Name\_\_\_\_\_Bibek Kumar Sharma\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

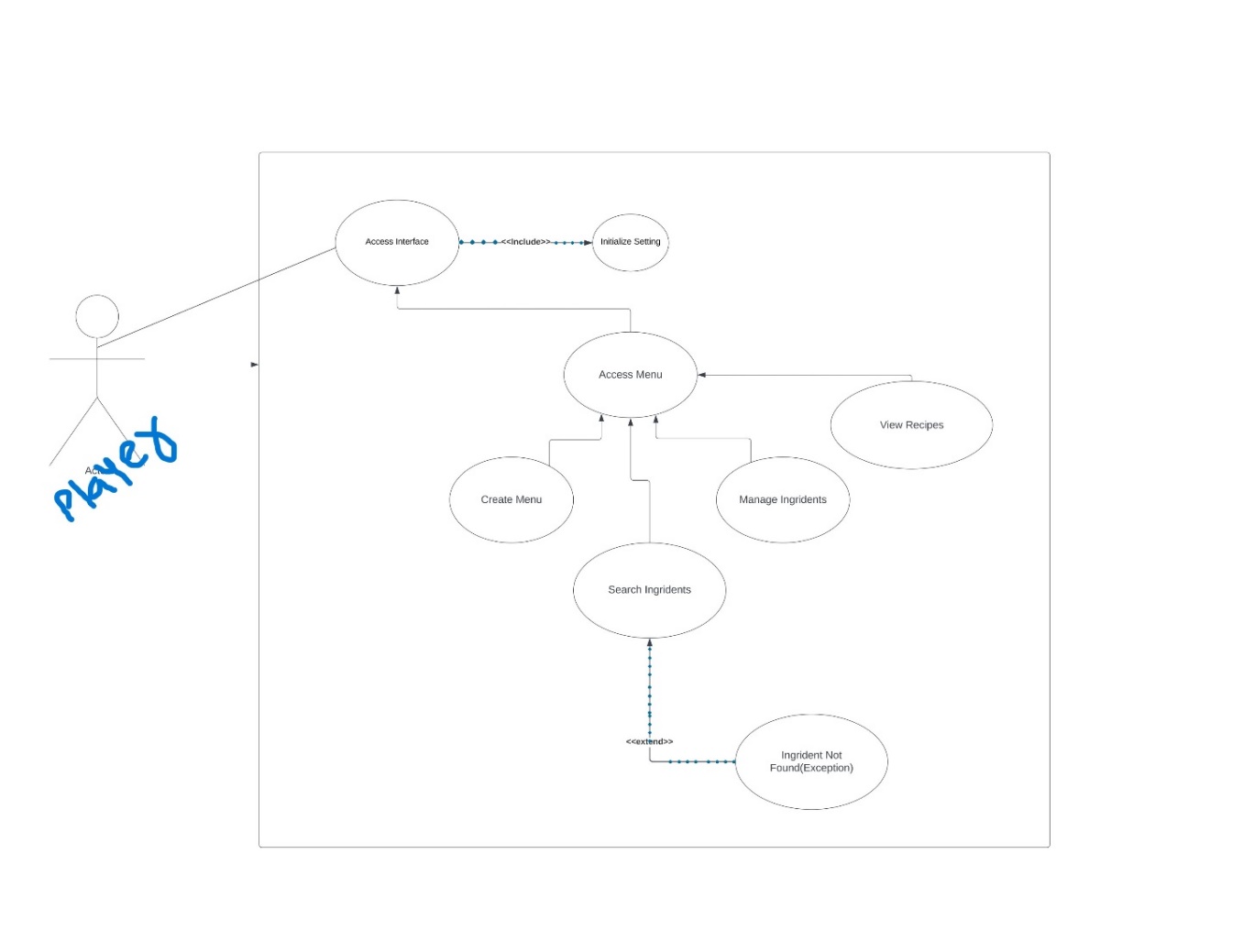
## Brief introduction \_\_/3

My Feature for the Game is to have the Menu,Help and VI feature. Also, to make it more interactive to the player. Our game is a mixture of Diner Dash and PlateUp. For the Menu feature, I will utilise an extensive array of ingredients to create delectable dishes, making sure to cater to various tastes and preferences. As you progress, unlock rare ingredients and recipes, expanding your culinary repertoire and wooing your customers with innovative and tantalising dishes.

