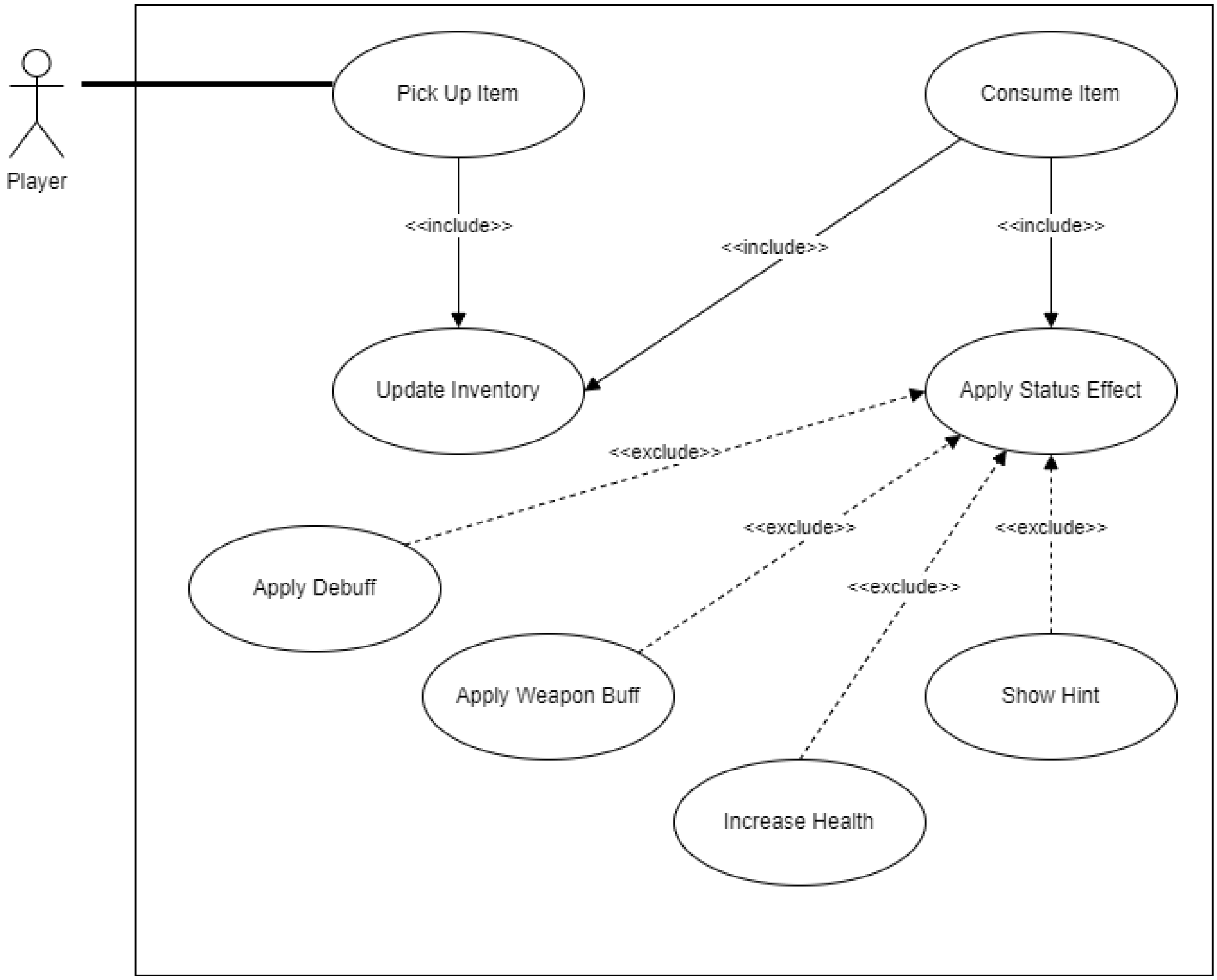
Name: Cole Paulik Mark _______________________/50

1. Brief introduction __/3

As the user plays the game, they will pick up items like ingredients and weapons that will be used to help them progress through the game. The user will also pick up powerups that will aid them in making it through the game farther by healing any damage they may have taken.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

Name: Pick Up Item

Summary: The user picks up an item and stores it in their inventory. From the inventory the player will be able to consume the item. The item will apply an effect such as buffing weapon damage, increasing the players health, apply a debuff to the player, or show hints on what ingredients the player needs to grab.

Actors: Player

Preconditions: Items have spawned in the room.

Basic sequence:

Step 1: The Player enters a room

Step 2: The player picks up an item off the ground

Step 3: The player consumes the item from their inventory

Step 4: The item applies an effect to the player

Exceptions:

Step 1: The item applies a weapon buff, increases player health, applies a debuff to the player, or gives a hint to the player.

Post conditions: The item is now in the inventory or consumed and the player can pick up more items.

Priority: 2*

ID: I01

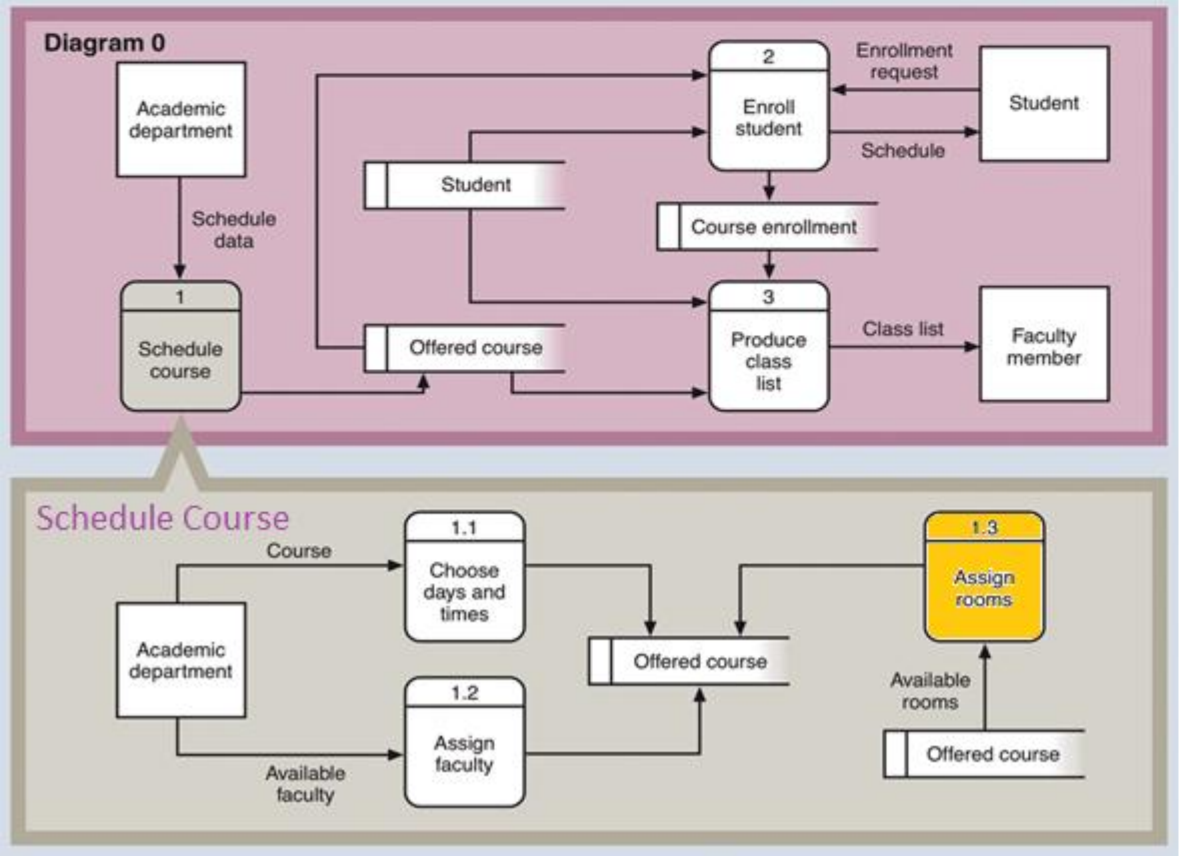
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

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

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

Example for random number generator feature

Run feature 1000 times sending output to a file.

The output file will have the following characteristics:

- Max number: 9
- Min number: 0
- Each digit between 0 and 9 appears at least 50 times

- No digit between 0 and 9 appears more than 300 times
- Consider each set of 10 consecutive outputs as a substring of the entire output. No substring may appear more than 3 times.

Example for divide feature

Output	Numerator (int)	Denominator (int)	Notes
0.5	1	2	
0.5	2	3	We only have 1 bit precision for outputs. Round all values to the nearest .5
0.0	1	4	At the 0.25 mark always round to the nearest whole integer
1.0	3	4	At the 0.75 mark always round to the nearest whole integer
255.5	5	0	On divide by 0, do not flag an error. Simply return our MAX_VAL which is 255.5.

5. Timeline ________/10

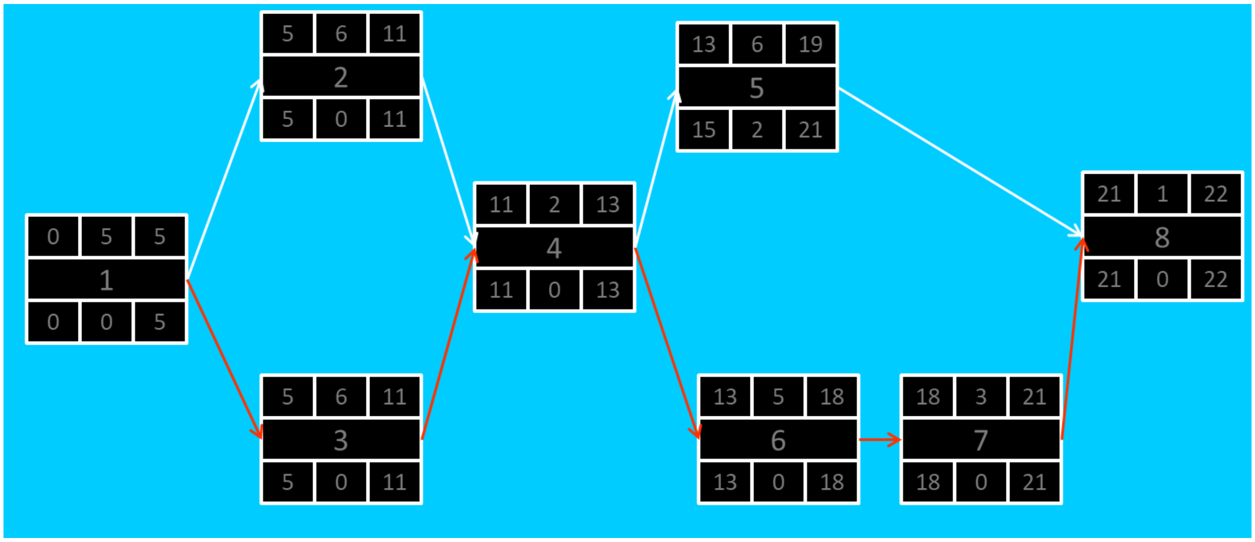
[Figure out the tasks required to complete your feature]

Example:

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Requirements Collection	5	-
2. Screen Design	6	1
3. Report Design	6	1
4. Database Construction	2	2, 3
5. User Documentation	6	4
6. Programming	5	4
7. Testing	3	6
8. Installation	1	5, 7

Pert diagram



Gantt timeline

1	🟥	🟥	🟥	🟥	🟥																			
2						🟦 1	🟦	🟦	🟦	🟦	🟦													
3						🟥 1	🟥	🟥	🟥	🟥	🟥													
4												🟥 3	🟥	🟥	🟥									
5																🟦 4	🟦	🟦	🟦	🟦	🟦	▨	▨	
6																🟥 4	🟥	🟥	🟥	🟥				
7																					🟥 6	🟥	🟥	
8																								🟥 7
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24