Name: Daniel Franks Mark _____/50

1. Brief introduction _/3

I will be designing the shop system, which will allow players to “purchase” items from a variety of shopkeepers to improve their character in different ways.

2. Use case diagram with scenario _14

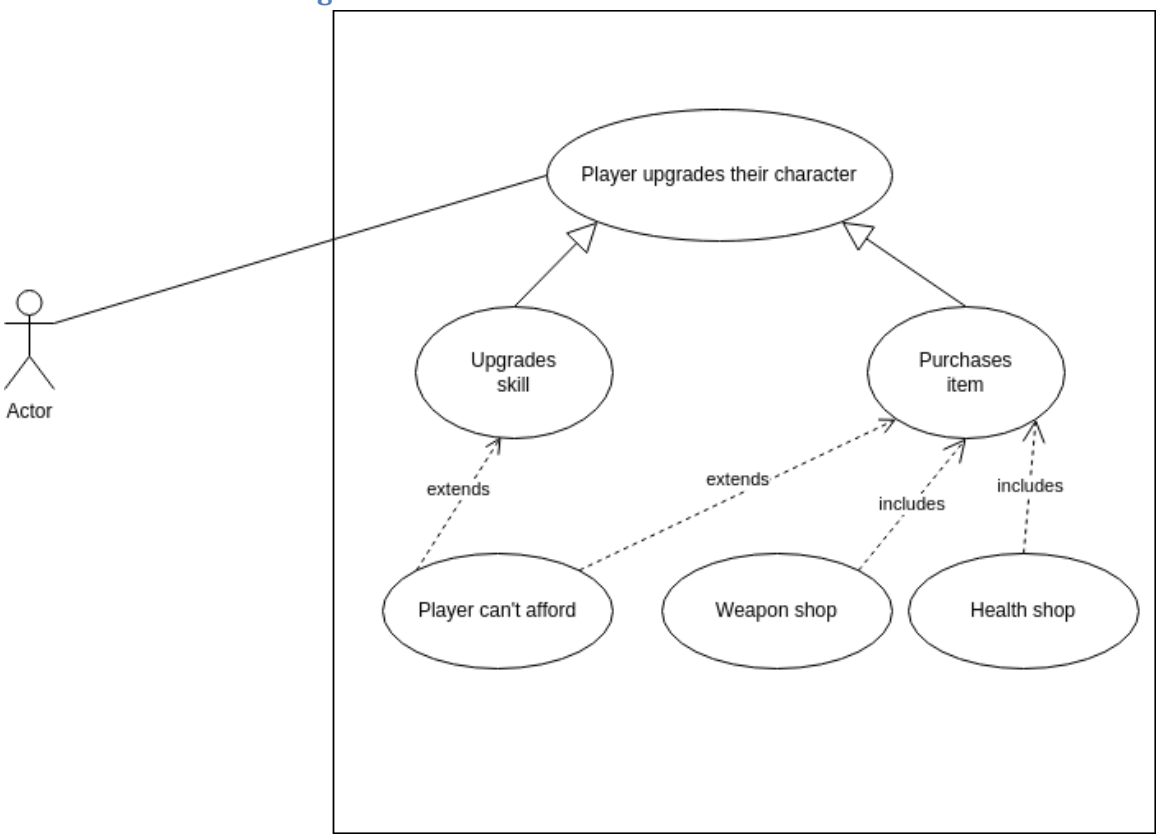
[Use the lecture notes in class.

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

Name: Purchase Item

Summary: The player buys an item from one of the in game shopkeepers.

Actors: Player

Preconditions: Player is in a game session (ie, not in the home)

Basic sequence:

Step 1: Player initiates interaction with shopkeeper.
Step 2: Shopkeeper UI comes up (game is paused? TBD)
Step 3: Player selects their preferred item
Step 4: Item is applied to player (the exact meaning of this varies depending on the item) and value of item is deducted from player's cash.
Step 5: Player exits the shopkeeper UI.