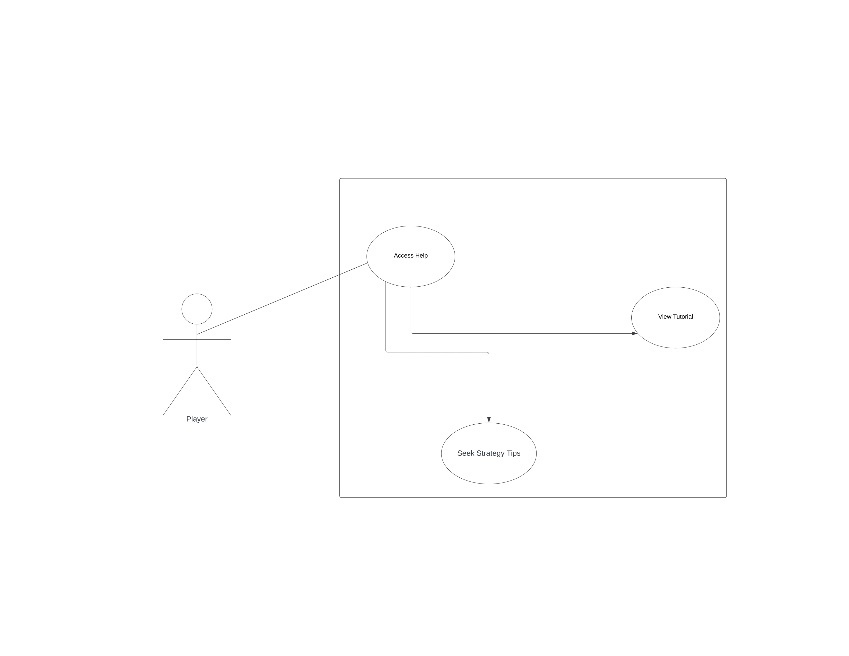
Also, with the Help Feature, navigating through the fast-paced culinary world is no easy feat! Our interactive help system is designed to assist you every step of the way. Whether you're finding the best strategies to serve customers quickly or seeking ways to enhance your culinary skills, the help system will be your ever-reliable guide.

## Use case diagram with scenario \_\_14

### Use Case Diagrams



A diagram of a diagram

Description automatically generated

### Scenarios

**Scenario 1 (1st Use Case Diagram):**

* **Name**: Crafting a Culinary Masterpiece
* **Summary**: The player utilizes the menu feature to craft a variety of dishes using ingredients and recipes available. The system assists in combining them to create a culinary masterpiece which can be served to the customers.
* **Actors**: Player, System (assisting in crafting the dishes)
* **Preconditions**: The player has accessed the menu and chosen the 'create menu' option.
* **Basic Sequence**:
  + **Step 1**: The player chooses a category of dish they want to create (appetizer, main course, dessert).
  + **Step 2**: The system presents a list of available recipes and ingredients.
  + **Step 3**: The player selects a recipe and starts combining ingredients as per the instructions.
  + **Step 4**: The system verifies the combination of ingredients and the method used.
  + **Step 5**: If correctly followed, a culinary masterpiece is created ready to be served to the customers.
* **Exceptions**:
  + **Step 2**: If the player chooses a recipe with missing ingredients.
  + **Step 3**: A prompt is shown to either choose a new recipe or find the missing ingredients.
  + **Step 4**: If an incorrect method is followed, the player is prompted to correct it before proceeding.
* **Postconditions**: A new dish is added to the menu, ready to be served to the customers.
* **Priority**: 1
* **ID**: CM1

