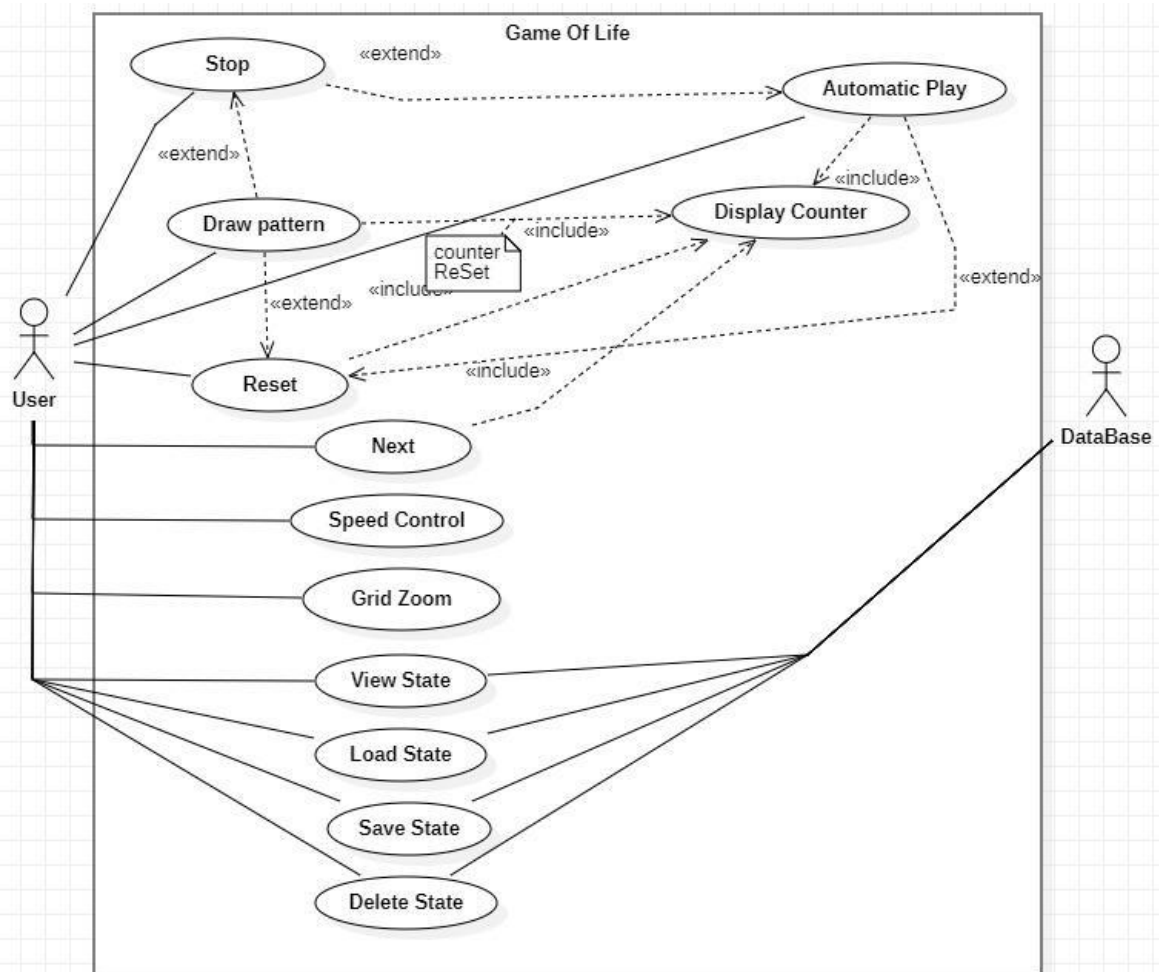# SDA Project Phase 1

Submitted to: Sir Amir Iqbal
Section: 5G
Submission date: October 31, 2021

Name	Roll Number
Bilal Sardar	**19L-1232**
Minnah Amir	**19L-2391**
Shaheer Ahmed Virk	**19L-2390**
Shimaim Khan	**19L-1170**
Iman Jawad	**19L-2737**
Raza Malik	**19L-2368**
Muhammad Hassnain	**19L-1055**

USE CASE DIAGRAM:

USE CASE DESCRIPTION:

Use Case: Automatic play

Summary: The game selects a pattern for gameplay(can be from lexicon) and then displays the counter for it according to the moves generated.

Actors: User

Preconditions: The game is waiting to be started with a generated pattern.

Description: When the user starts the game,it's played with the pattern pre generated(can be from lexicon). Different cells are populated and killed based on the rules of the game. With each change in shape,it creates a new pattern which increments and displays the counter. The user can stop and reset the game and the counter when he wants

Exceptions:
Cancellation: When selecting a cell/cells during gameplay near the pattern during gameplay,the pattern may turn grey with increasing counter.
Post conditions: The game is being played.

Use Case: Draw Pattern
Summary: The user can draw a pattern by selecting/deselecting cells. This will start the gameplay with options to stop/reset the game.
Actors: User
Preconditions: The game is waiting to be started.
Description: The user can start a game by drawing a pattern of their own choice, new patterns will be generated from that. They can also stop and restart the game from where they left. The counter and game can also be reset.
Exceptions:
Cancellation: When selecting a cell/cells during gameplay near the pattern during gameplay,the pattern may turn grey with increasing counter.
Post conditions: The game is being played.

Use Case: Stop
Summary: The user can stop the game at any point.
Actors: User
Preconditions: The game has been started.
Description: A game when started can be stopped anytime by a user if he wants, whether it is from an automatically generated pattern or drawn by a user. The stopped game can be restarted again anytime.
Post conditions: The game is stopped.

Use Case: Reset
Summary: The game being played can be reset any moment, which in result resets the counter as well.
Actors: User
Preconditions: The game has been started.
Description: A game when started can be rest anytime by a user if he wants, whether it is from an automatically generated pattern or drawn by a user. The reset game will also reset the counter to 0.
Post conditions: The game is reset and ready to start again.

