Name_Pramil Shrestha_____ Mark _____/50

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

Brief introduction __/3

I am working in a character and equipment system where players can enter a lobby, manage coins, select characters, equip weapons, and access an in-game shop. The coin management system allows earning and spending on weapons, outfits, and characters. The inventory system lets players view and manage their items, while the weapon equipping feature enhances gameplay strategy. A save and exit function ensures progress is stored. The game's mechanics are supported by use case diagrams, data flow diagrams, and pseudo code, with thorough testing.

Use case diagram with scenario __14

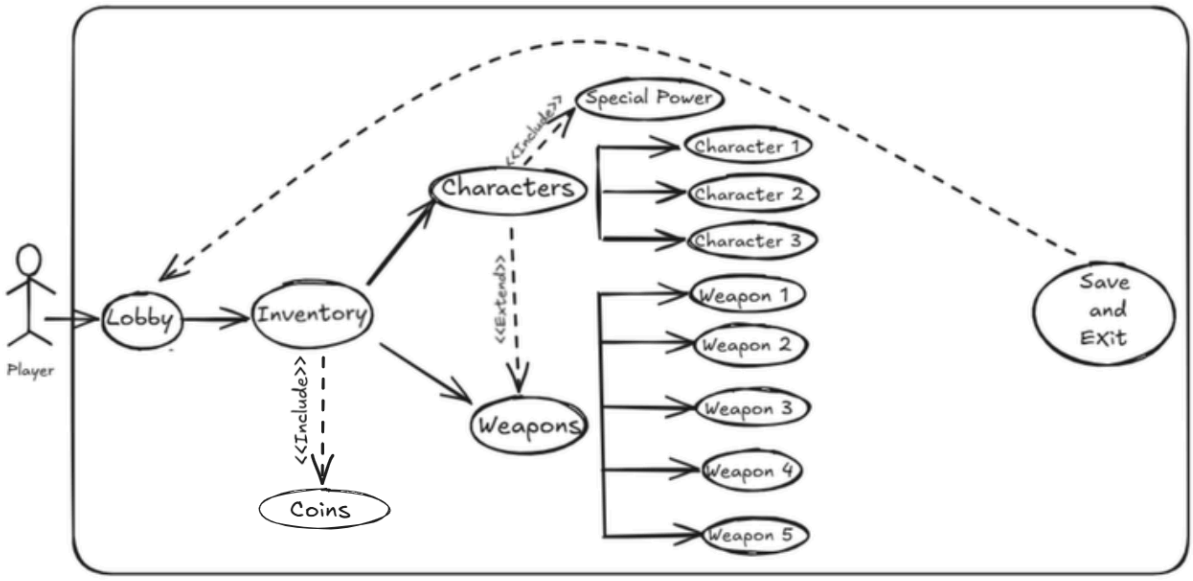
[Use the lecture notes in class.

Ensure you have at least one exception case, and that the <<extend>> matches up with the Exceptions in your scenario, and the Exception step matches your Basic Sequence step.

Also include an <<include>> that is a suitable candidate for dynamic binding]

Example:

Use Case Diagrams



Scenarios

Use Case 1: Enter Lobby

Summary: The player accesses the lobby to begin interacting with the game.

- **Actors:** Player.
- **Preconditions:** The player has logged into the game.
- **Basic Sequence:**
    - The player selects "Enter Lobby" from the main menu.
    - The lobby interface loads successfully.
- **Exceptions:**
    - If the player is not logged in, an error message appears: "Please log in to continue."
- **Postconditions:** The player is in the lobby interface.
- **Priority:** 1
- **ID:** UC01

Use Case 2: Manage Coins

Summary: Players can manage their coins to buy weapons or view their current balance.

- **Actors:** Player.
- **Preconditions:** The player is logged in and has access to their coins.
- **Basic Sequence:**
    - The player earns coins by completing challenges.
    - The player views their coin balance in the lobby or inventory.
    - The player spends coins to purchase weapons.
- **Exceptions:**
    - If the player has insufficient coins, display: "Not enough coins to purchase this weapon."
- **Postconditions:** The updated coin balance is saved.
- **Priority:** 1
- **ID:** UC02

Use Case 3: View Inventory

Summary: The player views items in their inventory.

- **Actors:** Player.
- **Preconditions:** The player is in the lobby.
- **Basic Sequence:**
    - The player selects the "Inventory" option.
    - The system retrieves and displays the inventory items.
- **Exceptions:**
    - If you don’t have enough coins, display: "You can’t buy this Item."

