

Elsa Morina
89211086

Domenico Salvatore Maria Panarello
76230039

SRS DOCUMENT

Elsa Morina

Domenico Salvatore Maria Panarello

University of Primorska
FAMNIT – Computer Science

Contents

Problem Definition	3
Functional Requirements	4
Non-Functional Requirements	4
Feasibility Study	5
User Story:.....	6
Use-Case Diagram:	6
Use Case Descriptions	7
Use Case: Play Game (Elsa Morina).....	7
Use Case Select Room (Domenico Salvatore Maria Panarello).....	8
Use Case Create Room	10
Use Case Join Room	10
Use Case Alternate Game Mode	11
Use Case Draw Cards.....	11
Use Case Move	12
Use Case Stay in Jail.....	13
Use Case Roll Dices.....	13
Additional Diagrams.....	14
Activity Diagram (Elsa Morina).....	14
Sequence Diagram (Domenico Salvatore Maria Panarello)	15
Mock Ups	16

Problem Definition

A U.S gaming company specializing in remote board games, wants to meet user demands by introducing Monopoly game to its online platform. The challenge lies in creating an engaging and user-friendly version of Monopoly that captures the essence of traditional board games. Players have shown a great desire to play Monopoly virtually against people from around the world as well as with friends and family. The company's goal is to create an entertaining and intuitive online version of Monopoly that maintains the classic gameplay while providing cross-platform compatibility, customizable settings, and additional modes.

Functional Requirements

- The players can join games from different devices and locations.
- The players can pause and resume the game at their convenience.
- The players have the option to leave the game at any point.
- The game needs to be aesthetically pleasing.
- The players can navigate themselves easily.
- The players need internet to play.
- The players can create a game room.
- The players can join a game room.
- The players that are creators can cancel a created room.
- The players can change the game mode.

Non-Functional Requirements

1. Performance: The system must support a minimum of 200 players that concurrently play in different game rooms with no problems.
2. Reliability: The system must have a 99% uptime, certifying minimal disturbances while playing the game.
3. Usability: UI is simple and user-friendly. All players with or without technical knowledge can use it.
4. Compatibility: The game can be played and accessed from different web browsers.
5. Scalability: The game's architecture must enable future updates.

Feasibility Study

Technical Feasibility:

- Develop a Monopoly platform from scratch.
- Have compatibility between every device and browser.
- Apply features like play and quit game.
- Enable playing of game at real time.

Economic Feasibility:

- Initial development costs are high.
- Maintenance costs are low and manageable.
- Possible profits from ads.
- Could take approximately 8 months to make.

Organizational Feasibility:

- Existing tech team of the company can develop, design and maintain the platform.
- Addressing legal aspects in regards to data privacy.

According to the feasibility study done for the game, the solution proposed gives profits and saves some costs. It also has a general lower initial cost. There is enough control on how the platform will look like and work. It does require a bit of time to build. Nonetheless the game is a feasible project.

