Name Nyah Nelson Mark _______________/50

NAME IT WITH FIRST TWO INITIALS

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

1. Brief introduction __/3

[Describe your feature briefly]

- User interface
- Inventory
- Player profile

1. Use case diagram with scenario __14

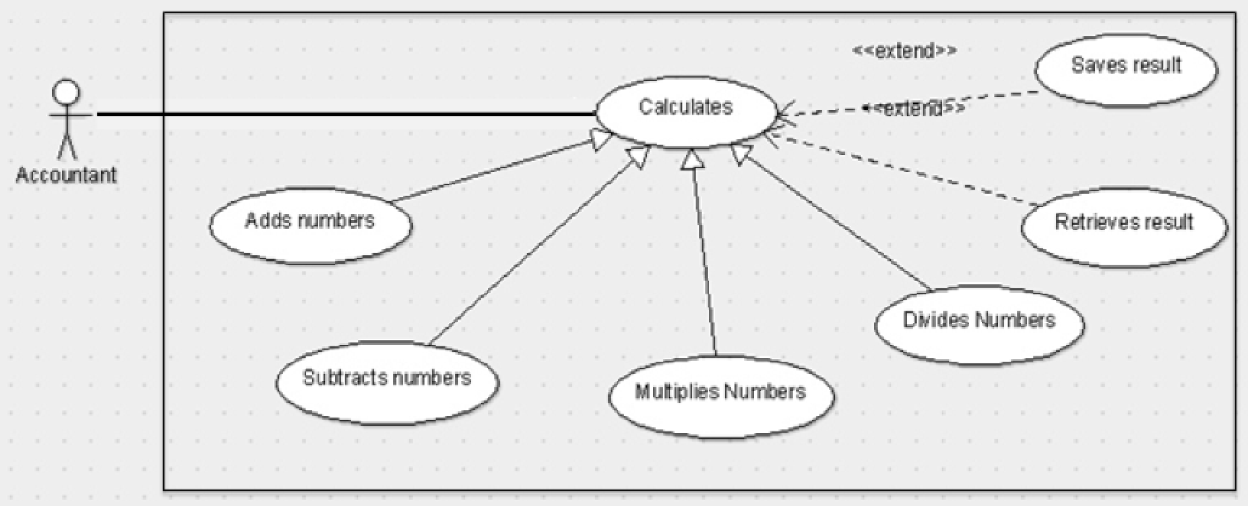
[Use the lecture notes in class.

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

[You will need a scenario for each use case]

Name: Add Numbers
Summary: The accountant uses the machine to calculate the sum of two numbers.
Actors: Accountant.
Preconditions: Calculator has been initialized.
Basic sequence:

Step 1: Accept input of first number.
Step 2: Continue to accept numbers until [calculate] is entered.
Step 3: Accept calculate command.
Step 4: Calculate and show result.

Exceptions:

Step 1: [calculate] is pressed before any input: Display 0.
Step 2: A button other than [calculate] or a number input is pressed: ignore input.

Post conditions: Game object moves correctly in response to user input.
Priority: 2*
ID: C01
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: User Interface
Summary: The player uses the touchpad and keyboard to interact with the game.
Actors: Player
Preconditions: The keyboard and touchpad are programmed.
Basic sequence:

Step 1: Accept input of device (keyboard or mouse/touchpad).
Step 2: Process input and display menu (if menu button is clicked) or move player object (if correct keys are pressed).

Exceptions:

Step 1: A key on the keyboard that is not programmed is pressed: ignore input (do nothing)
Step 1: The mouse/touchpad clicks on something other than a button: ignore input (do nothing)

Post conditions: Game object moves correctly in response to user input.
Priority: 1*

ID: C01
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Inventory
Summary: The player uses the backpack for inventory.
Actors: Player
Preconditions: the "backpack" is initialized and empty at the beginning of the game
Basic sequence:

Step 1: Collect weapons.
Step 2: Store weapons in inventory.
Step 3: Use weapons.
Step 4: Keep track of which weapons have been used in the inventory.

Exceptions:

Step 3: Attempt to use a weapon that is not in the inventory.

Post conditions: Chosen weapon is used and inventory is up to date.
Priority: 2*
ID: C01
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Player Profile
Summary: The accountant uses the machine to calculate the sum of two numbers.
Actors: Accountant.
Preconditions: Calculator has been initialized.
Basic sequence:

Step 1: Accept input of first number.
Step 2: Continue to accept numbers until [calculate] is entered.
Step 3: Accept calculate command.
Step 4: Calculate and show result.

Exceptions:

Step 1: [calculate] is pressed before any input: Display 0.
Step 2: A button other than [calculate] or a number input is pressed: ignore input.

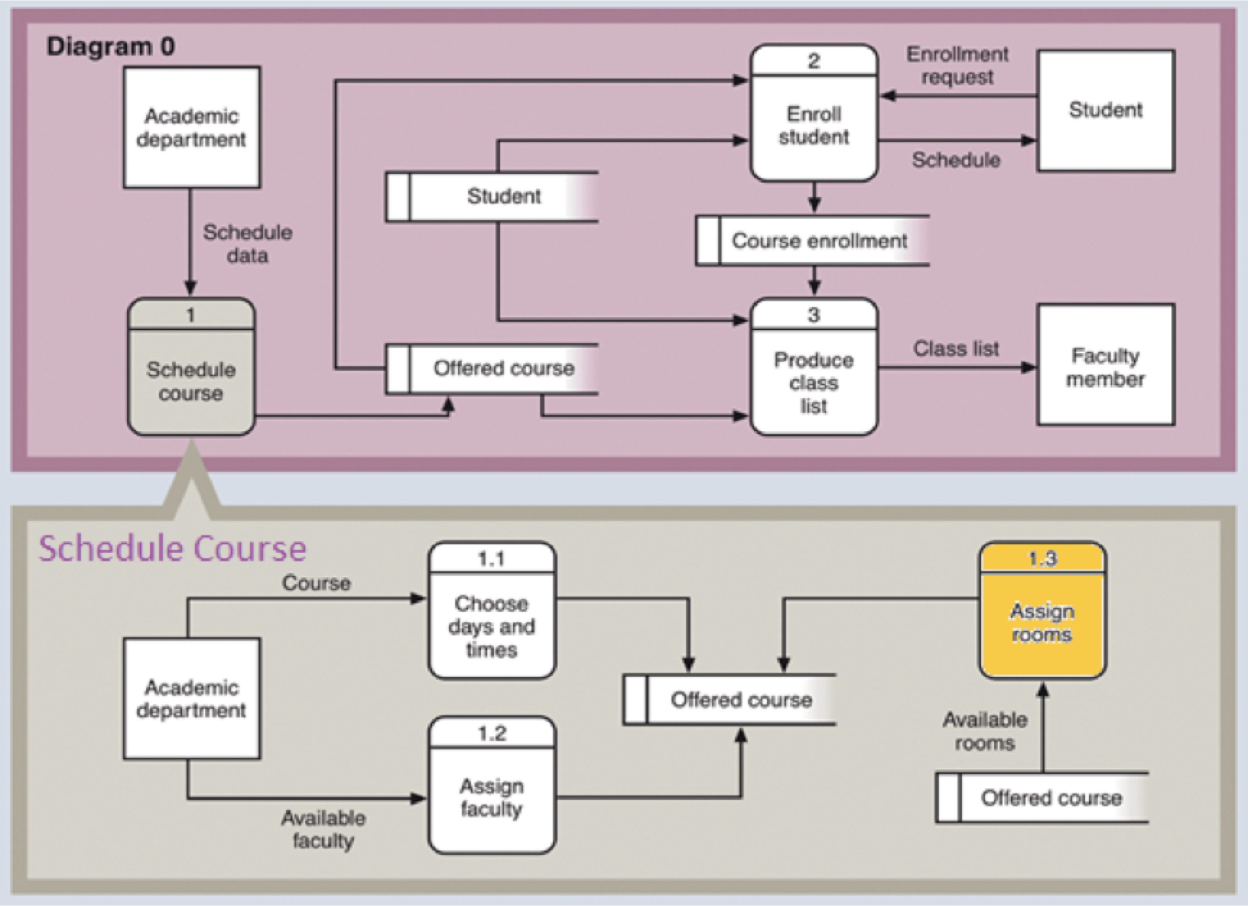
Post conditions: Calculated value is displayed.
Priority: 3*
ID: C01
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

1. Data Flow diagram(s) from Level 0 to process description for your feature _______14

[Get the Level 0 (context diagram) 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.

1. 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.

1. Timeline ________/10

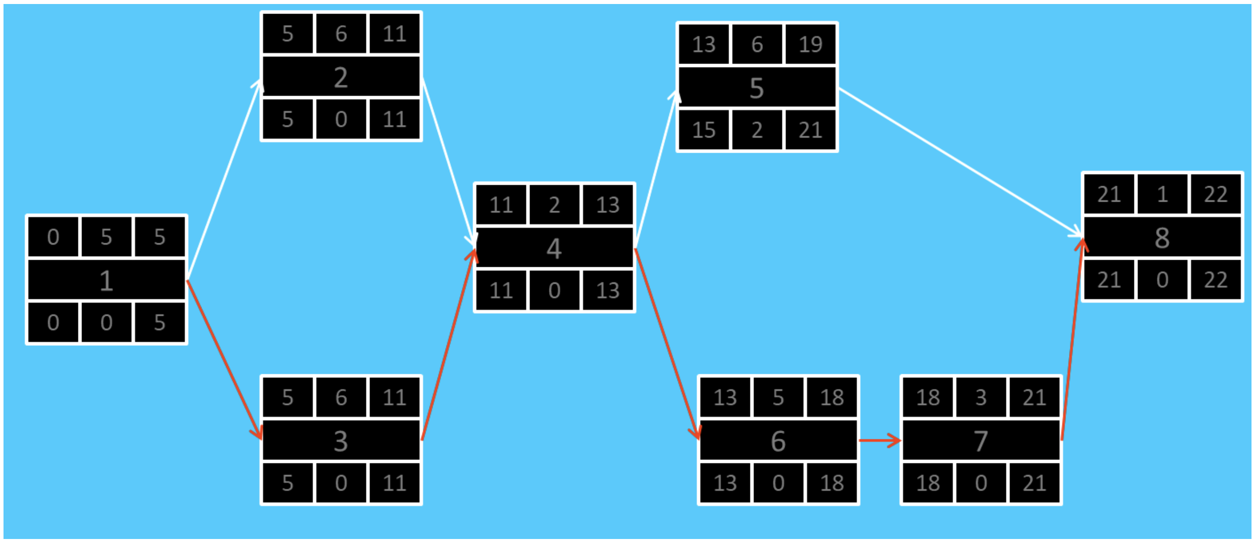
[Figure out the tasks required to complete your feature]

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