Exceptions:

Step 3: User exits without purchasing anything: don't apply anything to them
Step 4.1: Item cannot be applied to player: display a message to that effect, refund them.
Step 4.2: Player cannot afford item: display a message to that effect, do not apply item to player, do not charge player.

Post conditions: Player has item applied to them.
Priority: 2*
ID: C01

Name: Obtain Skill
Summary: The player obtains a skill in the at-home skill tree.
Actors: Player
Preconditions: Player is at home
Basic sequence:

Step 1: Player initiates interaction with skill tree
Step 2: Skill tree UI comes up (game is paused? TBD)
Step 3: Player selects their preferred skill
Step 4: Skill is applied to player and skill points are deducted from player.
Step 5: Player exits the skill tree UI.

Exceptions:

Step 3: User exits without selecting skill: don't apply anything to them
Step 4.1: Skill cannot be applied to player: display a message to that effect, refund them.
Step 4.2: Player does not have enough skill points: display a message to that effect, do not apply skill to player, do not charge player.

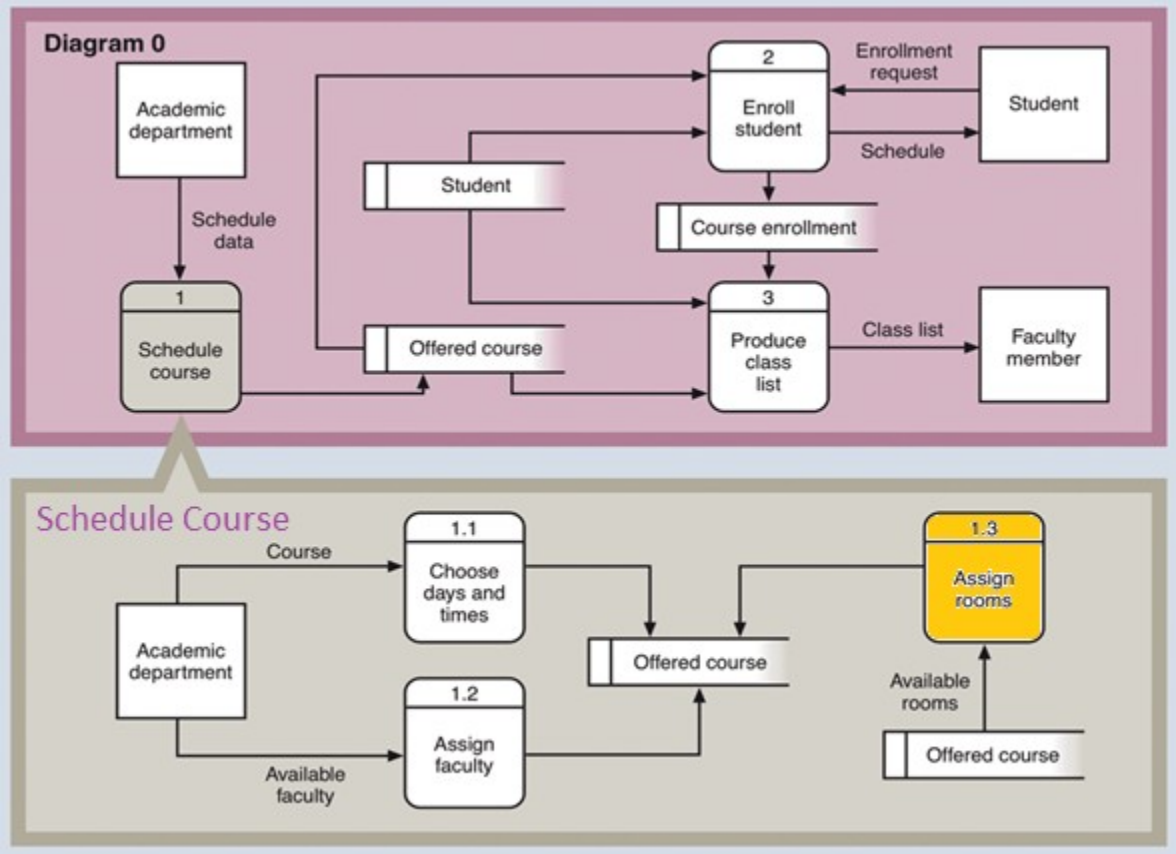
Post conditions: Player has skill applied to them.
Priority: 3*
ID: C02
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. 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

