



Web Project

Documentation



NOVEMBER 11, 2023

CAPSTONE PROJECT

Curtin University – Bachelor of Software Engineering

Task No.	Sub Task No.	Task	Assigned to	Start Date	Deadline	Story Points (Weeks)
1		Define project scope and objectives	All	8 th August	24 th September	8
2		Requirements Elicitation Techniques	All	20 th August	4 th September	4
3		Work Breakdown Structure	Shenal	22 nd October	11 th November	3
4		Define Functional Requirements	Uhass	11 th November	18 th November	1
5		Define Non-Functional Requirements	Pasindu	11 th November	18 th November	1
6		Form Use Case Descriptions	Anuk	11 th November	18 th November	1
7		Design Diagrams	All			
	7.1	Use Case Diagrams	Yasith	18 th November	25 th November	1
	7.2	Sequence Diagram	Shenal	18 th November	25 th November	1
	7.3	UML Activity Diagram	Yasith	18 th November	25 th November	1
	7.4	UML State Diagram	Patterson	18 th November	25 th November	1
	7.5	UML Class Diagram	Uhass	18 th November	25 th November	1
	7.6	Strategic Dependency Model	Shenal	18 th November	25 th November	1
	7.7	Strategic Rationale Model	Anuk	18 th November	25 th November	1
8		Product Backlog	Uhass	25 th November	2 nd December	1
9		Burn up Charts	Pasindu	25 th November	2 nd December	1
10		1 st Sprint Meeting	All	2 nd December	2 nd December	1
11		Software Requirements Specification				
	11.1	Introduction	Anuk	2 nd December	16 th December	1
	11.2	Overall Description	Anuk	2 nd December	16 th December	1
	11.2.1	Product Perspective				
	11.2.2	Product Features				
	11.2.3	User Classes and Characteristics				
	11.2.4	Operative Environment				

		11.2.5	Design and Implementation Constraints				
		11.2.6	User Documentation				
		11.2.7	Assumptions and Dependencies				
		11.3	System Features				
		11.3.1	Feature 1	Uhass	2 nd December	16 th December	1
		11.3.2	Feature 2				
		11.3.3	Feature 3				
		11.4	External Interface Requirements				
		11.4.1	User Interfaces	Shenal	2 nd December	16 th December	1
		11.4.2	Hardware Interfaces				
		11.4.3	Software Interfaces				
		11.4.4	Communication Interfaces				
		11.5	Other Non-Functional Requirements	Patterson	2 nd December	16 th December	1
		11.6	Other Requirements	Patterson	2 nd December	16 th December	1
		12	Wireframes	Kavidu	2 nd December	16 th December	1

Start Date: 8th August 2023

Team Members:

1. Anuk Salgado (Team Leader)
2. Shenal Perera
3. Uhass Jayaweera
4. Pasindu Perera
5. Yasith Dharmasena

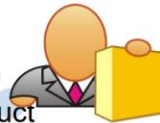
- Each sprint will consist of 1 week.
- There will be 2 scrums instead of the daily scrum in each sprint (week)
 - Tuesday (4pm/9pm)
 - Friday (4pm/9pm)

Product Backlog

- Uthass who will be in charge of preparing the product backlog will be considered as the product owner.

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

Product owner



- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

Sprint Backlog

- At the beginning of each sprint the product owner will specify the sprint backlog. Items for the sprint backlog will be extracted from the product backlog.
 - This can be discussed with the team at the start of each sprint under sprint planning
 - What happens in sprint backlog is where you take a user story from the product backlog and convert into a set of tasks to be performed which paves the way of fulfilling user requirements. It will take the format shown below
 - High-level design is considered

As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

- At this stage documentation will be happening concurrently with the technical phases of the project based on the product backlog
 - Alternatively, another work breakdown structure could be created to understand and learn the required theories as per the suggestion of the team leader Anuk

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

Scrum

- Pasindu will be considered the scrum master who will be in charge of each scrum.