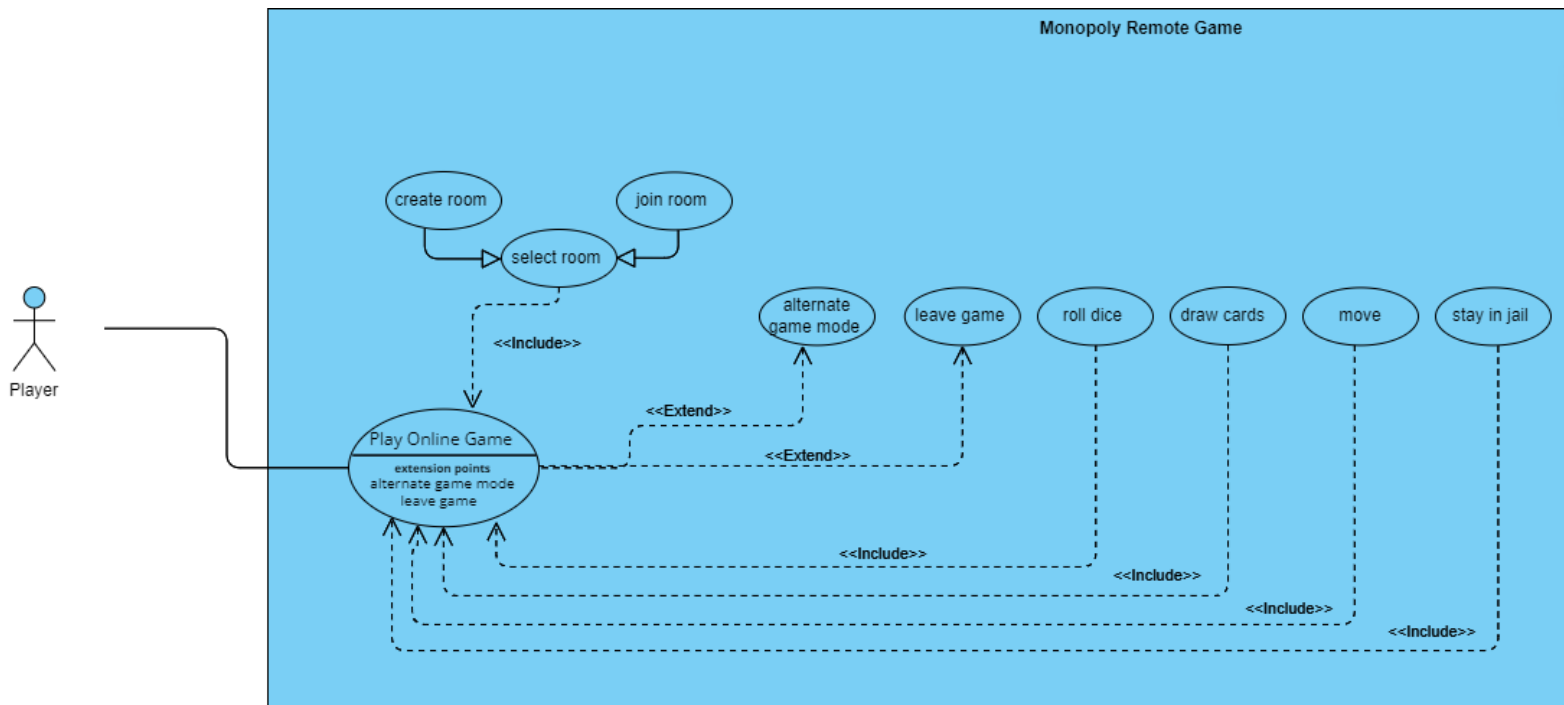
User Story:

The Player wants to play remote Monopoly game.

Actor: Player

-The Player decides to play a game online. He must select the type of room he is going to play in. In the join room, he already gets a code and in the create room he creates a room where other players are invited. The player can alternate game mode to change the board appearance and potentially modify rules for a customized experience. The player can decide to leave the game when no longer interested in playing. The player must roll dice, take cards and move when turn is theirs.

Use-Case Diagram:



Use Case Descriptions

Use Case: Play Game (Elsa Morina)

1. Use Case Name: Play game
 2. Actor: Player
 3. Preconditions: The game is launched, and the player is in main menu screen.
 4. Trigger: Player enters link on browser for the game.
 5. Main Flow:
 1. Actor Action: Player selects "Play Game."
 2. System Response: The system displays the options to either "Create Room" or "Join Room."
 3. Actor Action: The player is the game creator: refer to "Select Room"
 4. Actor Action: The player enters the game room
 5. Actor Action: Players wait for others to join.
 6. System Response: The system provides real-time updates on the number of joined players and displays relevant information about the game (e.g., current players, settings).
 7. Actor Action: Game creator starts the game when ready by pressing start game button.
 8. System Response: The system initiates the Monopoly game with the selected settings.
 9. Actor Action: Player takes their turn in a predefined order.
 10. System Response: The system displays relevant information (e.g., current position, available actions).
 11. Actor Action: Player rolls the dice by clicking the roll dice button.
 12. System Response: The system generates a random result representing the dice roll and moves the player's token on the game board.
 13. Actor Action: Player gets to be in jail.
 14. System Response: System puts him to jail
 15. Actor Action: Player take cards from the deck provided
 16. System Response: System gives the cards to player
- (The game continues until a win condition is met (e.g., one player bankrupts' others)).
17. Actor Action: Players view results of the game.
 18. System Response: The system presents the winner.