Assign rooms*:

WHILE teacher in two places at once OR two classes in the same room
Randomly redistribute classes
END WHILE

***Notes**: Yours should be much longer. You could use a decision tree or decision table instead if it is more appropriate.

4. Acceptance Tests _____9

Shop

Open Shop UI
Press Down arrow 99 times to attempt to navigate past the end
Give the player enough money to buy a basic item 5 times
Attempt to buy it 10 times
Player should have that perk only 5 times
Verify that perk works as expected (ie: stacks)

Skill Tree

Open Skill Tree UI
Press Down arrow 99 times to try to navigate past the end
Try to obtain a skill the player already has
    verify that it didn't work and didn't use skill points
Try to obtain a locked skill
    verify that it didn't work and didn't use skill points
Obtain a new skill
    verify that it worked and used skill points

5. Timeline _____/10

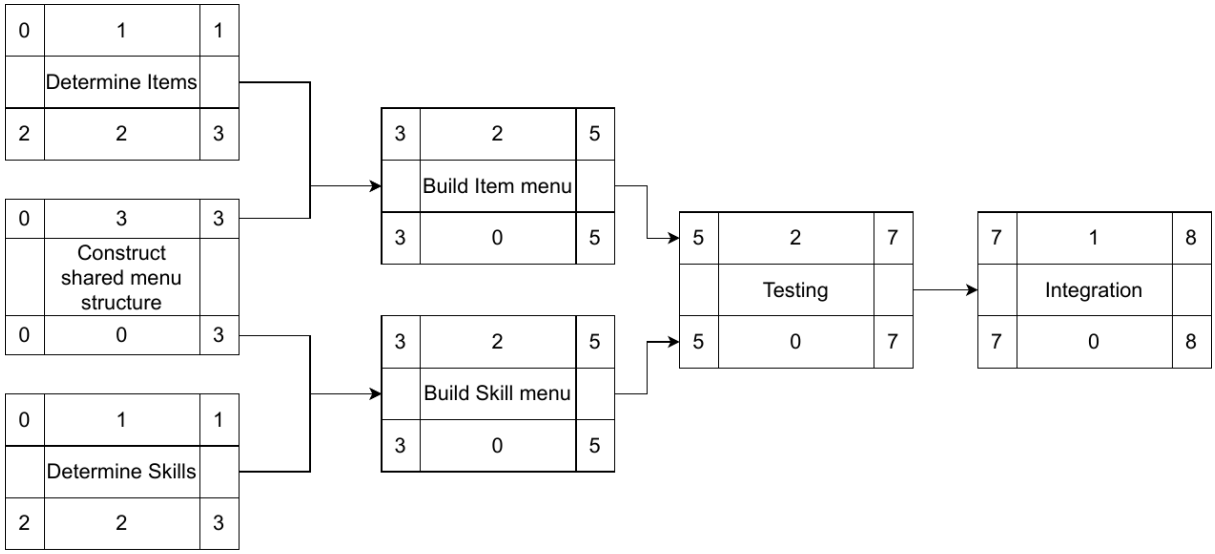
[Figure out the tasks required to complete your feature]

Example:

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Determine items	1	-
2. Determine skills	1	-
3. Construct shared underlying menu structure	3	-
4. Construct item menu	2	1, 3
5. Construct skill menu	2	2, 3
6. Testing	2	4.5
7. Integration	1	6

Pert diagram



Gantt timeline