**Problem identification and requirements analysis**

**Case study : Implementation of the Game "Uno" in java**

| Customer | Icesi University CED Block |
| --- | --- |
| User | **People interested in playing "Uno"** |
| Context of the problem | *The project consists of the development of a card game "Uno" in the Java programming language, where various data structures will be used to manage aspects of the game, such as card distribution, player turn and rule verification. . The game will follow standard "Uno" rules, where each player attempts to run out of cards in hand by playing a card that matches in color, number or symbol with the top card in the discard pile. Special cards, such as Color Change, Draw 2, Reverse, and Jump, add complexity to the game and will require careful implementation to ensure they function correctly. The end of the game will be determined when a player runs out of cards in their hand. Additionally, a user interface will be designed that allows players to interact with the game intuitively, showing the cards in each player's hand and the options available on each turn.* |
| Functional requirements | ***RF1 Distribute Cards***  ***RF2 Start the Game***  ***RF3 Determine the Objective of the Game***  ***RF4 Manage Player Turn***  ***RF5 Manage the Deck of Cards***  ***RF6 Check the Rules***  ***RF7 End Game***  ***RF8 Handle Special Cards***  ***RF9 Create the User Interface (UI)*** |
| Non-functional requirements | **Efficiency:** The system must be able to handle a specific number of simultaneous players without experiencing significant performance degradation.  **Security:** Player data, such as names and scores, must be stored securely and protected from unauthorized access.  **Usability:** The user interface should be intuitive and easy to use for players of all ages and experience levels.  **Reliability:** The system must be able to handle unexpected interruptions, such as power outages, without losing data from the game in progress.  **Compatibility:** The game should be compatible with a variety of devices and platforms, including desktop, mobile, and tablets. |

| Identifier and name | ***RF1 Distribute Cards*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement consists of the activity of distributing cards to the players at the beginning of the "Uno" game. To do this, it begins with the initialization and shuffling of the entire deck of cards, followed by the equal division of this deck among the number of participating players. This ensures that each player receives an equal number of cards to start the game. As a result, each player will have a starting set of cards, while the remaining deck becomes the discard deck, with one card face up to start the game. This process ensures that the game begins with a fair distribution of cards among the players and establishes the initial conditions for the development of the game.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| N/A | N/A | | *N/A* |
|  |  | |  |
| Result or Postcondition | Each player has a starting set of cards to start the game.  The remaining deck of cards becomes the discard deck, with one card face up to start the game. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| message | | String | *Cartas == 7* |
|  |  | |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| Identifier and name | ***RF2 Start the Game*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement refers to the process of starting a game of the game "Uno". To start the game, a card is placed face up in the center as the starting card of the discard pile. This starting card is chosen randomly from the deck of cards previously shuffled during the distribution. Choosing this starting card sets the color and starting number for players to continue playing their cards. This process marks the start of the game and establishes the initial conditions so that players can begin playing.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| N/A | N/A | | *N/A* |
|  |  | |  |
| Result or Postcondition | The game is officially in progress and players can begin playing their cards from the initial card placed in the discard pile. This process marks the start of the game and establishes the initial conditions so that players can begin playing. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| message | String | | *N/A* |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| Identifier and name | ***RF3 Determine the Objective of the Game*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement involves continuously verifying if a player has run out of cards in their hand to determine the winner of the game "Uno". The game continues until at least one player runs out of cards in their hand, triggering the end of the game and the determination of the winner. This process ensures that the game's objective, which is to get rid of all cards in hand, is continuously evaluated and that a winner is declared once this condition is met.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| N/A | N/A | | *N/A* |
|  |  | |  |
| Result or Postcondition | When a player runs out of cards in their hand, that player is determined to be the winner of the game. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| message | String | | *N/A* |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| Identifier and name | ***RF4 Manage Player Turn*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement implies that the system handles the player's turn in the game "Uno". In this process, the system verifies if the current player can play a card that matches the top card in the discard pile. If possible, the system allows the player to play the card; otherwise, the player must draw cards from the deck until finding one they can play. Once they find a playable card, the player can immediately play it, and then the turn passes to the next player. This process ensures that each player has the opportunity to actively participate in the game during their turn, allowing them to draw cards until they find one they can play, and ensures that the game rules are followed to maintain its fluidity.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| input | String | | *N/A* |
|  |  | |  |
| Result or Postcondition | The current player has played a valid card from the cards in their hand deck or has drawn cards from the deck until finding a playable card. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| message | String | | *N/A* |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| Identifier and name | ***RF5 Manage the Deck of Cards*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement implies that the system properly manages the deck of cards during the development of the "Uno" game. To achieve this, it begins with initializing the deck using the standard deck of 108 cards and proceeds to shuffle the cards to ensure randomness in distribution. During the distribution phase, the cards are evenly dealt among the players, also setting up the discard pile with an initial card. Throughout the game, card movements occur between the deck of cards and the discard pile, depending on the players' actions, such as playing valid cards or drawing cards from the deck. This ensures proper gameplay development.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| N/A | N/A | | *N/A* |
|  |  | |  |
| Result or Postcondition | The deck of cards is correctly initialized, shuffled, and distributed at the beginning of the game, and cards are moved between the deck of cards and the discard pile according to the players' actions during the game, thus ensuring a proper flow of the game. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| message | String | | *N/A* |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| Identifier and name | ***RF6 Check the Rules*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement implies that the system constantly verifies that players' moves comply with the rules of the "Uno" game and updates the game state accordingly. To achieve this, the system evaluates if each card played by a player matches in color, number, or symbol with the top card in the discard pile. If a player cannot play any valid card, they are prompted to draw cards from the deck until they find a playable card or it passes to the next player. Additionally, the system must also handle the special rules of the game, such as Color Change, Draw 2, Reverse, and Skip cards, adjusting the game flow accordingly. This process ensures that the game progresses according to the established rules and maintains coherence in the game state at all times.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| N/A | N/A | | *N/A* |
|  |  | |  |
| Result or Postcondition | The game continues coherently and progresses according to the established rules after verifying them. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| message | String | | *N/A* |
| message | | String | N/A |
| message | | String | N/A |
|  | |  |  |
|  | |  |  |