The daily scrum

- Parameters
 - ✓ Daily
 - ✓ 15-minutes
 - ✓ Stand-up
- Not for problem solving
 - ✓ Whole world is invited
 - ✓ Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings



1 What did you do yesterday?

2 What will you do today?

3 Is anything in your way?

The ScrumMaster



- Represents management to the project
 - Responsible for enacting Scrum values and practices
 - Removes impediments
 - Ensure that the team is fully functional and productive
 - Enable close cooperation across all roles and functions
 - Shield the team from external interferences
- As mentioned earlier these scrum meetings will be a part of the two weekly meetings.

Sprint Review

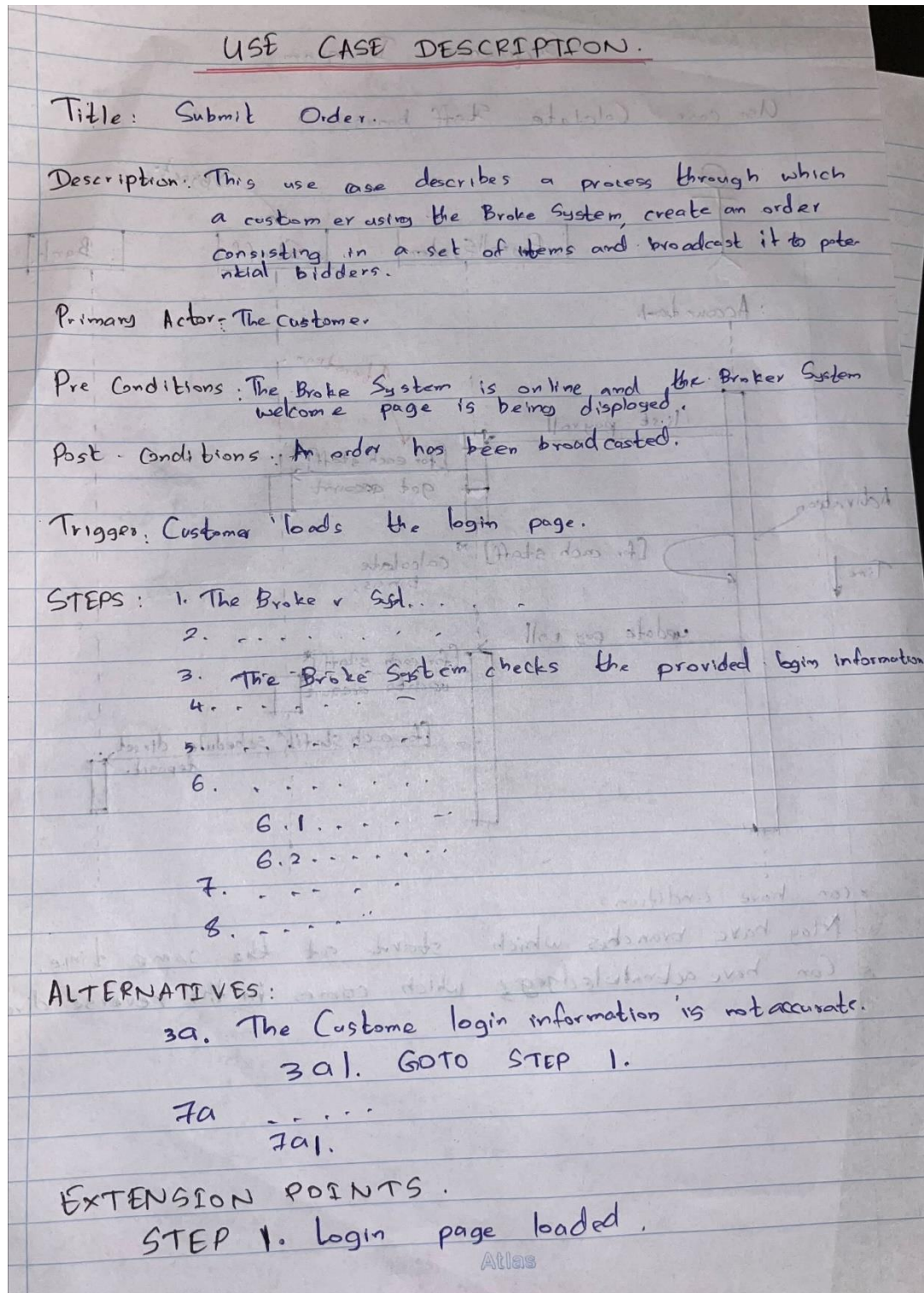
- A sprint review will take place at the end of each sprint (once a week – Tuesday or Friday)
- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
 - ✓ 2-hour prep time rule
 - ✓ No slides