- **Postconditions:** The inventory list is displayed.
- **Priority:** 2
- **ID:** UC03

Use Case 4: Select Characters

Summary: The player selects a character to use.

- **Actors:** Player.
- **Preconditions:** The player is in the inventory.
- **Basic Sequence:**
    - The player selects "Characters."
    - A list of available characters is displayed.
    - The player selects a character.
    - The selected character is activated.
- **Exceptions:**
    - If no characters are available, display: "No characters unlocked. Please unlock a character."
- **Postconditions:** The selected character is ready for use.
- **Priority:** 1
- **ID:** UC04

Use Case 5: Equip Weapons

Summary: The player equips weapons to characters.

- **Actors:** Player.
- **Preconditions:** The player has accessed the weapons menu and selected a character.
- **Basic Sequence:**
    - The player selects "Weapons."
    - A list of available weapons is displayed.
    - The player selects a weapon.
    - The weapon is equipped to the selected character.
- **Exceptions:**
    - If you don’t have enough coins available, display: "You can’t buy this weapon."
- **Postconditions:** The weapon is equipped to the character.
- **Priority:** 2
- **ID:** UC05

Use Case 6: Save and Exit

Summary: The player saves progress and exits the game.

Actors: Player.

Preconditions: The player has completed actions and wants to leave the session.

Basic Sequence:

a. The player selects "Save and Exit."
b. The system saves all progress.
c. The game session ends.

Exceptions:

d. If saving fails, display: "Save failed. Please try again."

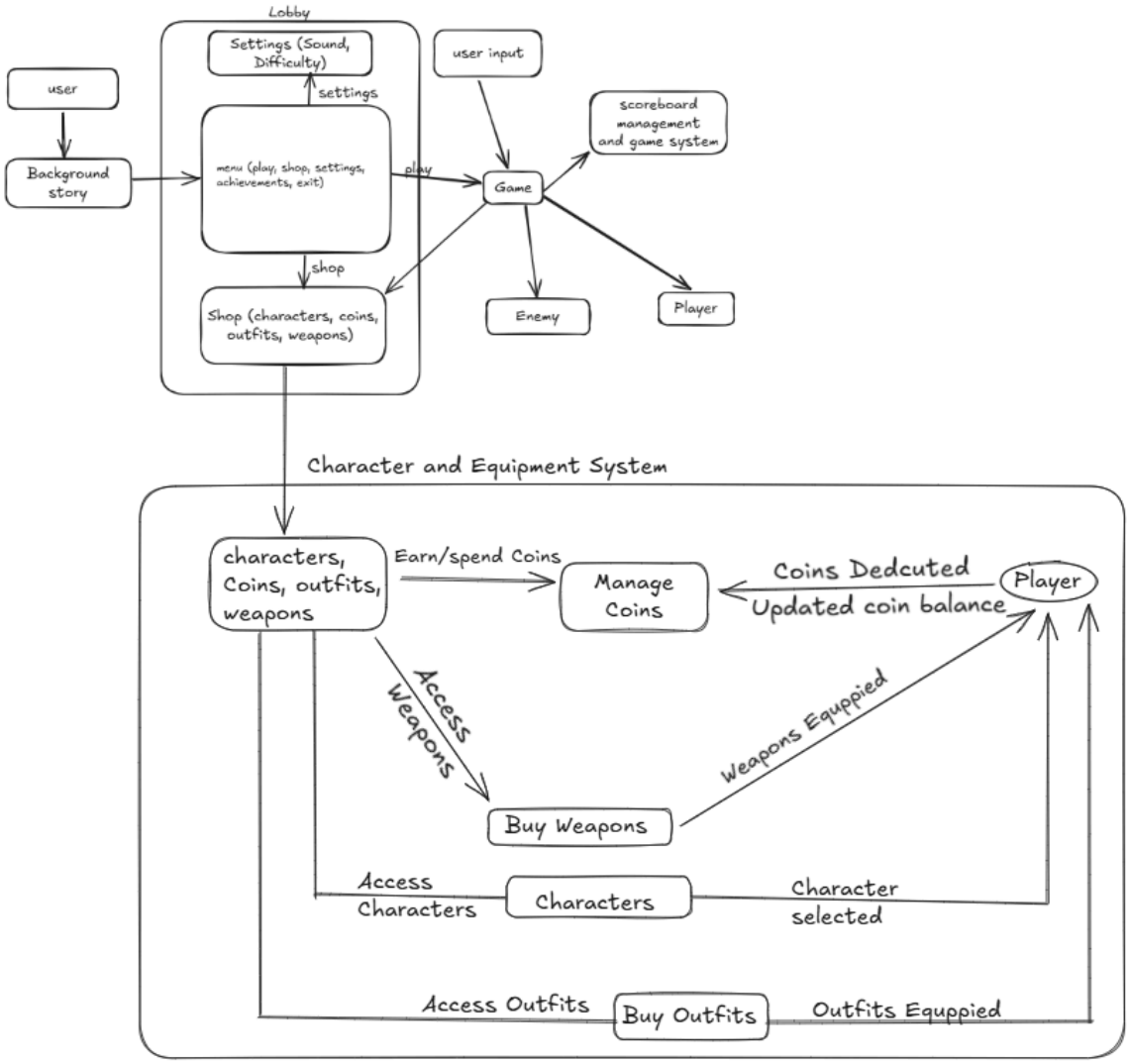
Postconditions: The player's progress is saved, and the session is closed.