6. Post-condition: The game ends and we have a winner.

7. Alternative Flow:

5.4.a: Game Creator Cancels Room Creation:

5.4.a.1. Actor Action: Game creator cancels room creation.

5.4.a.2. System Response: The system acknowledges the cancellation and returns the player to the main menu.

5.11.a: Player Quits Mid-Game:

5.11.a.1 Actor Action: Player decides to quit the game.

5.11.a.2 System Response: The system updates the game state, potentially allowing other players to continue or ending the game.

Use Case Select Room (Domenico Salvatore Maria Panarello)

1. Use Case Name: Select Room

2. Actor: Player

3. Preconditions: The player is on the main menu screen and has selected the "Play Game" option.

4. Trigger: Clicking Play Button

5. Main Flow:

5.1. The system prompts the player to set up a new game room by selecting "Create Room."

Actor Action: Player selects "Create Room."

System Response: The system guides the player through customizing game settings (e.g., number of players, starting funds).

Actor Action: Player customizes game settings.

System Response: The system generates a unique game code for the room.

Actor Action: Player waits for other players to join by sharing the game code.

System Response: The system displays the unique game code and provides a waiting interface for other players to join.

5.2. The player selects "Join Room" and enters the provided game code.

Actor Action: Player selects "Join Room."

System Response: The system prompts the player to enter the provided game code.

Actor Action: Player enters the game code.

System Response: The system validates the code and allows entry into the existing game room.

5.3. The player is now in the game room, where they can see information about the game and other players.

System Response: The system displays information about the game room, such as the current players, game settings, and any relevant details.

6. Postconditions: The player is in the specified game room, ready to participate in the Monopoly game with other players.

7. Alternative Flows:

5.2.a: Invalid Game Code:

5.2.a.1. Actor Action: Player enters an invalid game code.

5.2.a.2. System Response: The system displays an error message indicating that the entered game code is invalid.

5.2.a.3. Actor Action: Player acknowledges the error message.

5.2.a.4. System Response: The system prompts the player to re-enter a valid game code.

5.2.b: Room is Full:

5.2.b.1. Actor Action: Player enters a valid game code.

5.2.b.2. System Response: The system validates the code but detects that the room is already full.

5.2.b.3. Actor Action: Player acknowledges the room is full.

5.2.b.4. System Response: The system provides options to join another room or return to the main menu.

5.2.c: Game Already Started:

5.2.c.1. Actor Action: Player enters a valid game code.

5.2.c.2. System Response: The system validates the code but detects that the game has already started.

5.2.c.3. Actor Action: Player acknowledges the game has started.

5.2.c.4. System Response: The system provides options to join another room or return to the main menu.

Use Case Create Room

1. Use Case Name: Create Room
2. Actor: Player
3. Preconditions: The player is on the select room screen and has selected the "Create room" option.
4. Trigger: Clicking the "Create Room" button.
5. Main Flow:
 1. Actor Action: Player customizes the game settings.
 2. System Response: System changes the game settings.
 3. System Response: System creates unique game code.
 4. Actor Action: Player waits for other players.
 5. System Response: The system displays a waiting interface for other players to join.
7. Postconditions: The player is in the specified game room, ready to start the Monopoly game with other players.
8. Alternative Flows:
 - 5.4.a Create Room Cancelled:
 - 5.4.a.1 Player cancels room creation
 - 5.4.a.1 System accepts the cancelling of the room and returns the player in the main page.