Sprint Retrospective

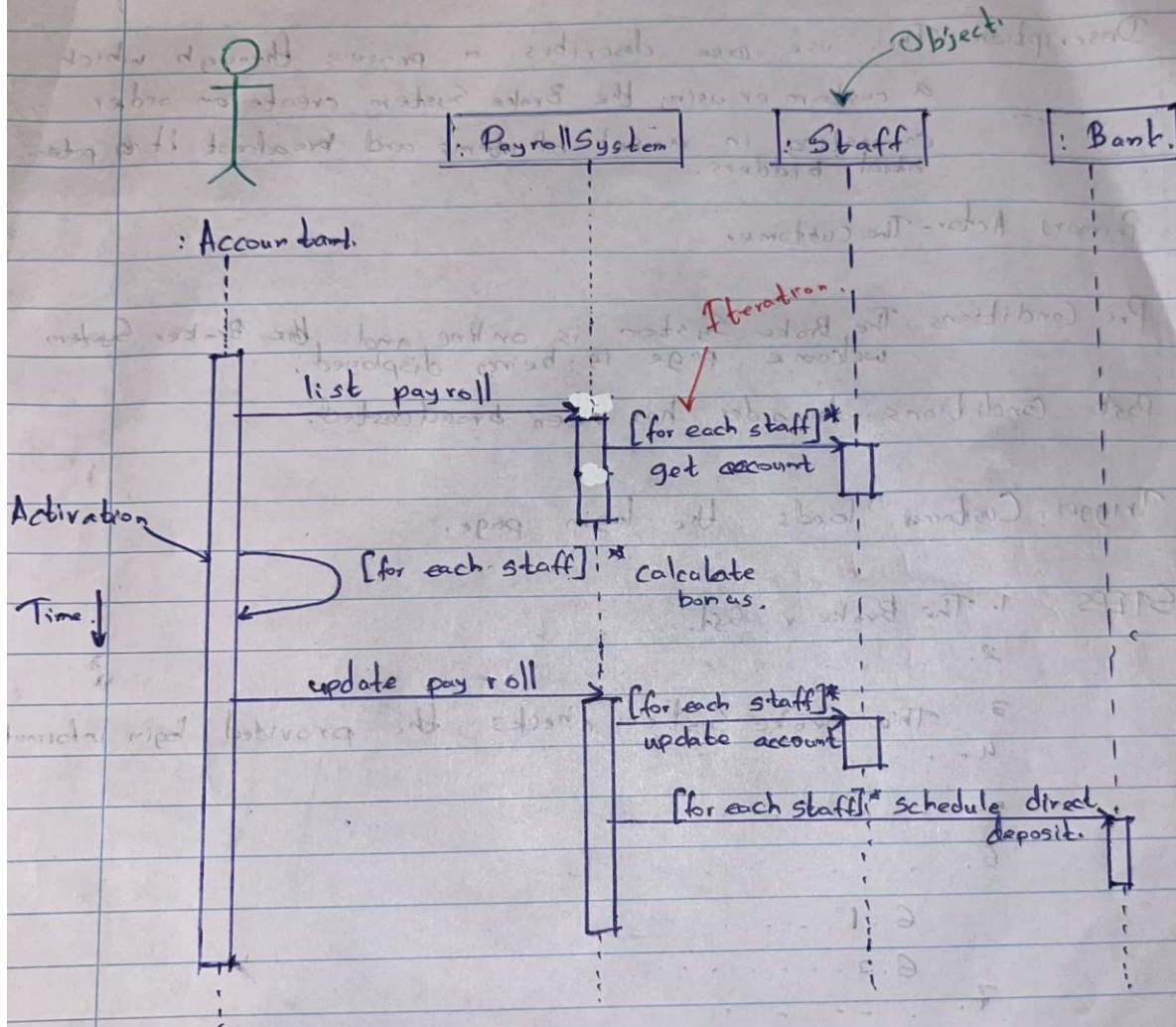
- A sprint retrospective will be held once in every 2 weeks as it is a bit tough to determine the functionality of a feature within a week.
- Periodically take a look at what is and is not working
- Typically 15–30 minutes
 - Basically, the team will get together and decide on the start, stop and continuing tasks.