Priority: 1

ID: UC06

Data Flow diagram(s) from Level 0 to process description for your feature _____14

Data Flow Diagrams



Process Descriptions

Process Descriptions

The shop displays available items (e.g., Characters, Coins, Outfits, Weapons). Routes the player's action to the appropriate purchase submenu or action.

If the player chooses Manage Coins, it will allow the player to earn coins through gameplay or purchase coins using real money. Updates the player's coin balance after the transaction.

If the player chooses Buy Weapons, it will deduct the required number of coins and add the selected weapon to the player's inventory. Updates the inventory and the coin balance after purchase.

If the player chooses Buy Outfits, it will deduct the required number of coins and equip the purchased outfit to the selected character. Updates the inventory and the coin balance after purchase.

If the player selects Characters, it will display a list of available characters. Allows the player to select a character, which will then be updated and equipped for gameplay.

All purchases or updates made by the player will automatically update the player's profile, ensuring the correct coin balance, equipped weapons, and outfits are saved.

Pseudo code:

1. Main shop menu:

```
FUNCTION ShopMenu()
  DISPLAY "Select an option: Manage Coins, Buy Weapons, Buy Outfits, Characters"
  INPUT playerChoice
  IF playerChoice == "Manage Coins" THEN
    CALL ManageCoins()
  ELSE IF playerChoice == "Buy Weapons" THEN
    CALL BuyWeapons()
  ELSE IF playerChoice == "Buy Outfits" THEN
    CALL BuyOutfits()
  ELSE IF playerChoice == "Characters" THEN
    CALL ManageCharacters()
  ELSE
    DISPLAY "Invalid Choice"
  END IF
END FUNCTION
```

2. Manage coins:

```
FUNCTION ManageCoins()
    DISPLAY "Earn or Purchase Coins"
    INPUT coinOption

    IF coinOption == "Earn" THEN
        coinsEarned = CALCULATE_EARNED_COINS()
        UPDATE playerCoinBalance WITH coinsEarned
        DISPLAY "Coins added: " + coinsEarned
    ELSE IF coinOption == "Purchase" THEN
        DISPLAY "Enter amount to purchase"
        INPUT amount
        coinsPurchased = CALCULATE_PURCHASED_COINS(amount)
        UPDATE playerCoinBalance WITH coinsPurchased
        DISPLAY "Coins purchased: " + coinsPurchased
    ELSE
        DISPLAY "Invalid Option"
    END IF
END FUNCTION
```

3. Buy Weapons:

```
FUNCTION BuyWeapons()
    DISPLAY "Available Weapons: "
    DISPLAY ListOfWeapons()
    INPUT selectedWeapon

    weaponCost = GET_COST(selectedWeapon)
```

```
    IF playerCoinBalance >= weaponCost THEN
        UPDATE playerCoinBalance = playerCoinBalance - weaponCost
        ADD selectedWeapon TO playerInventory
        DISPLAY "Weapon purchased and added to inventory."
    ELSE
        DISPLAY "Not enough coins to purchase this weapon."
    END IF
END FUNCTION
```

4. Buy Outfits

```
FUNCTION BuyOutfits()
    DISPLAY "Available Outfits: "
    DISPLAY ListOfOutfits()
    INPUT selectedOutfit

    outfitCost = GET_COST(selectedOutfit)
    IF playerCoinBalance >= outfitCost THEN
        UPDATE playerCoinBalance = playerCoinBalance - outfitCost
        EQUIP selectedOutfit TO selectedCharacter
        DISPLAY "Outfit purchased and equipped."
    ELSE
        DISPLAY "Not enough coins to purchase this outfit."
    END IF
END FUNCTION
```