Use Case: Display Counter

Summary: With each change in pattern/reset the counter is set accordingly.

Actors: User

Preconditions: The game is waiting to be started.

Description: When a game is started the counter is incremented whether it is from an automatically generated pattern or drawn by a user. The reset game will also reset the counter to 0.

Exceptions:

Cancellation: When selecting a cell/cells near the pattern during gameplay,the pattern may turn grey with increasing counter.

Post conditions: The counter is being displayed

Use Case: Next

Summary: The user can see the next moves after stopping/ resetting the game

Actors: User

Preconditions: The game is waiting to be started/has already started.

Description: A User can review the next moves about to be generated by the game.

Post conditions: The moves are being displayed

Use Case: Speed Control

Summary: Before or during the game the speed can be increased/decreased.

Actors: User

Preconditions: The game is waiting to be started/has already started.

Description: A User can increase or decrease the speed of the game as much as he wants

Post conditions: The game speed is increased/decreased.

Use Case: Grid Zoom

Summary: Before or during the game the grid can be zoomed in/out.

Actors: User

Preconditions: The game is waiting to be started/has already started.

Description: A User can zoom in/out the grid of the game as much as he wants

Post conditions: The grid is zoomed in/out.

Use Case: View State

Summary: The state of the game is viewed through the view state option.

Actors: User, Database

Preconditions: The game is being viewed.

Description: When a user starts a game, the state of the game is viewed by the view state option.

Post conditions: The game has already started.

Use Case: Save State

Summary: A user can save the current state of the game.

Actors: User, Database

Preconditions: The game has already started.

Description: When a user saves a game, the state of the game is saved by the database.

Post conditions: The game's state has been saved.

Use Case: Load State

Summary: When a user saves a state it can be loaded again by the load state option.

Actors: User, Database

Preconditions: The game has already been saved.

Description: A saved game can be loaded again.

Post conditions: The saved game's state has been loaded.

Use Case: Delete State

Summary: When a user saves a state it can be deleted by the delete state option.
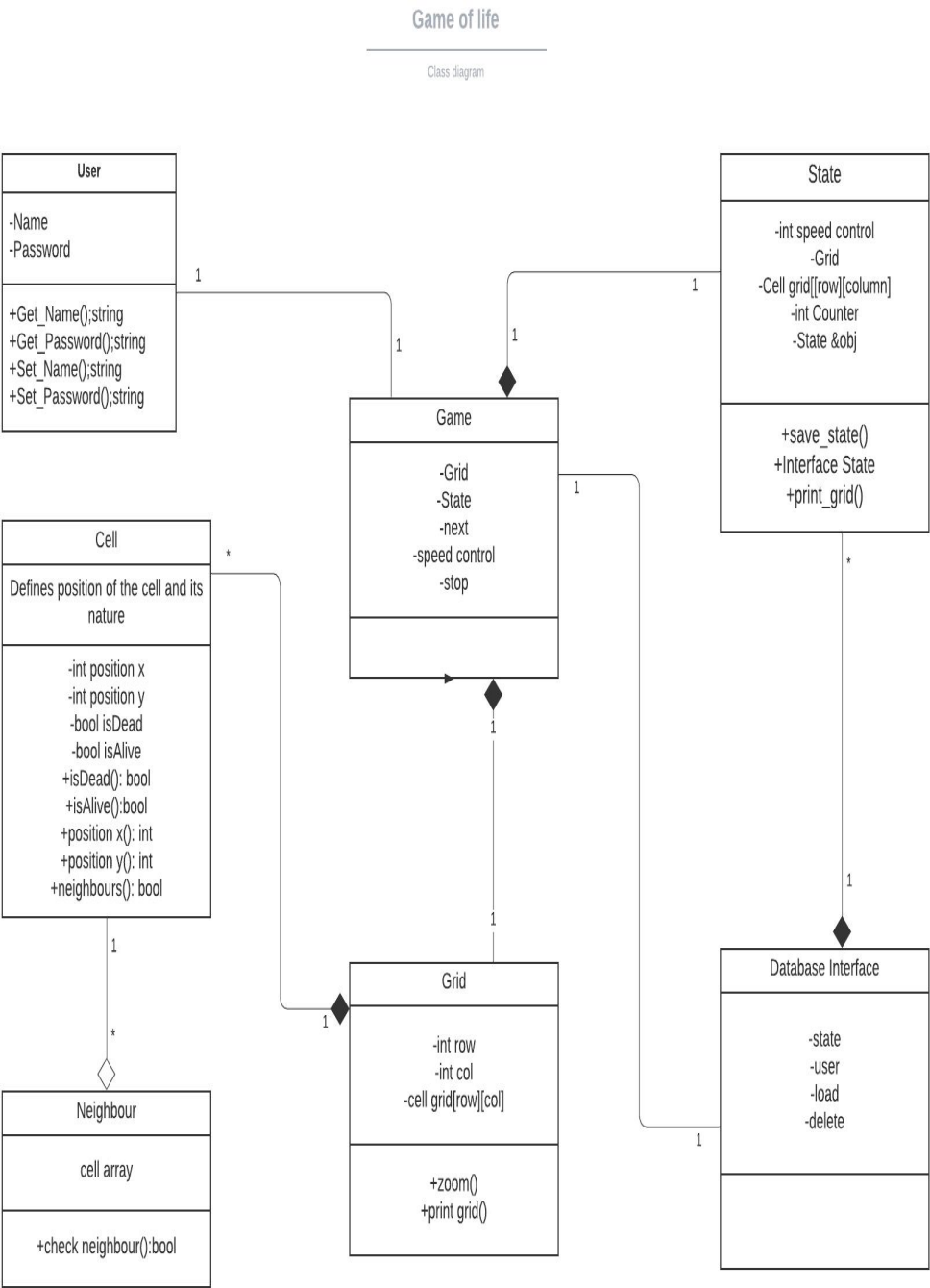
Actors: User, Database

Preconditions: The game has already been saved.