Sample structures of a few diagrams for your reference



Sequence Diagram

Use case: Calculate Staff bonus.

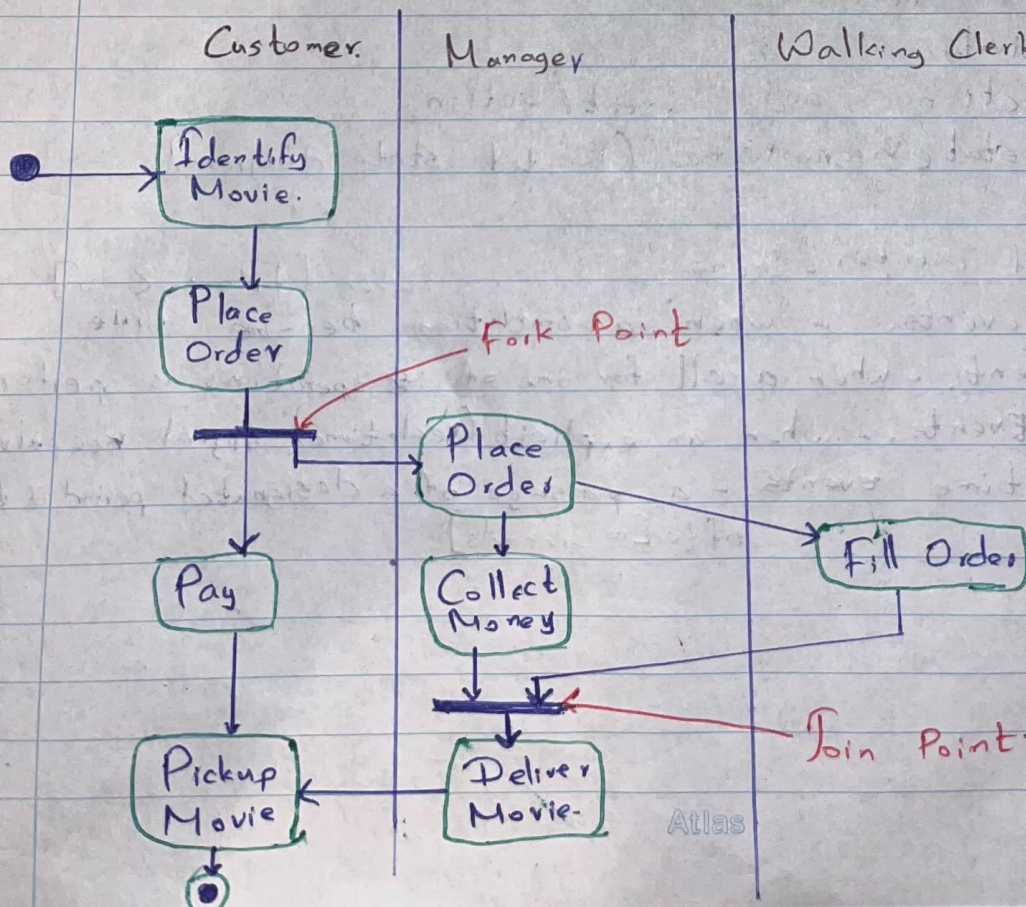
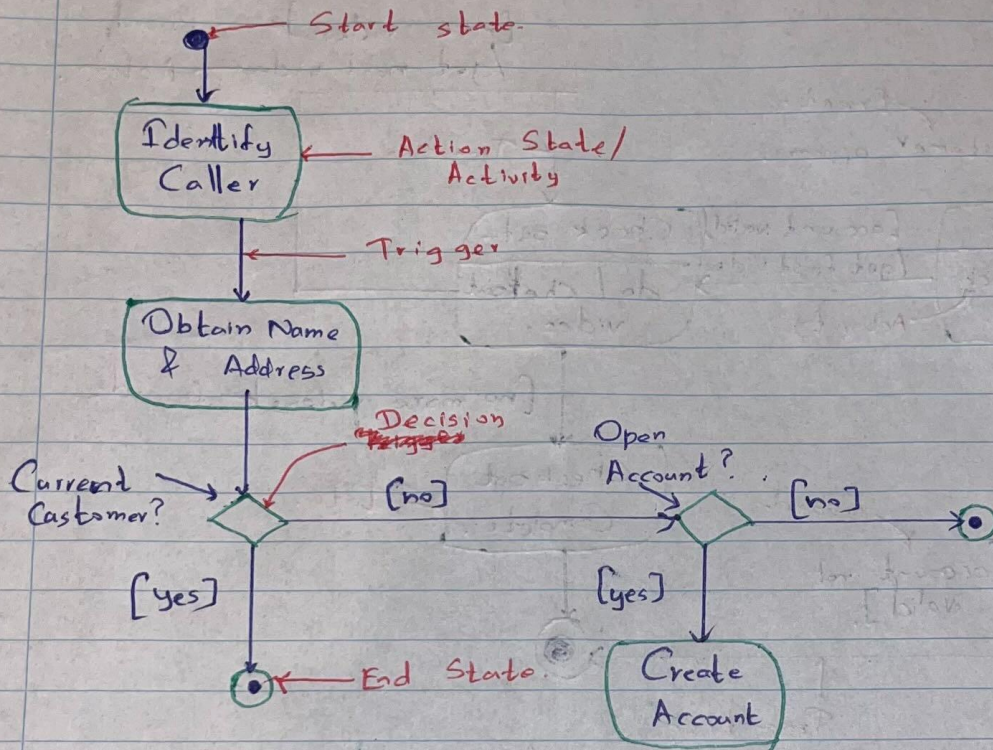