5. Manage Characters

```
FUNCTION ManageCharacters()
    DISPLAY "Available Characters: "
```

```
    DISPLAY ListOfCharacters()
    INPUT selectedCharacter

    IF selectedCharacter IN UnlockedCharacters THEN
        UPDATE activeCharacter = selectedCharacter
        DISPLAY "Character selected and ready for gameplay."
    ELSE
        DISPLAY "Character is locked. Unlock it to use."
    END IF
END FUNCTION
```

6. Profile Updates

```
FUNCTION UpdatePlayerProfile()
    SAVE playerCoinBalance
    SAVE playerInventory
    SAVE equippedWeapons
    SAVE activeCharacter
    DISPLAY "Profile updated successfully."
END FUNCTION
```

Acceptance Tests _____9

Feature	Input	Output	Boundary Cases	Test Scenarios	Expected
Manage Coins	Earn Coins	Updated coin balance	Earning zero coins	1.Earn 50 coins 2. Attempts to purchase	1. Balance increases by 50. 2.Error:purch

				with insufficient coins.	ase failed.
Buy Weapons	selected weapons, current coin balance	Weapon added, coins deducted	Buying with exact coins, insufficient coins	1.Buy a weapon costing 50 coins with 50 coins. 2.Buy a weapon with insufficient coins.	1.Wepaon added, balance is 0. 2.Error:Not enough coins
Buy Outfits	Selected outfit, current balance	Outfit equipped, coins deducted	Buying with exact coins, equipping an outfit to a locked character	1.Buying an outfit costing 50 coins with 50 coins 2.Equip an outfit to a locked character	1.Outfit equipped, balance is 0. 2.Error:chara cter is locked.
Manage characters	Selected character	Active Character Updated	Selecting a locked character	1.Select an unlocked character. 2.Select a locked character.	1.Character updated successfully. 2.Error:Chara cter is locked.

Timeline _____/10

[Figure out the tasks required to complete your feature]

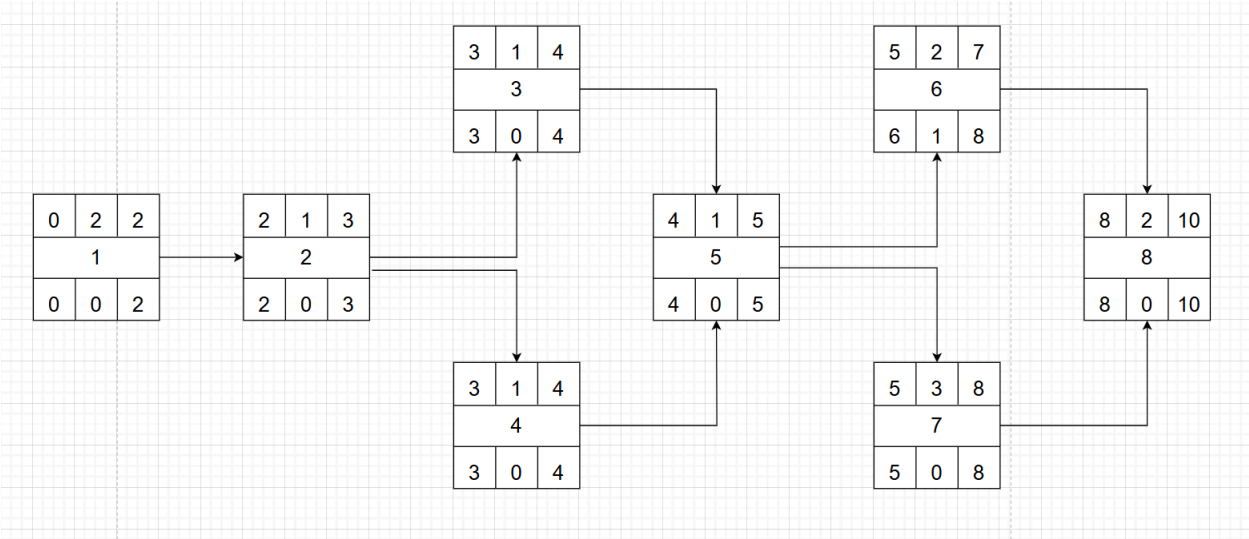
Example:

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Character designs	2	-
2. Special power designs	1	1
3. Weapons designs	1	-
4. Coins	1	-

5. outfit designs	1	1
6. Shop designs	2	1,2,3,4,5
7. Testing	3	6
8. Documentation	2	6, 7

Pert diagram



Gantt Diagram