Description: A saved game can be deleted.

Post conditions: The saved game's state has been deleted.
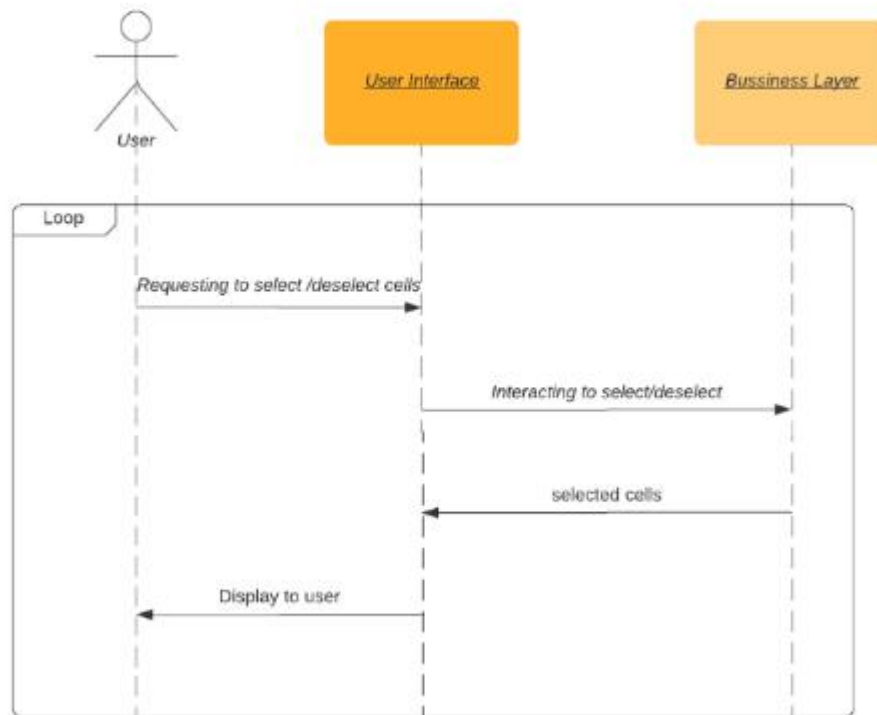
UML Class Diagram:

Game of life

Class diagram

User

-Name
-Password

+Get_Name();string
+Get_Password();string
+Set_Name();string
+Set_Password();string

State

-int speed control
-Grid
-Cell grid[[row][column]
-int Counter
-State &obj

+save_state()
+Interface State
+print_grid()

Game

-Grid
-State
-next
-speed control
-stop

Cell

Defines position of the cell and its nature

-int position x
-int position y
-bool isDead
-bool isAlive
+isDead(): bool
+isAlive():bool
+position x(): int
+position y(): int
+neighbours(): bool

Neighbour

cell array

+check neighbour():bool

Grid

-int row
-int col
-cell grid[row][col]

+zoom()
+print grid()

Database Interface

-state
-user
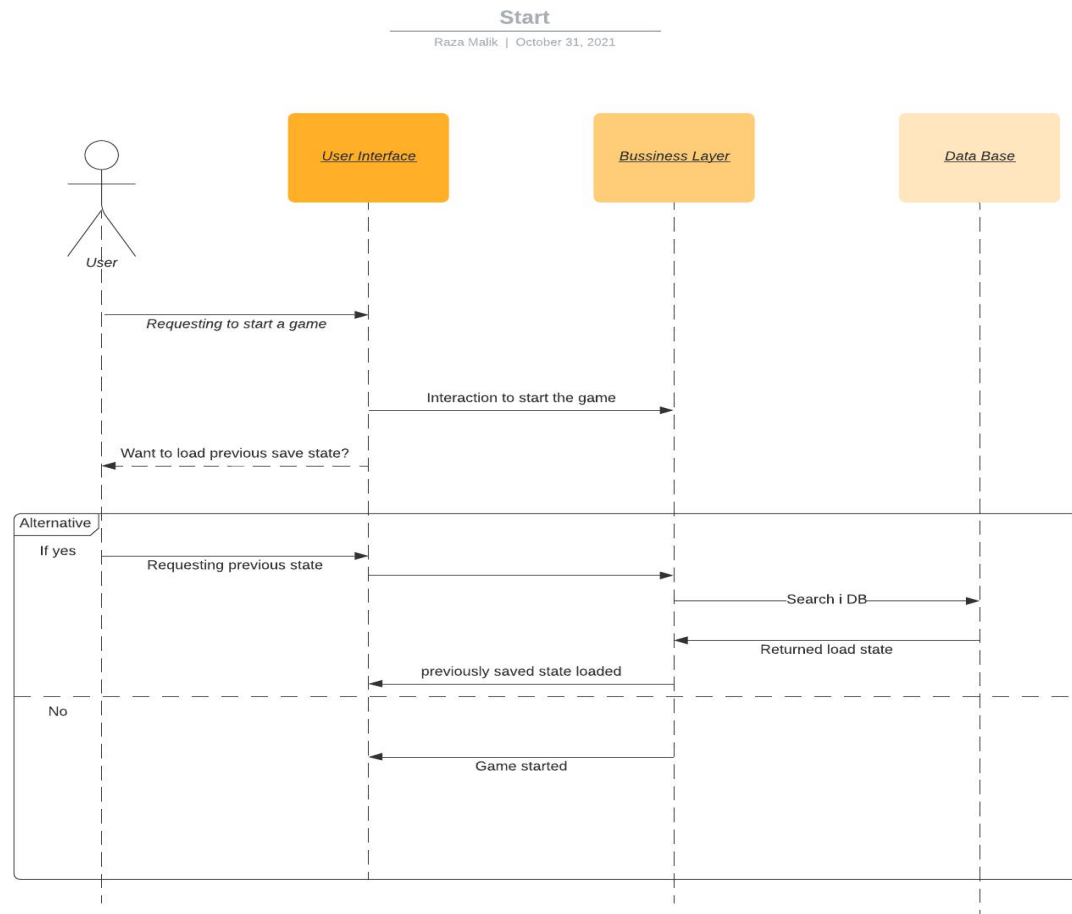-load
-delete

SEQUENCE DIAGRAM:

Use Case 1:

➢ Manually draw a pattern by selecting/deselecting cells
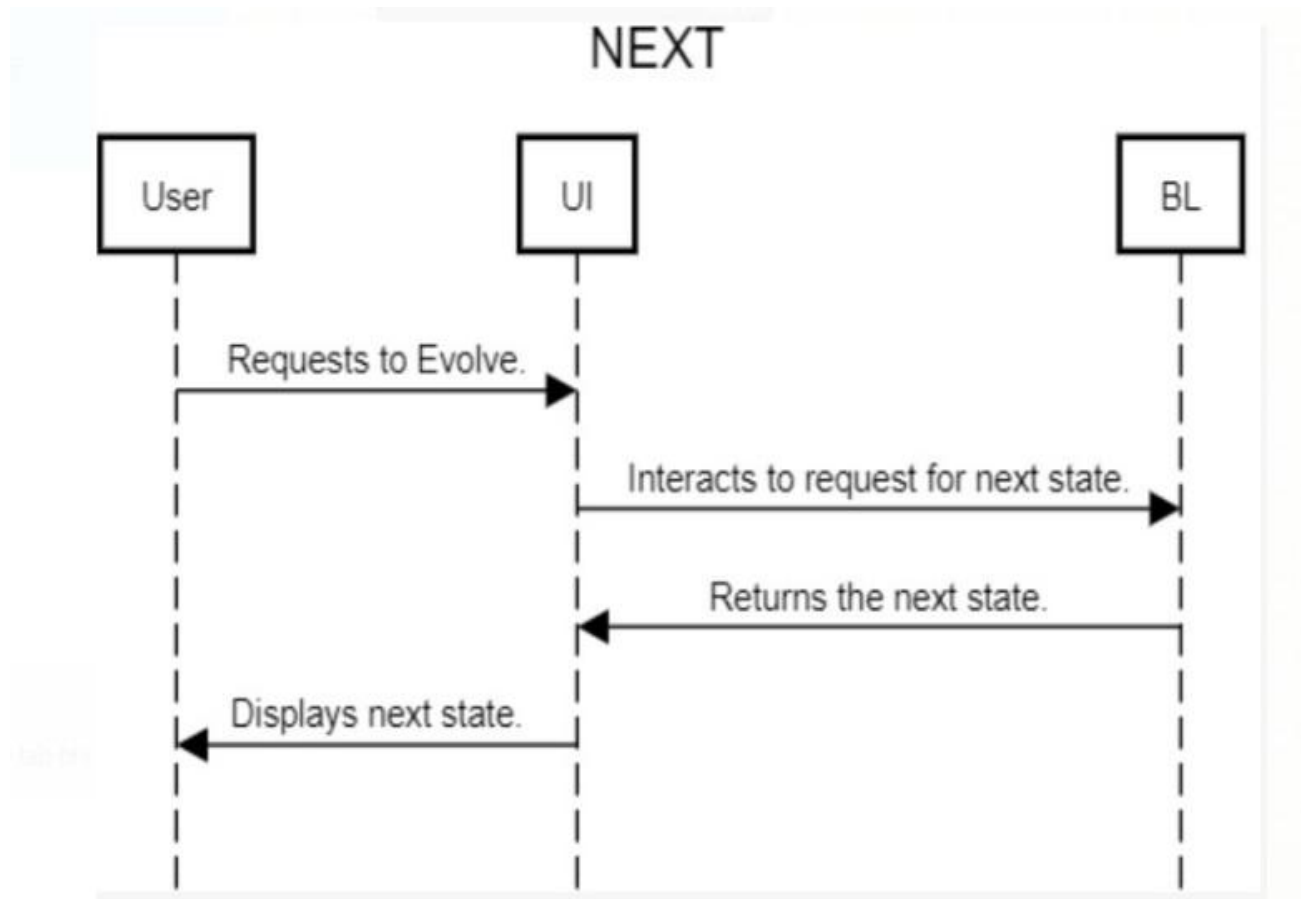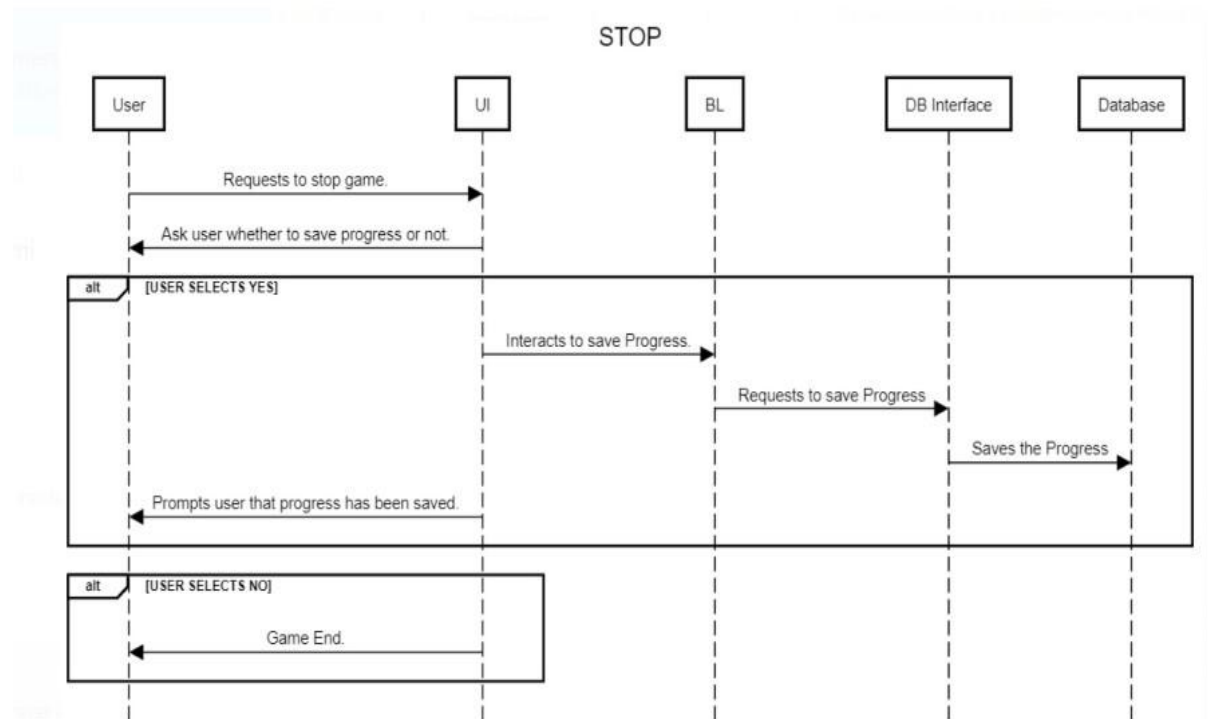
Selecting/Deselecting cells
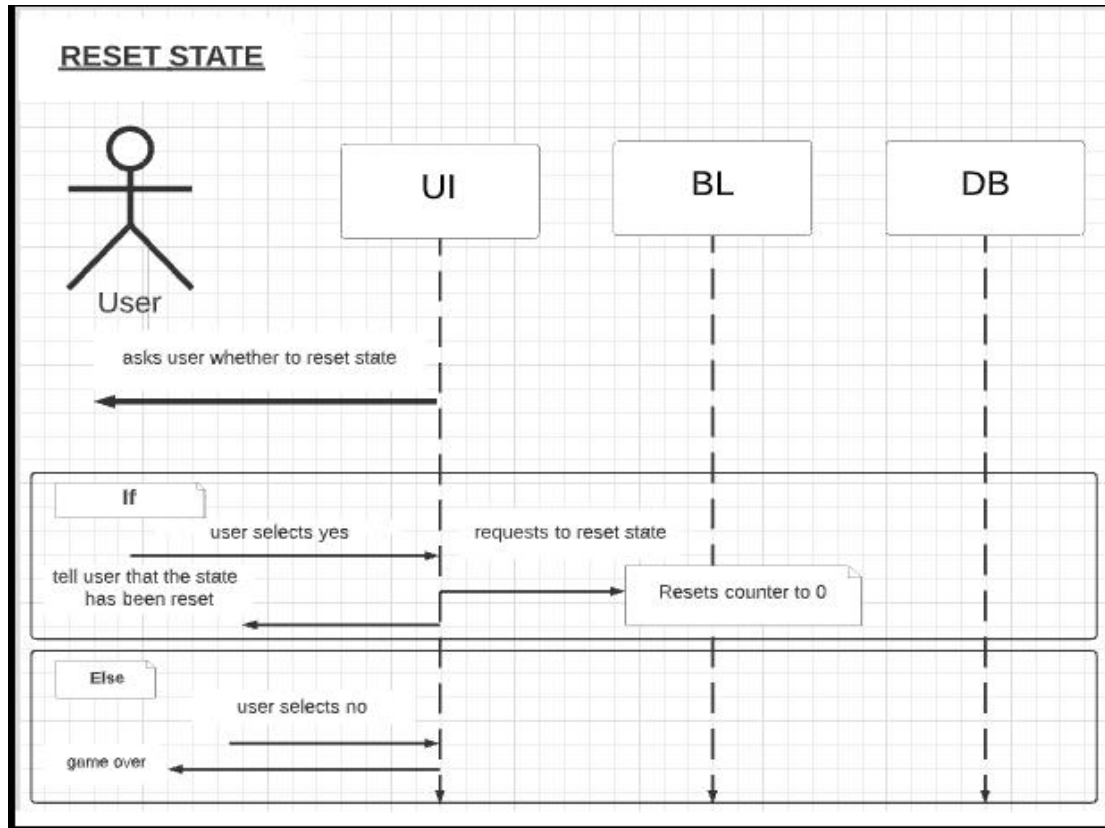
Use Case 2:
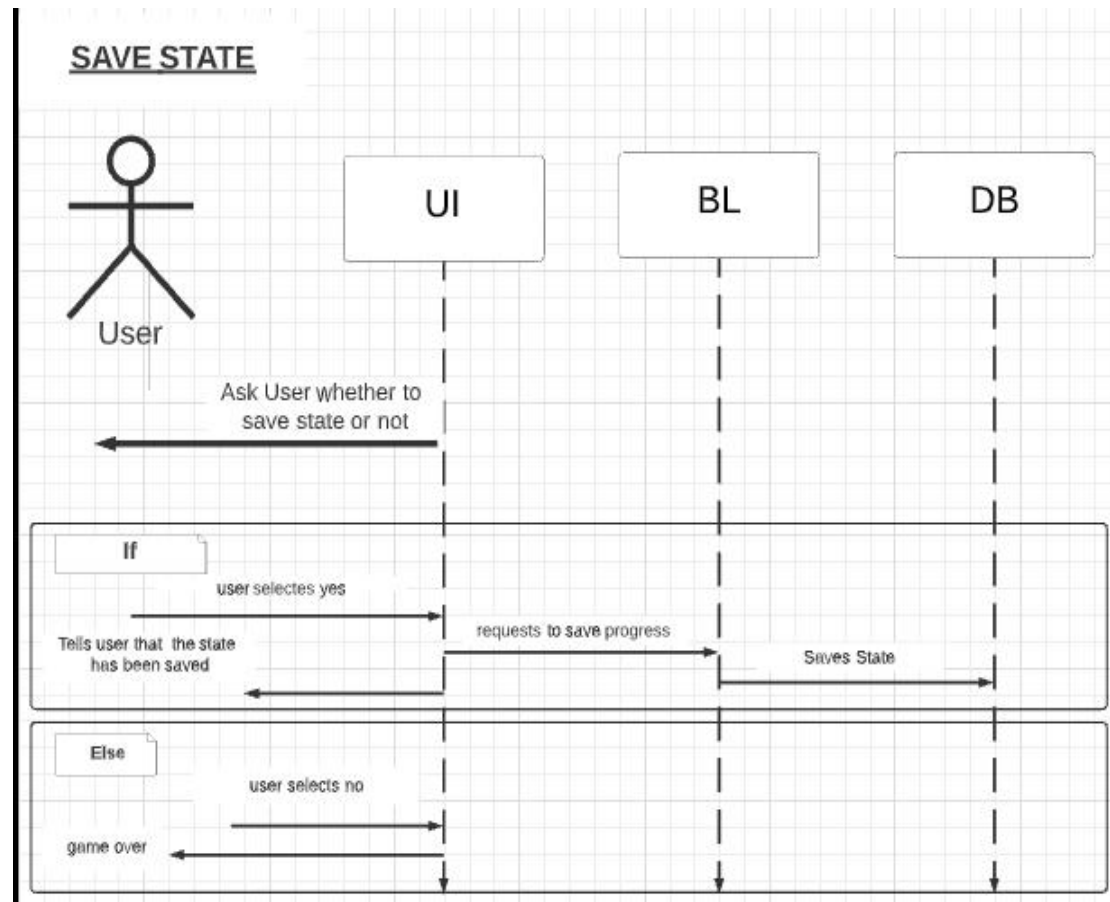
> ## Start

Use Case 3:

➤ *Next*

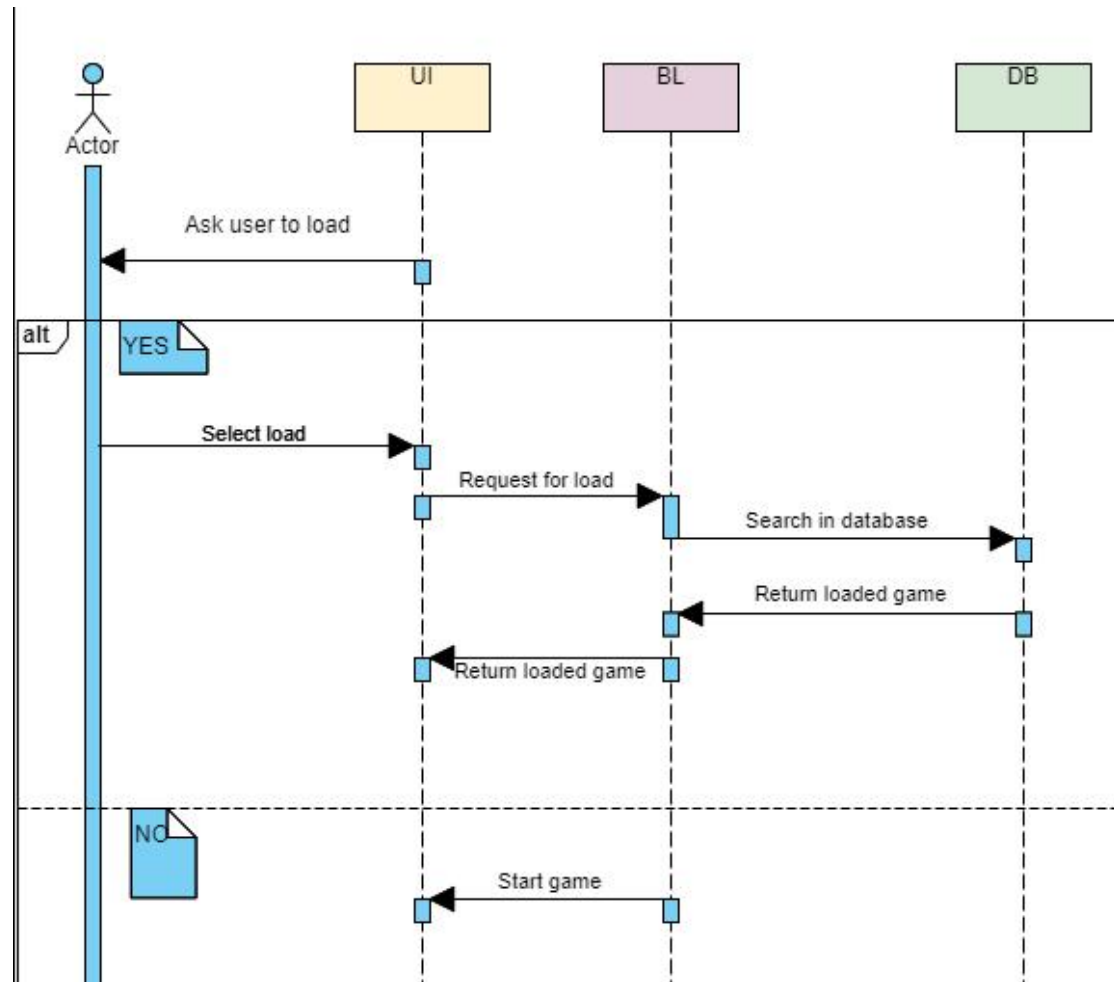## Use Case 4:

➢ *Stop*

STOP

Use Case 5:

➢ Reset State

RESET STATE

User → UI → BL → DB

asks user whether to reset state

If
user selects yes → requests to reset state
Resets counter to 0
tell user that the state has been reset

Else
user selects no
game over

Use Case 6:

➢ Save State

SAVE STATE

User

UI   BL   DB

Ask User whether to save state or not

If

user selectes yes

requests to save progress

Saves State

Tells user that the state has been saved

Else

user selects no

game over

Use Case 7:

➢ Load

Use case 8:

➢ *Speed Control*

	UI	BL	DB

User

Request for speed control

Interacts for speed control

Return controlled speed

Display controlled speed

Use Case 9:

➢ Grid Zoom

User	UI	BL	DB

Request to zoom grid

Interacts to zoom grid

Returns zoomed grid

Display zoom grid
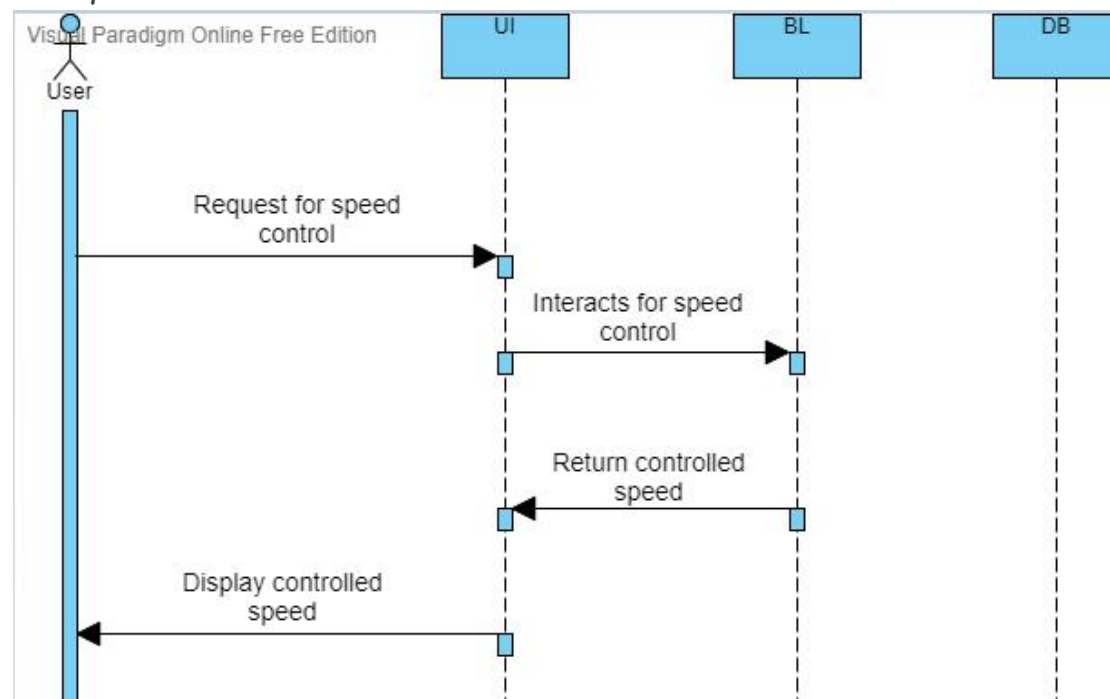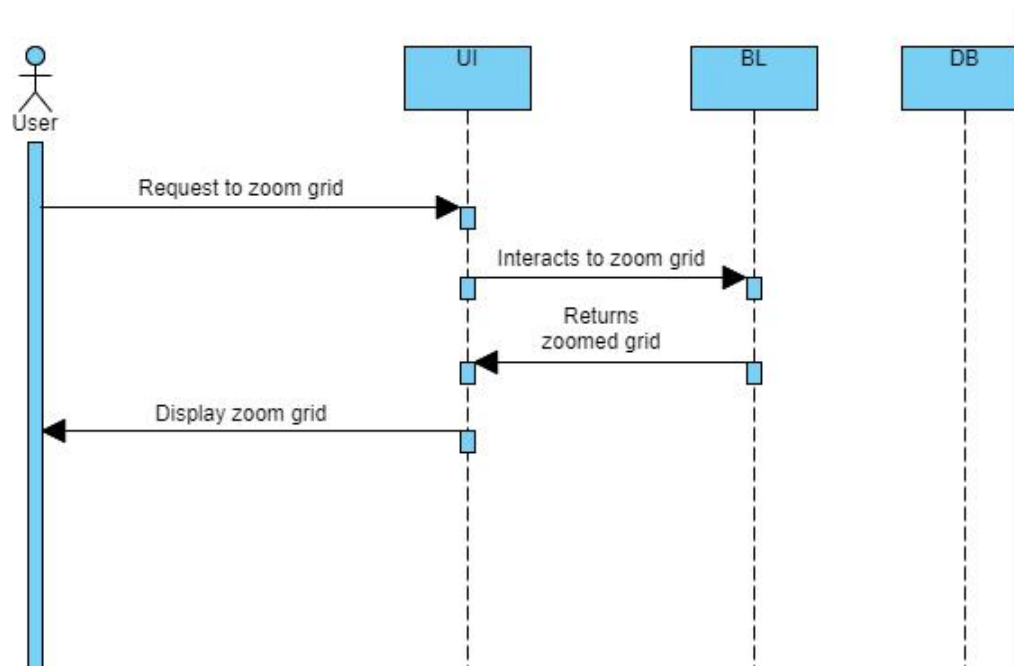
Use Case 10:

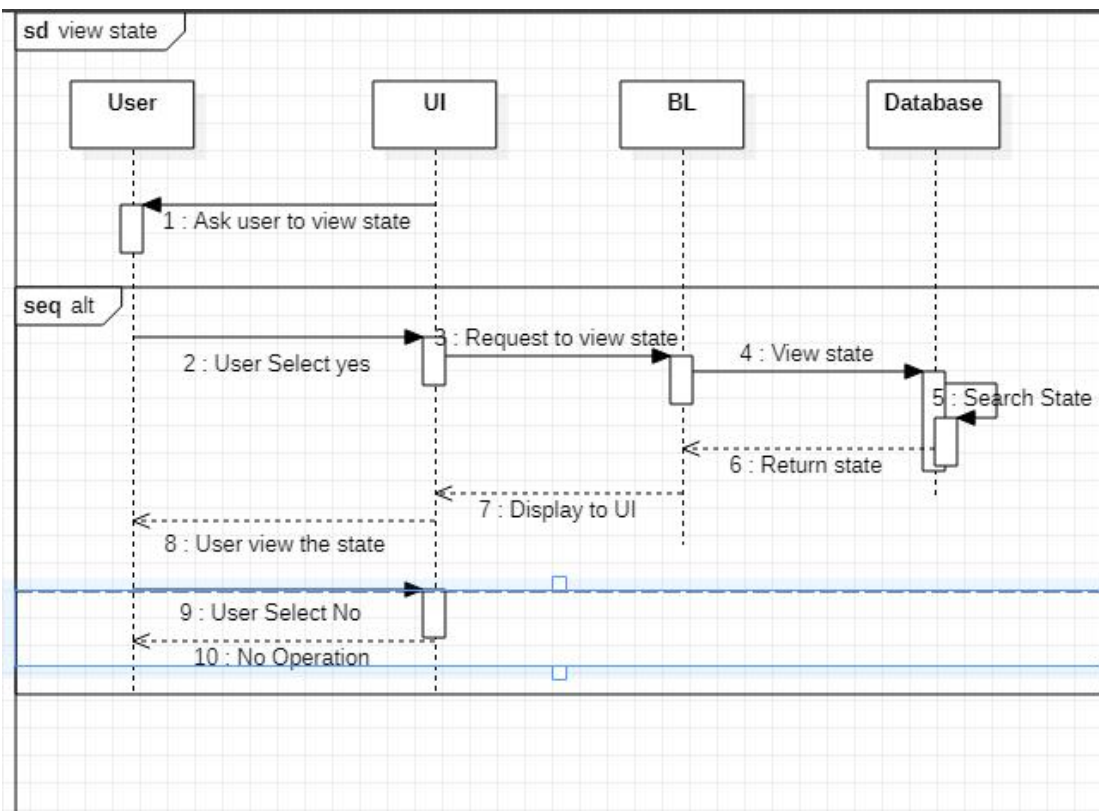➢ *Counter*
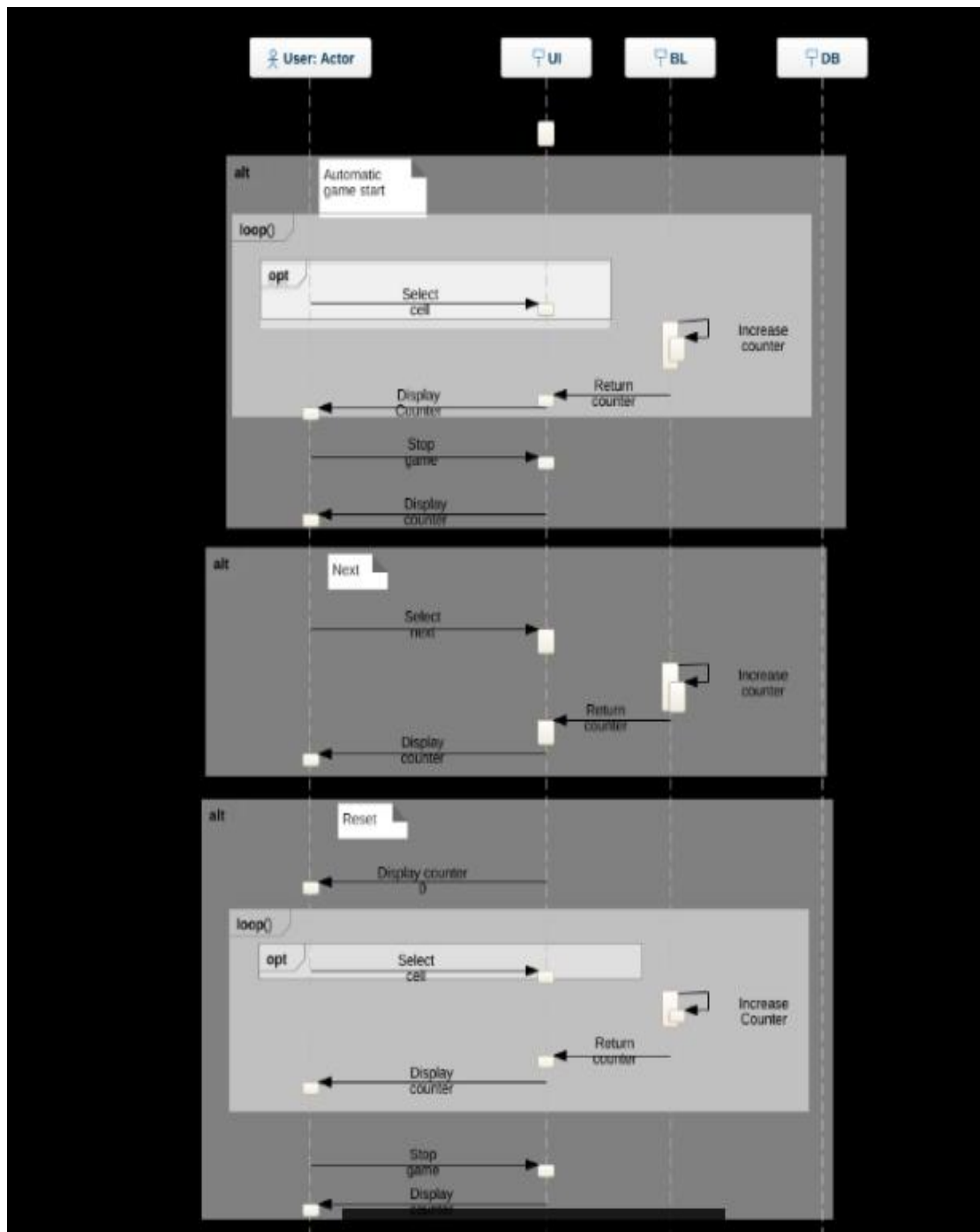
Use Case 11:

➢ View Saved States

Use Case 12:

➢ Delete a Saved State

sd Delete state

| User | UI | BL | Database |

1 : Ask user to Delete state

seq alt

2 : User Select yes

3 : Request to Delete state

4 : Delete state

5 : Search State

6 : Return Message(0 or 1)

7 : Display to UI

8 : User Receive message

9 : User Select No

10 : No Operation