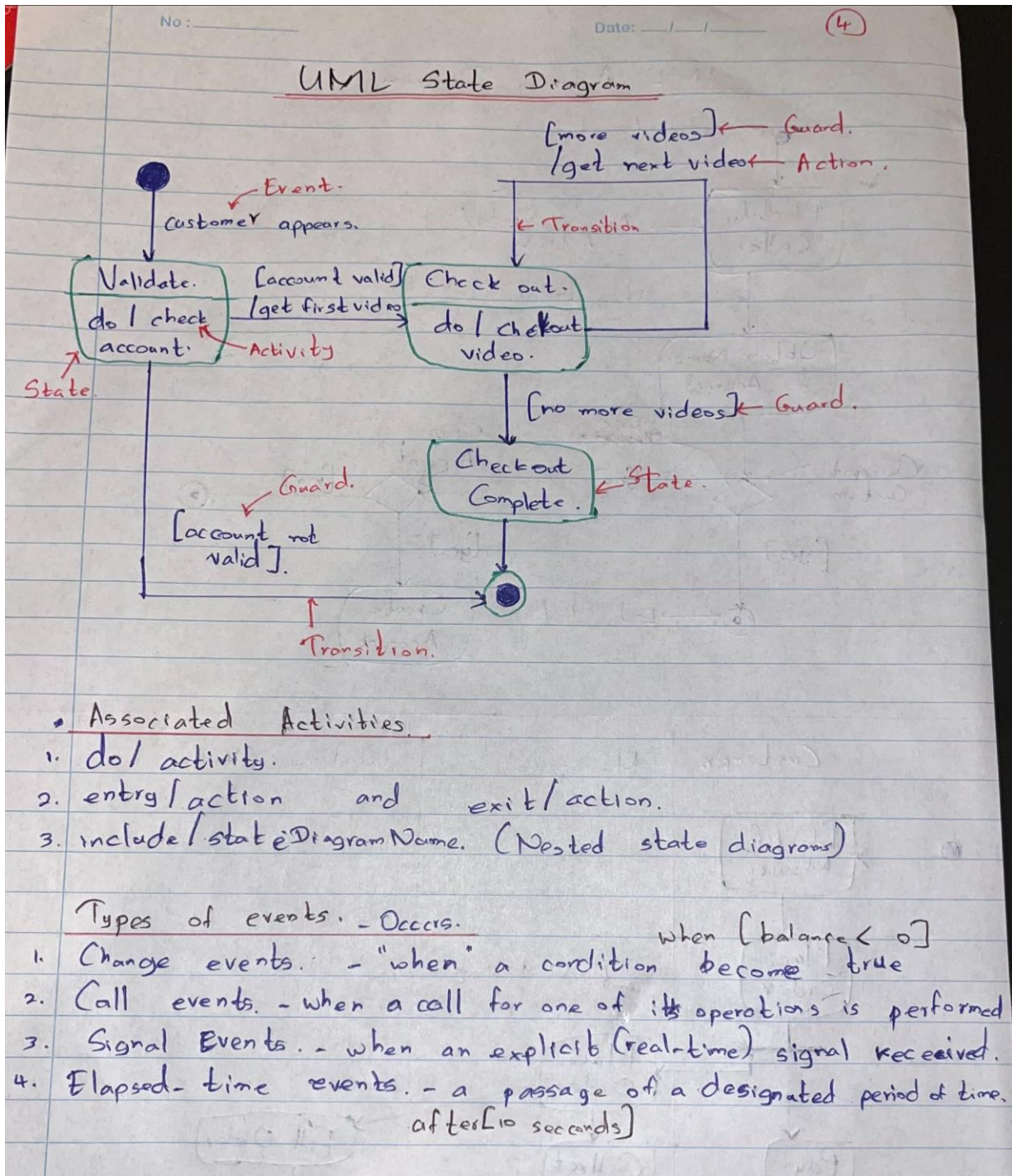
* can have conditions

* May have branches which start at the same time.

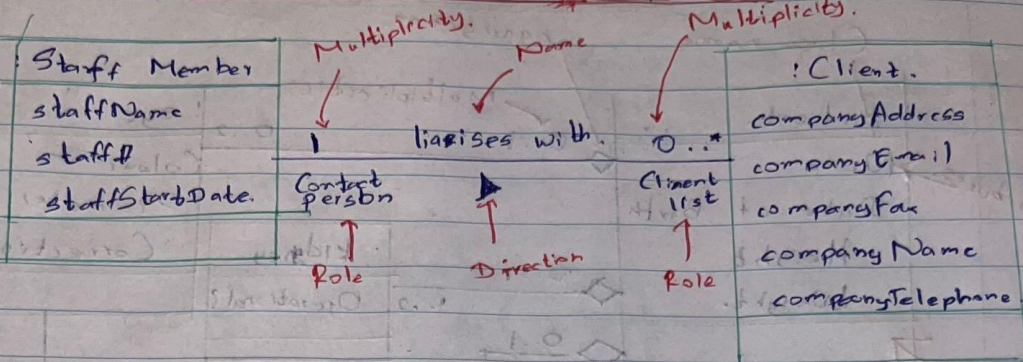
* Can have acknowledgements