**Scenario 2 (2nd Use Case Diagram):**

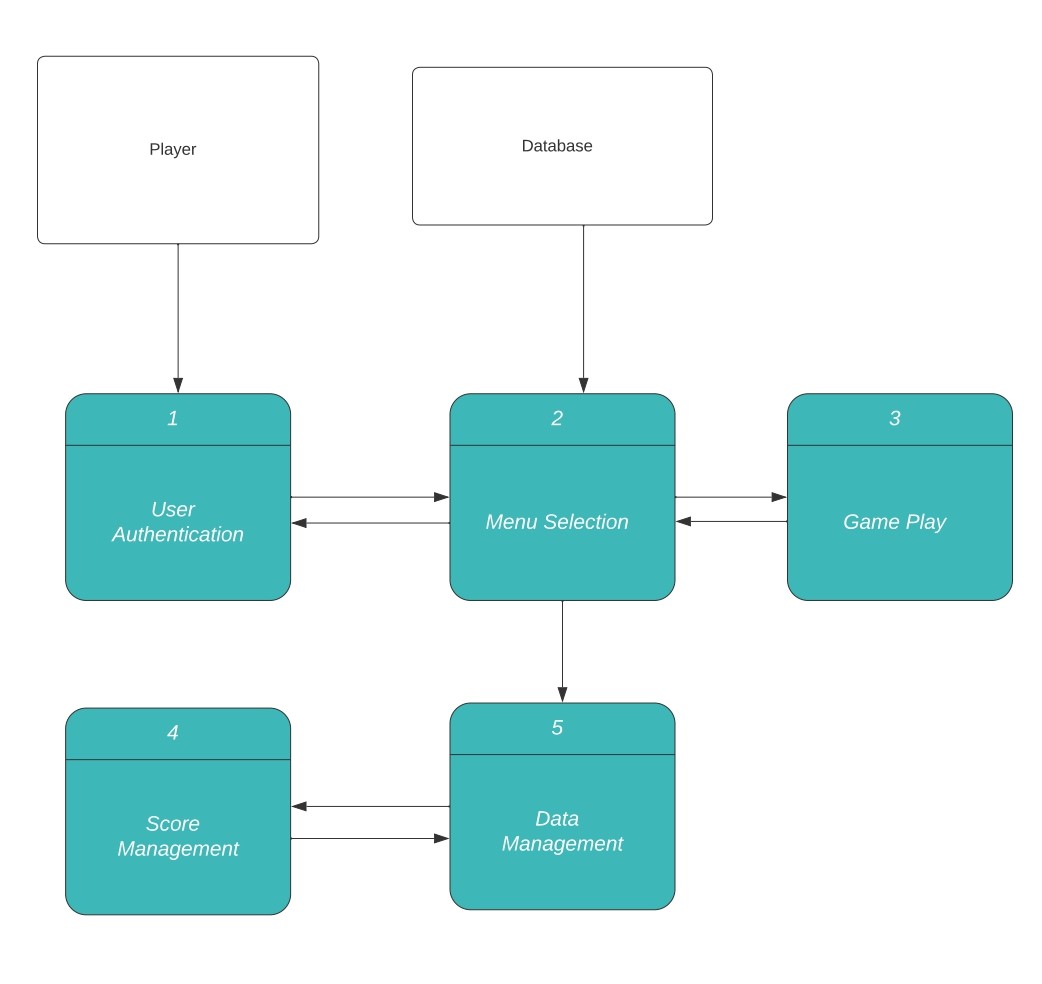
* **Name**: Managing a Busy Dining Area
* **Summary**: The player interacts with the visual interface to manage a busy dining area, ensuring customers are served in a timely manner and the dining area maintains a harmonious environment.
* **Actors**: Player
* **Preconditions**: The dining area has customers, and orders are ready to be served.
* **Basic Sequence**:
  + **Step 1**: The player reviews the pending orders and assigns them to servers.
  + **Step 2**: The servers serve the orders to the respective tables.
  + **Step 3**: The player monitors the satisfaction level of customers.
  + **Step 4**: If satisfied, customers leave making space for new customers.
* **Exceptions**:
  + **Step 3**: If a customer is not satisfied (due to delay or wrong order).
  + **Step 4**: The player needs to take corrective actions like offering a discount or a complimentary item.
* **Postconditions**: The dining area operates smoothly with high customer satisfaction.
* **Priority**: 2
* **ID**: MB1

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

[Get the Level 0 from your team. Highlight the path to your feature]

Example:

### Data Flow Diagrams



### Process Descriptions

1 . User Authentication:

**1.1 Collect User Details**

* **Input**: User information (such as username, password)
* **Process**: The system prompts the user to enter necessary details.
* **Output**: Collected user details ready for verification.

**1.2 Verify User Details**

* **Input**: Collected user details.
* **Process**: The system verifies the entered details against the data stored in the database.
* **Output**: Authentication result (success or failure).

2. Menu Selection:

**2.1 Display Game Menu**

* **Input**: Player's request to view the menu.
* **Process**: The system retrieves the necessary data and displays the game menu to the player.
* **Output**: The visible game menu on the player's screen.

**2.2 User Makes a Selection**

* **Input**: Player's selection/input.
* **Process**: The system captures the player's selection from the displayed menu.
* **Output**: Confirmation of the selection.

**2.3 Load Selected Feature**

* **Input**: Confirmation of selection.
* **Process**: The system loads the necessary components of the selected feature.
* **Output**: The selected feature is ready and displayed for interaction.