| Identifier and name | ***RF7 End Game*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement implies that once a player runs out of cards in their hand, the system must automatically recognize that player as the winner of the "Uno" game, thus marking the end of the game. The system must continuously monitor the state of the players' hands to check if any of them have completely emptied their hand. Once it's detected that a player has run out of cards, that player is automatically declared the winner of the game. This allows players to accurately and timely recognize the winner of the game and signals the appropriate end of the game.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| N/A | N/A | | *N/A* |
|  |  | |  |
| Result or Postcondition | Determines the winner of the game when a player runs out of cards in their hand, thus ensuring the appropriate conclusion of the game. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| message | | String | *N/A* |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| Identifier and name | ***RF8 Handle Special Cards*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement implies that the system properly manages special cards during the "Uno" game. To do this, the system must recognize and apply the rules associated with each type of special card, such as Color Change, Draw 2, Reverse, and Skip. This includes modifying the game flow accordingly when a special card is played, such as changing the game's color, making the next player draw additional cards, reversing the order of play, or causing a player to skip their turn. The system must ensure that special cards are handled correctly to maintain consistency in the game's development and ensure a fair and enjoyable gaming experience for all players.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| N/A | N/A | | *N/A* |
|  |  | |  |
| Result or Postcondition | The rules associated with special cards have been correctly applied, and the game flow has been modified accordingly. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| N/A | N/A | | *N/A* |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| Identifier and name | ***RF9 Create the User Interface (UI)*** | | | |
| --- | --- | --- | --- | --- |
| Abstract | *This requirement involves developing a User Interface (UI) for the "Uno" game, allowing players to interact with the game effectively. The user interface should clearly display the cards in each player's hand and the cards in the discard pile, while also providing interactive options for players to perform actions such as playing a card, drawing a card from the deck, and selecting the color of the card in the case of a Color Change. The interface should be intuitive and easy to understand for players of all ages and levels of experience, ensuring a smooth and enjoyable gaming experience.* | | | |
| Input | **Input name** | **DataType** | | Condition valid values |
| N/A | N/A | | *N/A* |
|  |  | |  |
| Result or Postcondition | The user interface is fully developed and functional, providing players with a smooth and satisfying gaming experience. | | | |
| Outpus | **Output name** | | **DataType** | **Format** |
| message | String | | *N/A* |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |