**3. Requirements**

**3.1 Functional Requirements**

* Player should be able to control the character with keyboard.
* Player should be able to pause the game.

1. Continuing game.
2. Changing settings.
3. Quitting game.

* Player should be able to choose settings

1. Changing the volume of background music.
2. Changing the volume of interaction sounds.

* The game should include different attributes that makes it interesting.

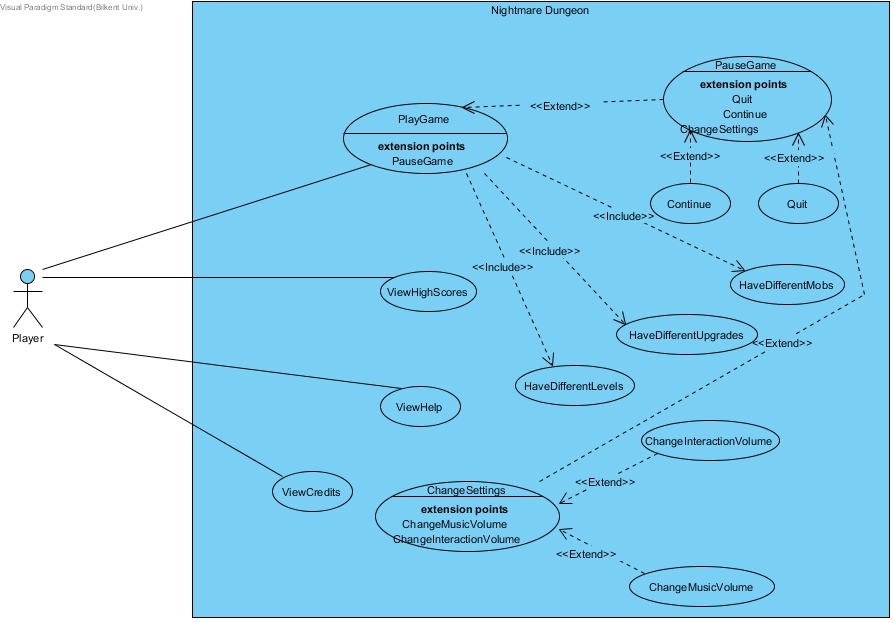
1. Having different upgrades for the player.
2. Having various levels.
3. Having diverse enemies and bosses.

* User should be able to see high scores of players.
* User should be able to see help which includes information of the game’s mechanics.
* User should be able to see information about credits.

* 1. **Non-Functional Requirements**
* Graphics should show that alice is in a nightmare to increase immersion.
* The game should run smoothly to offer quality gameplay.
* The controls should be responsive since the game is fast paced.
* All levels should allow the player to progress, meaning that randomly generated obstacles shouldn’t block the paths to other rooms.
  1. **Pseudo-Functional Requirements**
* The game will be implemented in java.
* JavaFX will be used for UI.

**4. System Model**

**4.1 Use Case Model**

****

**4.1.1 Play Game**

**Use case name:** PlayGame

**Participating actors:** Player

**Entry condition:** Player is on the main menu.

**Exit condition:**

* Player has defeated the final boss and won the game, OR
* Player ran out of lives and game is over, OR
* Player chose “Quit” from the pause menu.

**Main flow of events:**

1. Player starts the game.
2. Levels are randomly generated by the system.
3. Player starts from the neutral room without monsters or obstacles.
4. Player proceeds through the floors.
5. Player defeats the final boss.
6. Score of the player is displayed on the screen and if the score is in the top 10, game asks for the player’s name to save it to the top 10 list.
7. Player returns to main menu.

**Alternative flow of events:**

1. Player loses all his/her hearts and the game is over, player returns to the main menu.
2. Player exits the game with his own will.

**4.1.2 View Help**

**Use case name:** ViewHelp

**Participating actors:** Player

**Entry condition:** Player is on the main menu.

**Exit condition:** Player returns to main menu.

**Main flow of events:**

1. Player clicks on “View Help” on main menu.
2. Help is displayed.
3. Player chooses to return to the main menu.

**4.1.3 View High Scores**

**Use case name:** ViewHighScores

**Participating actors:** Player

**Entry condition:** Player is on the main menu.

**Exit condition:** Player returns to main menu.

**Main flow of events:**

1. Player clicks on “View High Scores” on main menu.
2. High scores and the names of the owners of high scores are displayed.
3. Player chooses to return to the main menu.