3. Game Play:

**3.1 Initialize Game Level**

* **Input**: Player's request to start a game level.
* **Process**: The system initializes the necessary elements to set up the chosen game level.
* **Output**: A fully loaded and initialized game level ready for play.

**3.2 Player Interaction with Game**

* **Input**: Player inputs and actions during gameplay.
* **Process**: The system processes player interactions, updating game states and responding dynamically.
* **Output**: Continuous feedback and updated game states displayed to the player.

**3.3 Progress Monitoring**

* **Input**: Ongoing game data.
* **Process**: The system continuously monitors and tracks the player's progress in the game.
* **Output**: Updated progress data that can be viewed or stored for future reference.

4. Score Management:

**4.1 Calculate Scores**

* **Input**: Data from the player's gameplay (like achievements, time taken, etc.).
* **Process**: The system calculates scores based on predefined criteria and algorithms.
* **Output**: Calculated scores ready for display or storage.

**4.2 Update Score Records**

* **Input**: Calculated scores.
* **Process**: The system updates the score records in the database with the new data.
* **Output**: Updated score records.

**4.3 Display Score Statistics**

* **Input**: Request to view score statistics.
* **Process**: The system retrieves and processes score data to generate statistics.
* **Output**: Display of score statistics to the player.

5. Data Management:

**5.1 Retrieve Necessary Data**

* **Input**: Data request.
* **Process**: The system retrieves the necessary data from the respective data stores.
* **Output**: Retrieved data ready for utilization.

**5.2 Update Game Records**

* **Input**: New game data.
* **Process**: The system updates the game records with the new data.
* **Output**: Updated game records stored in the database.

**5.3 Backup Data**

* **Input**: Command to backup data.
* **Process**: The system creates a backup of the current data stored in the database.
* **Output**: Confirmation of successful data backup.

## Acceptance Tests \_\_\_\_\_\_\_\_9

**Acceptance Test for Scenario 1 (Crafting a Culinary Masterpiece):**

1. **Feature: Crafting a Dish**
   * **Input: Select a category (appetizer, main course, dessert) and choose a recipe.**
   * **Output: A confirmation that the dish has been added to the menu.**
   * **Boundary Conditions:**
     + **The selected recipe must have all ingredients available.**
     + **The method followed must be as per the instructions in the recipe.**
   * **Acceptance Criteria:**
     + **All selected recipes should be added to the menu correctly.**
     + **If an ingredient is missing, a prompt should appear at least 90% of the time.**
     + **If the method is incorrect, a prompt should appear at least 90% of the time.**
2. **Feature: Adding New Ingredients**
   * **Input: Adding new ingredients to the available list.**
   * **Output: The ingredients list gets updated.**
   * **Boundary Conditions:**
     + **The ingredient added must not already be present in the list.**
   * **Acceptance Criteria:**
     + **The ingredient should correctly appear in the updated list at least 95% of the time.**

**Acceptance Test for Scenario 2 (Managing a Busy Dining Area):**

1. **Feature: Assigning Orders to Servers**
   * **Input: Review pending orders and assign them to servers.**
   * **Output: Servers receive the orders to serve to respective tables.**
   * **Boundary Conditions:**
     + **The server assigned should not already be serving another table at the same time.**
     + **The order assigned should match the order placed by the customer.**
   * **Acceptance Criteria:**
     + **Orders should be correctly assigned to servers 98% of the time.**
     + **Orders should match the customer's order 98% of the time.**
2. **Feature: Managing Customer Satisfaction**
   * **Input: Monitoring customer satisfaction levels and taking corrective actions if necessary.**
   * **Output: Satisfied customers or corrective actions taken.**
   * **Boundary Conditions:**
     + **Corrective actions should only be triggered if the customer satisfaction level is below a certain threshold.**
   * **Acceptance Criteria:**
     + **Corrective actions should successfully resolve customer complaints 90% of the time.**
     + **Customer satisfaction level should be correctly monitored 95% of the time.**

