big-blind.

4. Postcondition: The player calls the current bet.

5. Flow of Events:

5.1. Basic Flow:

5.1.1. The player request to call.

5.1.2. POT UPDATE

The existing game updates the player state and the pot so other

players will know the new pot amount.

5.2 Alternative Flow:

5.2.1 LOW CHIPS

The player doesnt have enough money to call.

The player make all-in and a new pot is generated to the rest of the

players.

15.

1. Use case: Fold.

1.1 Brief description: A player folds during a round.

2. Actors: Primary - player , Secondery - existing game.

3. Precondition: The game is an existing game and it’s the player turn.

4. Postcondition: The player folded.

5. Flow of Events:

5.1. Basic Flow:

5.1.1. The player requests to fold.

5.1.2. PLAYER UPDATE

The existing game updates the player's state.

16.

1. Use case: Check.

1.1 Brief description: A player decides to NOP during a round.

2. Actors: Primary - player , Secondery - existing game.

3. Precondition: The game is an existing game and it’s the player turn.

4. Postcondition: The player checks.

5. Flow of Events:

5.1. Basic Flow:

5.1.1. The player request to check.

5.1.2. POT UPDATE

The existing game updates the pot so other players will know the

new pot amount.

5.2 Alternative Flow:

5.2.1 BET EXISTS

One of the players already put a bet.

A message is shown which describes it.

17.

1. Use case: Placing blind bets for big and small blind.

1.1 Brief description: To start a round two players put a small and big blinds.

2. Actors: Primary - Small blind, big blind, Secondery - existing game.

3. Precondition: The game is an existing game.

4. Postcondition: The big and small blind putted chips in the pot.

5. Flow of Events:

5.1. Basic Flow:

5.1.1. The exitsting game asks from the small blind to put his chips.

5.1.2. SMALL BLIND

The small blind player put the needed chips.

5.1.3. The exitsting game asks from the big blind to put his chips.

5.1.4. BIG BLIND

The big blind player put the needed chips.

5.2 Alternative Flow:

5.2.1 NOT ENOUGH

One of the players dont have enough chips to be a blind.

The player makes all in and a new pot is generated to the rest of the

players.

18.

1. Use case: Support playing a Texas Hold'em game.

1.1 Brief description: users can play Texas Hold’em game with all it’s rules and it’s

steps.

2. Actors: Primary - user, Secondery - existing game.

3. Precondition: There are between 2 to 9 players that joined the game.

4. Postcondition: None.

5. Flow of Events:

5.1. Basic Flow:

5.1.0. Prepare player state for all players

5.1.1. Dealer Deals cards to all of the players playing the game.

5.1.2. UC Placing blind bets for big and small blind.

5.1.3. FIRST BET AFTER BLINDS:

Player left to the player who bet big blind choose between UC Bet, Fold

and Check.

5.1.4. REST PLAYERS PLAY:

Rest of players can choose between UC Call, Fold and Check

according to game rules. AF PLAYERS FOLD, PLAYERS LEFT THE

GAME EXCEPT ONE and PLAYERS LEFT THE GAME can occur.

5.1.5. FIRST REVEAL:

The first three cards are revealed.

5.1.6. FIRST BET:

The player left to the dealer choose between UC` Bet, Fold

and Check.

5.1.7. REST PLAYERS BET

5.1.8. SECOND REVEAL:

The turn card is revealed

5.1.9. FIRST BET

5.1.10. REST PLAYERS BET

5.1.11. THIRD REVEAL:

The river card is revealed

5.1.12. FIRST BET

5.1.13. REST PLAYERS BET

5.1.14. CHECK HIGHEST HAND:

The player with the highest hand according to game rules collect all

pot money.

5.2 Alternative Flow:

5.2.1 PLAYERS FOLD

All the players fold except one player that takes all the bets money.

5.2.2 PLAYERS LEFT THE GAME EXCEPT ONE

All the players left the game except one.

The remaining player takes all the bets money and waits to other

player(s) to join the game.

5.2.3 PLAYERS LEFT THE GAME

All the players left the game. The game ends.

20..

1. Use case: A Player sends a message

1.1 Brief description: All users in the game my publish messages to a single message stream.

2. Actors: Primary: Actor, Secondary: system.

3. Precondition: User is logged in and in a chat.

4. Postcondition: User recieved a message.

5. Flow of events:

5.1 Basic Flow:

5.1.1 SEND MESSAGE: The player sends in the chat. If the messages

cannot be sent to the user an error will presented to the user.

21.

1. Use case: A Spectator sends a message

1.1 Brief description: All users in the game my publish messages to a single message stream.

2. Actors: Primary: Actor, Secondary: system.

3. Precondition: User is logged in and in a chat.

4. Postcondition: User recieved a message.

5. Flow of events:

5.1 Basic Flow:

5.1.1 SEND MESSAGE: Spectator sends the message. If the messages

cannot be sent to the user an error will presented to the user. The message

is shown only to other spectators.

5.2 Alternative Flow:

5.2.1 INVALID RECIEVER: The spectator can’t send message to players.