Use Case Join Room

1. Use Case Name: Join Room
2. Actor: Player
3. Preconditions: The player is on the select room screen and has selected the "Join room" option.
4. Trigger: Clicking the "Join Room" button.
5. Main Flow:
 1. Actor Action: Player enters the join room screen.
 2. System Response: The system prompts the player to enter the provided game code.
 3. Actor Action: Player enters game code.
 4. System Response: The system validates the code and allows entry into the waiting screen.

6. Postconditions: The player is in the waiting room, waiting to participate in the game.

7. Alternative Flows:

5.3.a Invalid Game Code:

5.3.a.1 Player enters an invalid game code

5.3.a.2 System displays an error message that code is invalid

5.3.a.3 Player identifies the error message

5.3.a.4 System prompts payer to enter a working code.

Use Case Alternate Game Mode

1. Use Case Name: Alternate Game Mode

2. Actor: Player

3. Preconditions: The player is on the create room screen after selection of "Create Room" button.

4. Trigger: Clicking an Alternate Game Mode

5. Main Flow:

1. Actor Action: Player scrolls between alternative game modes

2. System Response: The system presents the options of alternative game modes

3. Actor Action: Player makes a selection of the game mode

4. System Response: The system applies the change to the game

6. Post-Conditions: The game continues with the customized settings.

Use Case Draw Cards

1. Use Case Name: Draw Cards

2. Actor: Player

3. Preconditions: The player is in an active Monopoly game and has the option to draw cards during their turn.

4. Trigger: Player chooses to draw cards during their turn.

5. Main Flow:

1. Actor Action: Player selects a card from the deck.

2. System Response: The system reveals the content of the selected card, which may include instructions, rewards, penalties, or other game-related actions.

3. Actor Action: Player follows card instructions.

4. System Response: The system updates the game state based on the card's instructions.

6. Postconditions: The player has drawn and resolved a card, and the game state is updated. Control is passed to the next player.

Alternative Flow:

5.1.a. Deck is empty.

5.1.a.1 System Response: If the deck is empty, system shuffles the discard pile to form a new deck.

Use Case Move

1. Use Case Name: Move
2. Actor: Player
3. Preconditions: The player is in an active Monopoly game and has the option to move during their turn.
4. Trigger: Player chooses to move during their turn.
5. Main Flow:
 - Actor Action: Player chooses to move.
 - System Response: The system prompts the player to roll virtual dice.
 - 5.1. Player rolls dice.
 - Actor Action: Player rolls the dice.
 - System Response: The system calculates the dice result.
 - 5.2. Player moves game piece.
 - Actor Action: Player moves their game piece.
 - System Response: The system updates the game board with the player's new position.
 - 5.3. Buy a property.
 - System Response: If applicable, the system prompts the player to purchase an unowned property.
6. Postconditions: The player has completed their move, and the game state is updated. Control is passed to the next player.
7. Alternative Flow:
 - 5.3.a Player decide to not buy the property
 - 5.3.a.1 System Response: If the player decides not to purchase an unowned property, the system skips the property purchase step and proceeds directly to ending the turn.

Use Case Stay in Jail

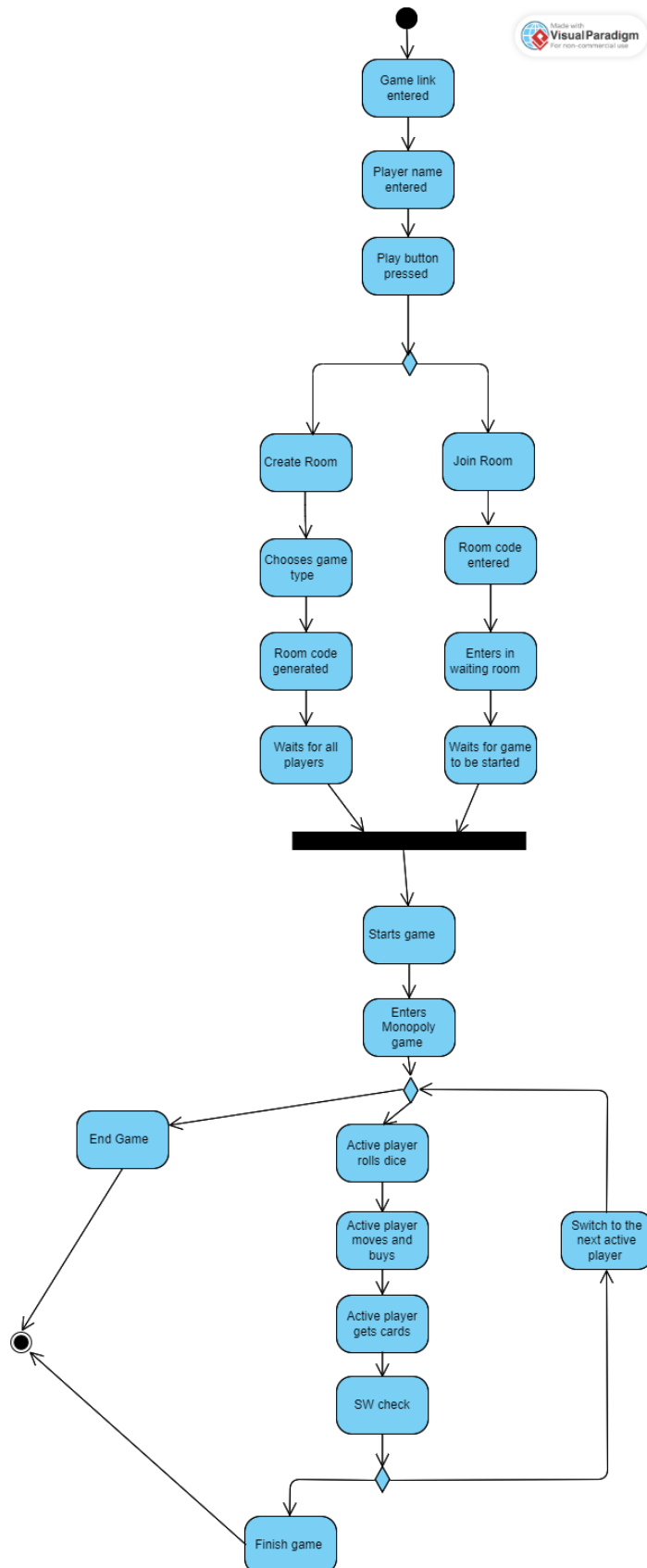
1. Use Case Name: Stay in Jail
2. Actor: Player
3. Preconditions: The player is in jail and has the option to stay in jail during their turn.
4. Trigger: Player chooses to stay in jail during their turn.
5. Main Flow:
 - 5.1. Player goes to jail.
Actor Action: Player chooses to stay in jail.
System Response: The system confirms the player's decision to stay in jail.
6. Postconditions: The player has chosen to stay in jail, and the game state is updated. Control is passed to the next player.
7. Alternative Flow
 - 5.1.a Player gets out of jail
 - 5.1.a.1 Actor Action: The player gets out of jail by various means (e.g., paying a fine or using a "Get Out of Jail Free" card).
 - 5.1.a.2 System Response: The system puts player out of jail.