**4.1.4 View Credits**

**Use case name:** ViewCredits

**Participating actors:** Player

**Entry condition:** Player is on the main menu.

**Exit condition:** Player returns to main menu.

**Main flow of events:**

1. Player clicks on “View Credits” on main menu.
2. Names of the developers and their contributions to the game are displayed.
3. Player chooses to return to the main menu.

**4.1.5 Pause Game**

**Use case name:** PauseGame

**Participating actors:** Player

**Entry condition:** Player is playing the game.

**Exit condition:** Player returns to playing the game.

**Main flow of events:**

1. Player presses pause button while playing the game.
2. Pause menu is displayed.
3. Player continues playing the game.

**Alternative flow of events:**

* Players chooses to change the settings and goes to settings menu.
* Player quits the game.

**4.1.6 Quit**

**Use case name:** Quit

**Participating actors:** Player

**Entry condition:** Player has paused the game.

**Exit condition:** Player quits the game.

**Main flow of events:**

1. Player presses “quit”.
2. Player quits the game.

**4.1.7 Change settings**

**Use case name:** ChangeSettings

**Participating actors:** Player

**Entry condition:** Player has paused the game.

**Exit condition:** Player returns to pause screen.

**Main flow of events:**

1. User changes the interaction volumes or the music volume.
2. User goes back to pause screen.

**4.1.8 Continue**

**Use case name:** Continue

**Participating actors:** Player

**Entry condition:** Player has paused the game.

**Exit condition:** Player returns to game.

**Main flow of events:**

1. User presses continue.
2. User goes back to playing the game.

**4.1.9 Change Music Volume**

**Use case name:** ChangeMusicVolume

**Participating actors:** Player

**Entry condition:** Player is in the settings menu.

**Exit condition:** Player returns to settings menu.

**Main flow of events:**

1. User adjusts the music volume.
2. User goes back settings menu.

**4.1.10 Change Interaction Volume**

**Use case name:** ChangeInteractionVolume

**Participating actors:** Player

**Entry condition:** Player is in the settings menu.

**Exit condition:** Player returns to settings menu.

**Main flow of events:**

1. User adjusts the interaction volume.
2. User goes back settings menu.

**4.1.11 Have Different Upgrades**

**Use case name:** HaveDifferentUpgrades

**Participating actors:** Player

**Entry condition:** Player is playing the game.

**Exit condition:** Player exits the game.

**Main flow of events:**

1. The game offers various upgrades to the player.

**4.1.12 Have Different Mobs**

**Use case name:** HaveDifferentMobs

**Participating actors:** Player

**Entry condition:** Player is playing the game.

**Exit condition:** Player exits the game.

**Main flow of events:**

1. The game has different mob types.

**4.1.13 Have Different Levels**

**Use case name:** HaveDİfferentLevels

**Participating actors:** Player

**Entry condition:** Player is playing the game.

**Exit condition:** Player exits the game.

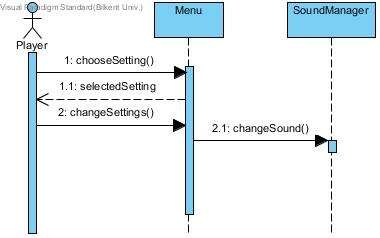
**Main flow of events:**

1. The game has different level layouts.

**4.2. Dynamic Models**

**Scenario Name:** Change Settings

**Scenario:** Oğuz wants to change the game’s settings. He selects the “Change Settings” icon from either the main menu or the pause menu. The system displays 2 settings: Adjust interaction voice and adjust music voice. Oğuz then selects the setting that he desires to change and changes it. Oğuz then exits the settings window and the system updates the changes that were made.



**Scenario Name:** Move Player

**Scenario:** Oğuz needs to move the Alice character to progress through the game. Oğuz presses the arrow keys to move the Alice character up, down, left or right. Alice moves as Oğuz directs her.