which comes in the reverse direction

UML Activity Diagram

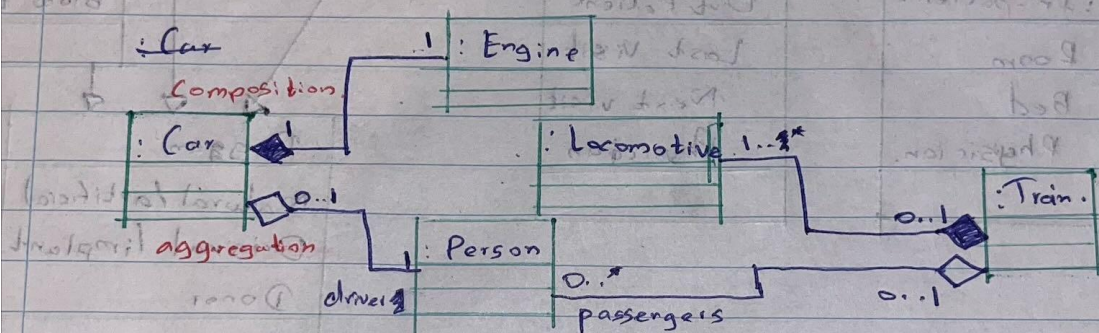




Class Associations - Multiplicity



Aggregation and Composition



Generalization