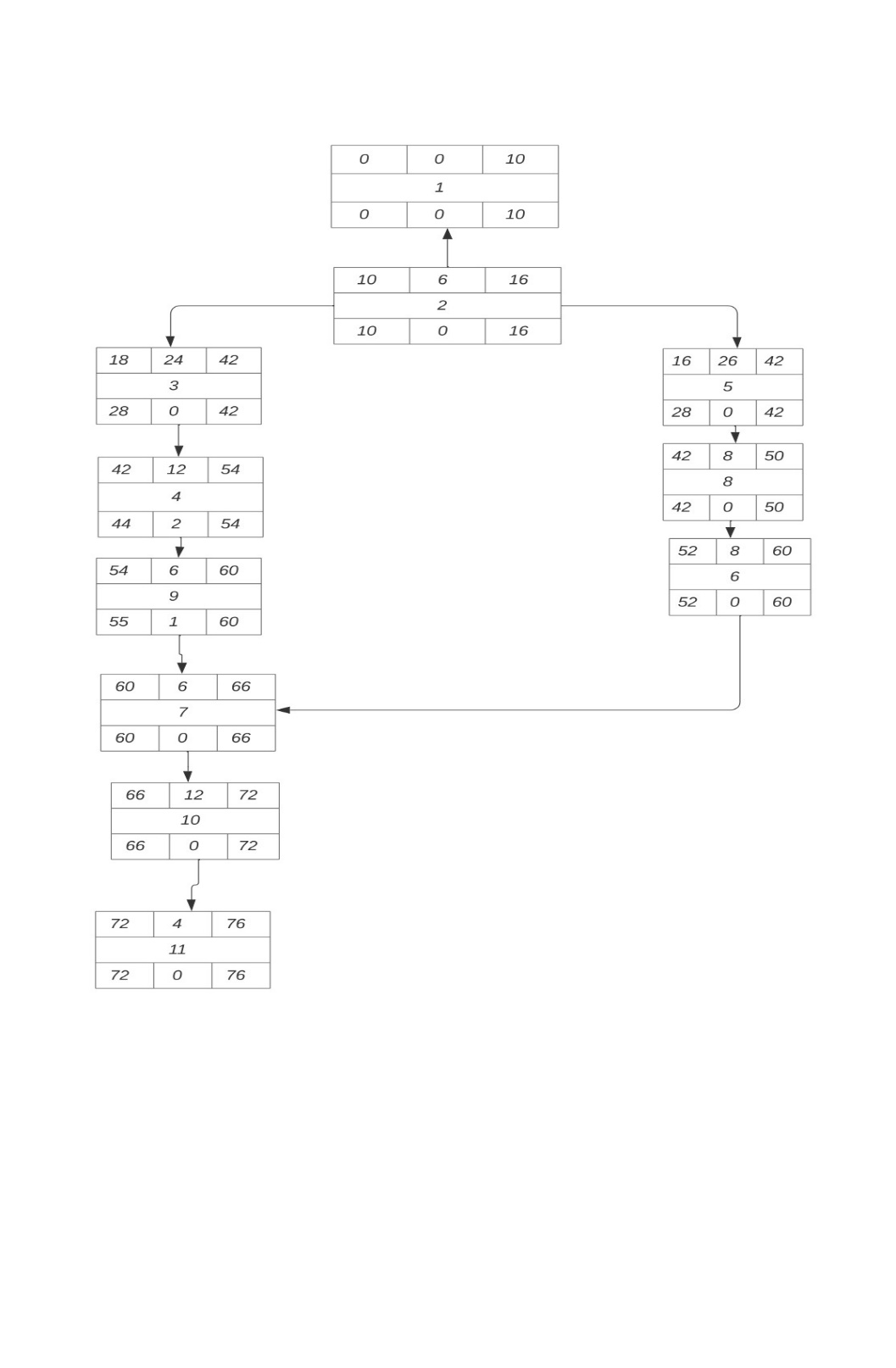
## Timeline \_\_\_\_\_\_\_\_\_/10

[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Database Schema Design for ingredients and recipes | 6 | - |
| 2. Feature Development(Menu crafting and Dining Area Management) | 10 | 1 |
| 3. Ingredient Selection and Recipe crafting Algorithm | 14 | 2 |
| 4. Exception Handling (for crafting errors and customer Issues) | 10 | 3 |
| 5. User Interface Design | 14 | 2 |
| 6. Integration and Functional Testing | 8 | 4,5 |
| 7. Documentation | 6 | 4,6 |
| 8. Artwork and visual Enhancements | 8 | 5 |
| 9. Programming | 5 | 2,3,5 |
| 10. Final Integration and Testing(Including Visual Enhancements) | 6 | 7,8 |
| 11. Deployment and Installation | 5 | 10 |



Here:-

Series1: TaskID

Series2:- Delay Hours

Series3:- Duration