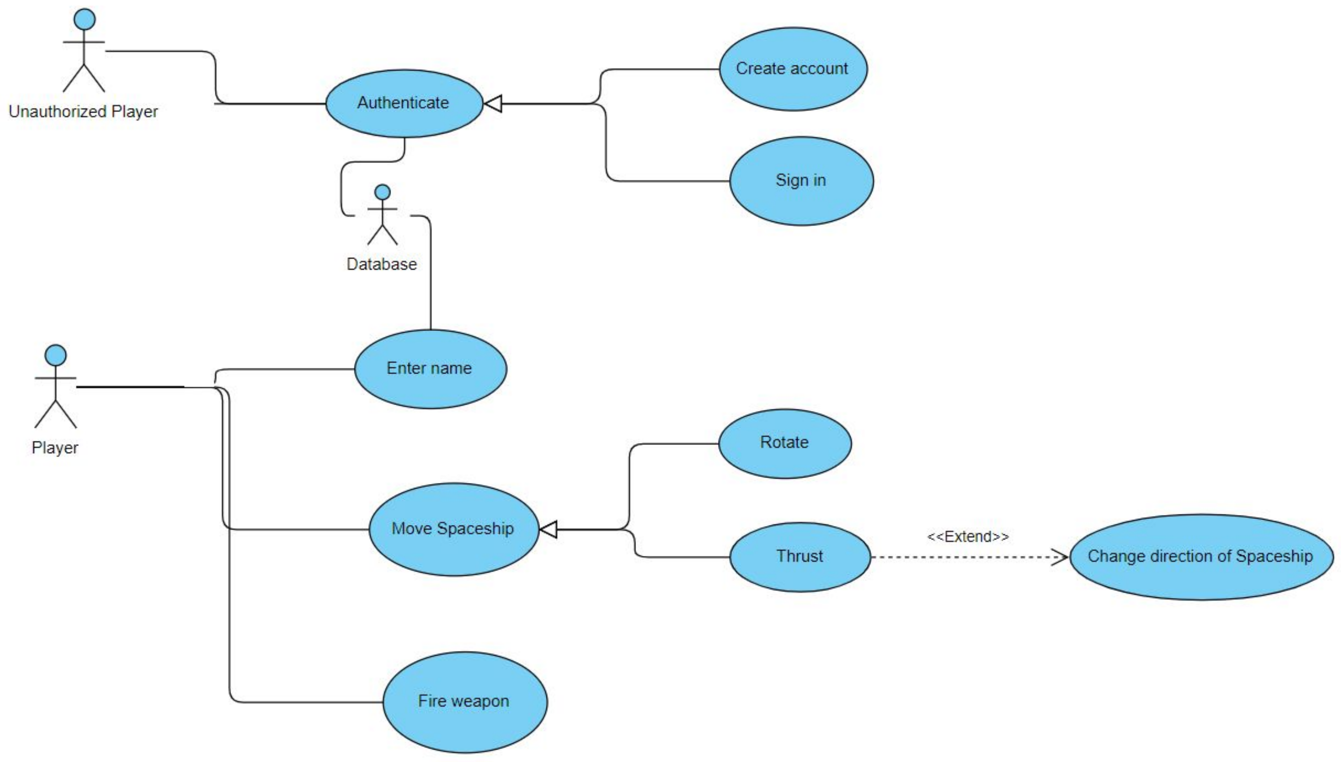
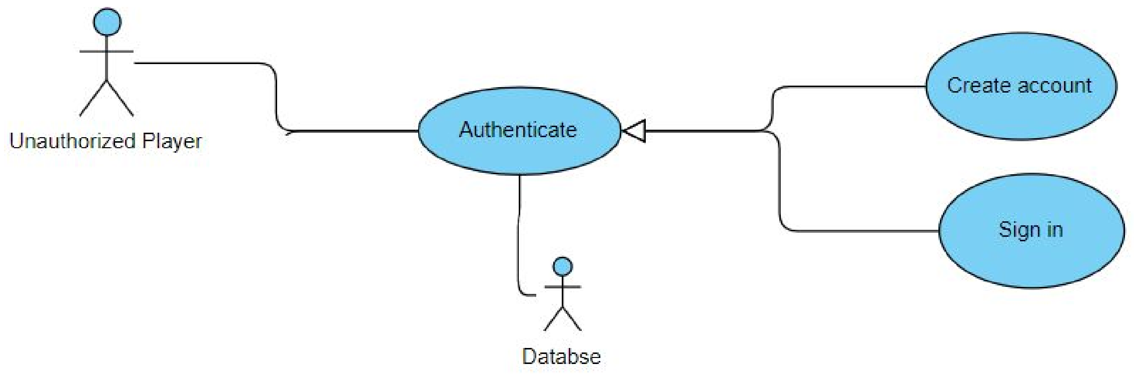
Exercise 2

The six functional requirements chosen are:

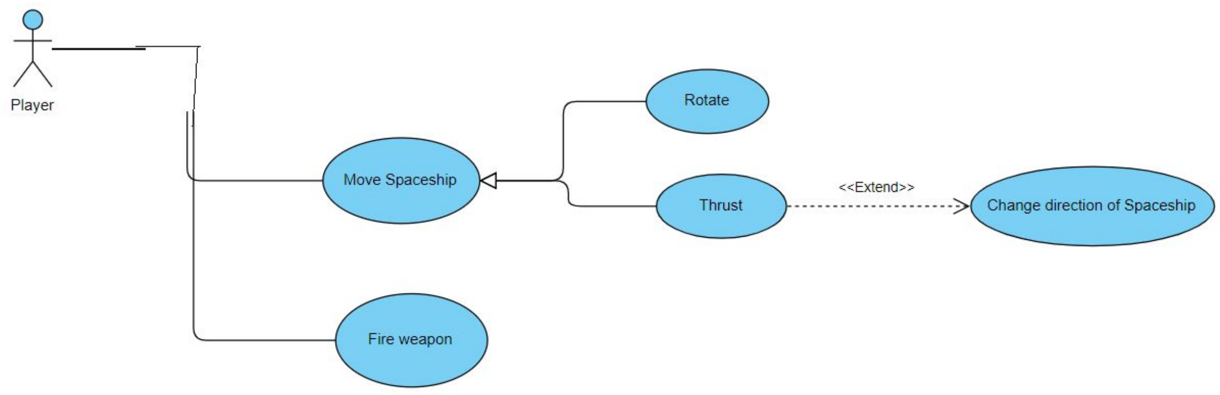
- The player shall be able to create an account with a username and password
- At the end of each play, the player shall be able to enter a name together with the recorded score.
- The player shall be able to rotate their spaceship at speed **tbd** using the left and right arrow keys (the way in which the spaceship will rotate (fixed amount on press or continuously while key is hold) is **tbd**)
- The player shall be able to thrust their spaceship forward by using the up arrow key
- The player can fire their weapon forward
- The player shall be able to change the direction their spaceship is moving in by thrusting



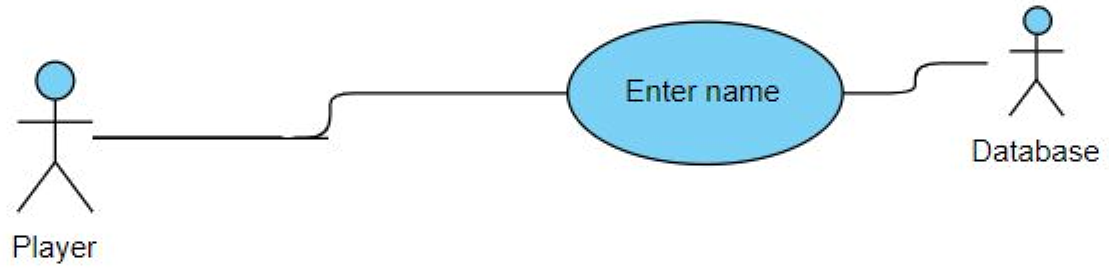
Authentication:



During the Game:



At the end of a play:



<u>Use case descriptions:</u>

1. Authenticate:

    Purpose: authenticating the players
    Overview: The player can either choose to create a new account or sign in by clicking on the ‘register’ or ‘log in’ buttons on the screen. Depending on which button they clicked on, the behaviour of that use case is applicable (in short: either a new account is made or the player logs in with an already existing account).
    Actors: Player, Unauthorized player, Database
    Pre condition: The program has started correctly, this will prompt the Main Screen page, where the user can decide whether to log in or create a new account.
    Post condition: Menu screen where the player can start the game appears and the player has a valid account.
    Normal flow of events: Follow the normal flow of events of the applicable use case.
    Alternate flow of events: Follow the alternate flow of events of the applicable use case.