Use Case Roll Dices

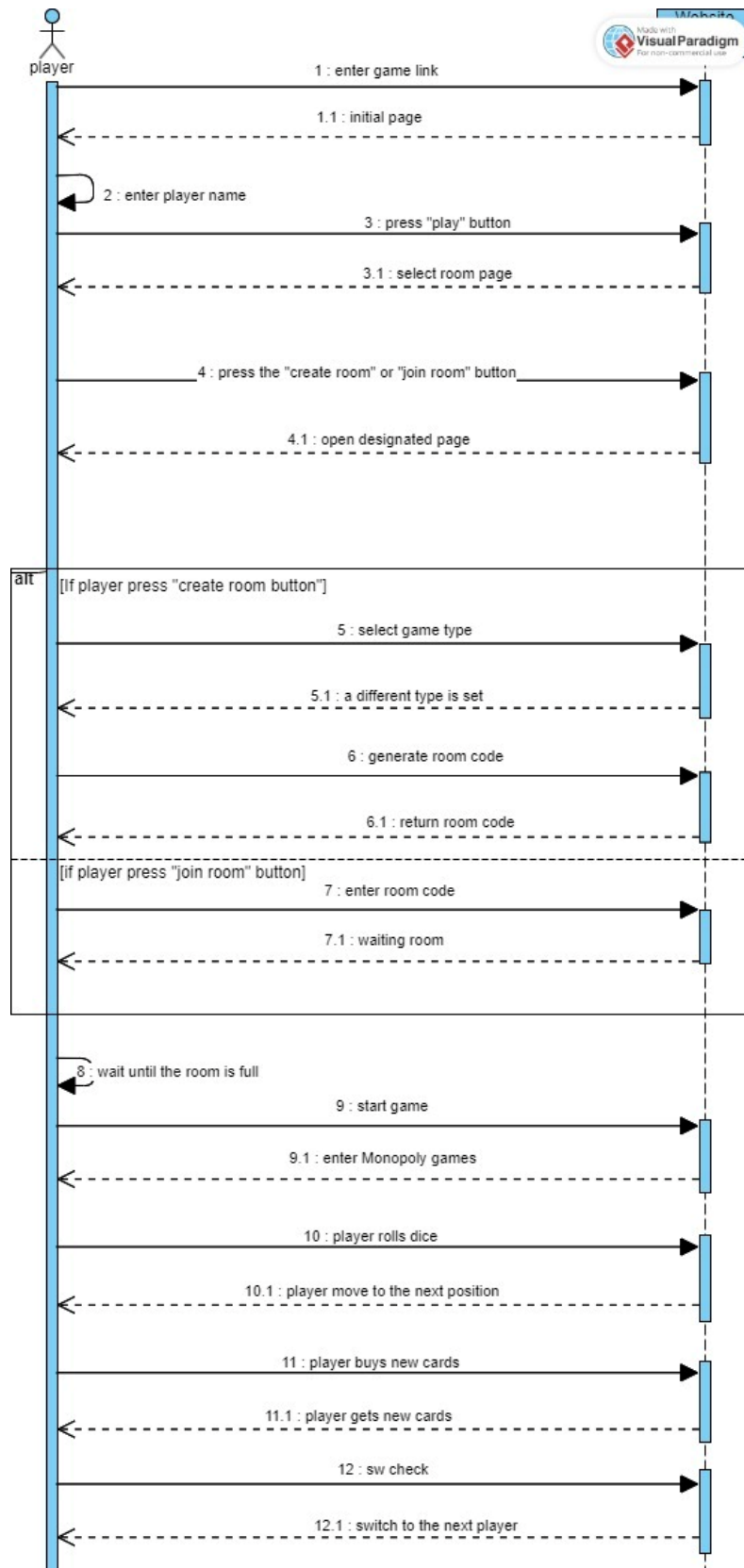
1. Use Case Name: Roll Dice
2. Actor: Player
3. Preconditions: The player's turn has started in an active Monopoly game, and they have the option to roll the dice to determine their move.
4. Trigger: Player rolls the dices during their turn.
5. Main Flow:
 - 5.1. The player rolls the virtual dice.
Actor Action: Player physically or virtually rolls the dice.
System Response: The system calculates the result of the dice roll.
 - 5.2. The turn ends, and control passes to the next player in the game
6. Postconditions: The player has completed their dice roll, and the game state is updated. Control is passed to the next player.
7. Alternative Flow:
 - 5.1.a Rolls double
 - 5.1.a.1 Actor Action: Player rolls doubles.
 - 5.1.a.2 System Response: The system let player rolls again.

Additional Diagrams

Activity Diagram (Elsa Morina)



Sequence Diagram (Domenico Salvatore Maria Panarello)



Mock Ups