2. Login:

    Purpose: player logs into their own account
    Overview: The player is prompted to type in their username, password. After filling all the fields, they can click on the sign in button to have their data checked for validity (with the ones stored in the database).
    Actors: Player, Database
    Pre conditions: The user has an account.
    Post conditions: The user is logged in and ready to start the game.
    Normal flow of events: The username-password combination given by the player constitutes a valid account. The player gets logged in and is ready to start the game.
    Alternate flow of events: The username-password combination given by the player doesn’t constitute a valid account. The player gets notified of this and they can try to log in multiple times.

3. Create an account:

    Purpose: creating a new account
    Overview: The player is prompted to type in their chosen username and password and they have to re-type the password for validation. The player clicks a button that sends this data to the database and the account is created.
    Actors: Unauthorized player, Database
    Pre conditions: The player doesn’t already have an account.
    Post conditions: The player now has an account: the given username and password combination form a valid account.
    Normal flow of events: The username given by the player is not taken yet. The account is created, the username and password are stored in the database and the player can now start the game.
    Alternate flow: The username given by the player is already taken. The player is prompted to give a different username.

4. Enter name:

    Purpose: the Player can choose to not have their username displayed in the highscores list but instead use an alias
    Overview: At the end of the play, the player gets prompted to type in an alias.
    Actors: Player, Database
    Pre conditions: The current play is finished.
    Post conditions: The player's score of the current play is stored.
    Normal flow: The player types in an alias that, together with their score of this play and the date, gets stored in the database.
    Alternate flow: The player doesn't type in an alias. Their score of the current play gets stored in the database together with their username.

5. Move spaceship:

    Purpose: let the Player control the movements of the spaceship
    Overview: Depending on the key pressed, the spaceship will be moved on the screen.
    Actors: Player
    Pre conditions: The game is started, is in progress and the player is alive.
    The player has pressed one of the following arrows: left, right, up, down or space.
    Post conditions: The spaceship has moved according to the keys pressed.
    Normal flow: Depending on the keys pressed, the spaceship will have moved in the desired direction. After movement is done, collisions need to be checked and action needs to be taken accordingly.
    Alternate flow: If the spaceship's new location falls outside of the game grid, it should wrap around: the spaceship should appear on the side opposite to the one where it "fell off" the grid.

6. Rotate spaceship

    Purpose: player can rotate their spaceship
    Overview: by pressing the left or right arrow keys, the spaceship will be rotated a certain amount of degrees around the x axis in the left or right direction.
    Actors: Player
    Pre conditions: The player has pressed the left or right arrow key.
    The game is started, is in progress and the player is alive.
    Post conditions: The spaceship is rotated in the desired direction.
    Normal flow of events: The player presses the left or right arrow key. The spaceship will be rotated a certain amount of degrees around the x axis in the specified direction.
    Alternate flow of events: -

7. Thrust spaceship

   Purpose: players can thrust their spaceship forward
   Overview: by pressing the up arrow key, the spaceship is moving for a certain distance 'forward' (in the direction it is facing) in a predetermined interval of time. Collisions are checked during and at the end of the transition.
   Actors: Player
   Pre conditions: The game is started, is in progress and the player is alive.
   The player has pressed the up arrow key.
   Post conditions: The spaceship's location is updated accordingly.
   Normal flow of events: The player presses the up arrow key. The spaceship is moving in the direction it is facing for a certain amount of time, after which it will stop.
   Alternate flow of events: If the spaceship's new location falls outside of the game grid, it should wrap around: the spaceship should appear on the side opposite to the one where it "fell off" the grid.

8. Fire weapon

   Purpose: players can shoot bullets with their weapon
   Overview: by clicking the space key, the player's weapon fires a bullet in the direction its spaceship facing. The bullet moves at a ***to-be-determined*** fixed speed.
   Actors: Player
   Pre conditions: The game is started and in progress, the player is alive.
   The player has bullets left, the weapon is not in cooldown.
   The player has pressed the to-he-determined key.
   Post condition: A bullet has been fired and collisions have been checked with the bullet.
   Normal flow of events: The player presses the space key. The spaceship fires a bullet in the direction that it is facing. This bullet travels with a ***to-be-determined*** speed in the given direction. Collisions are being checked and appropriate actions will be taken based on the possible collisions.
   Alternate flow of events: -