Eduvos
Your Education. Your Future.

Individual Assessment Coversheet

To be attached to the front of the assessment.

Campus:	Midrand
Faculty:	Information Technology
Module Code:	IT00A3-11
Group:	Group 2
Lecturer's Name:	Miss Sellwane Nthutang
Student Full Name:	Lethabo Kgasago
Student Number:	MD.2022.X1B8L2

Indicate	Yes	No
Plagiarism report attached	x	

Declaration:

I declare that this assessment is my own original work except for source material explicitly acknowledged. I also declare that this assessment or any other of my original work related to it has not been previously, or is not being simultaneously, submitted for this or any other course. I am aware of the AI policy and acknowledge that I have not used any AI technology to generate or manipulate data, other than as permitted by the assessment instructions. I also declare that I am aware of the Institution's policy and regulations on honesty in academic work as set out in the Conditions of Enrolment, and of the disciplinary guidelines applicable to breaches of such policy and regulations.

Signature	Date
Lethabo kgasago	2024/03/28

Lecturer's Comments:

Marks Awarded:	%

Signature	Date

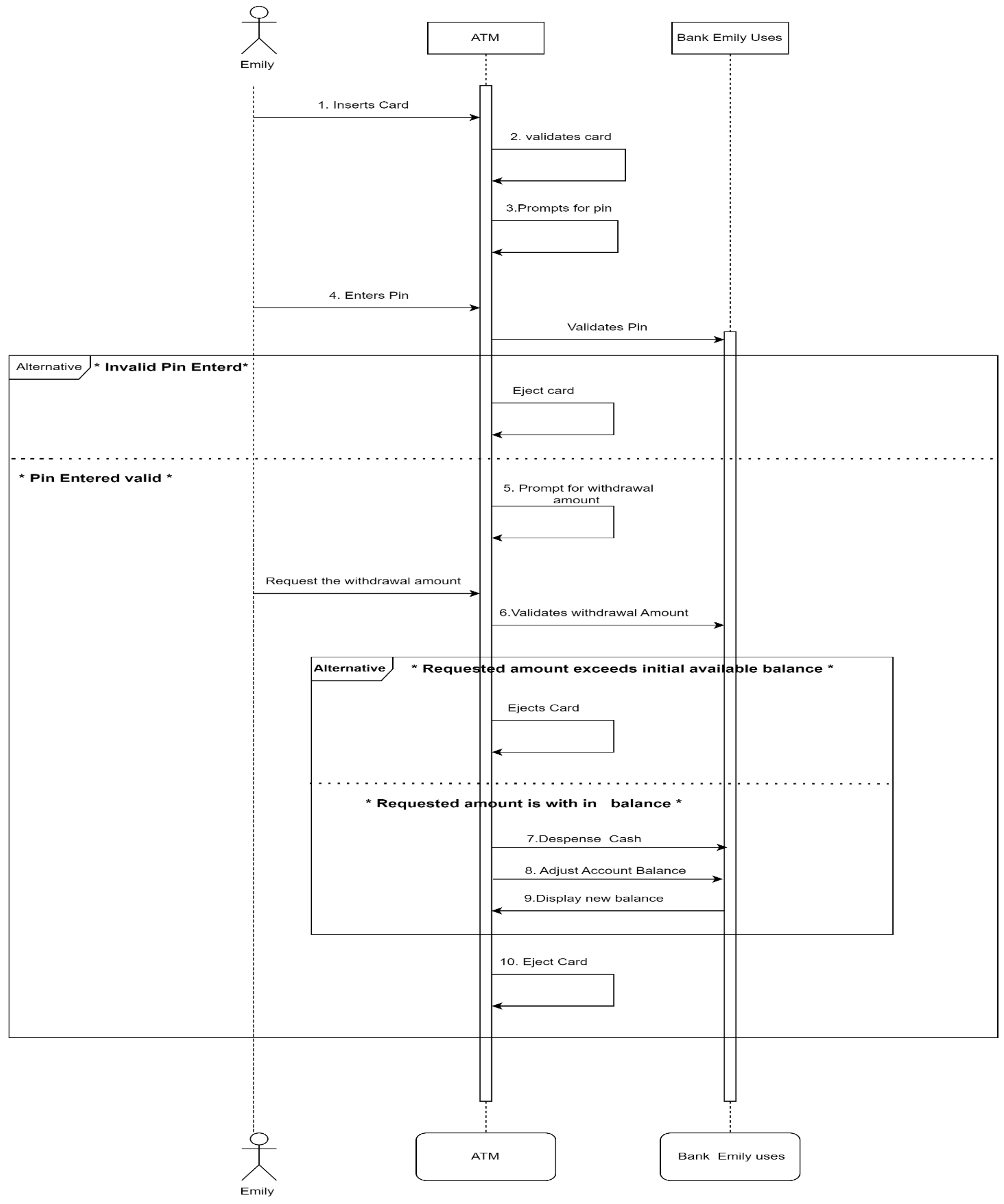
ASSIGNMENT ITOOA3-11

-11 - ASSIGNMENT- Midrand -MD.2022.X1B8L2
ITOO3

Contents

Question 1

Sequence diagram for Emily's withdrawal transaction.



A sequence diagram is a form of interaction diagram that illustrates how tangible objects operate in terms of each other the diagram is a unique UML diagram that illustrates actors, in certain mannered order with an objective to prove functionality of stipulated scenario. The sequence diagram is highly useful in the analysis and design phase of software development. The modeled diagram shows by visualization how the banking system dynamically behaves with transferring data in terms of messages (Dabdawb, 2024).

Sequence diagrams serve a grater purpose in software engineering with regards to visualizing the flow of messages, active events as well as actions happening between valid objects in a system involved (Marwah & Dabdawb, 2024).

Sequence diagrams operate based on characteristics such as:

Chronical order -All messages transferred between objects flow in chronical order to form interactions between objects, a stack of messages if referred to as a sequence of messages shared between objects over a period (Ziema & Abdul, 2024).

Clarity and Readability -The sequence diagram aims to promote clear readability as well as clear visualization of banking systems behavior, the sequence of message flows help stakeholders such as developers and designers to perform analysis at a lower level and analyze the systems interaction and control flow (Bowen , et al., 2024).

Object Centric - objects are centered meaning they are the main objects performing the tasks and making requests in the above scenario Emily as an acting object wants to perform a transection hence, she is interacting with the centric object which is the Atm machine which processes transections and interacts with the bank as well as Emily (Bian, et al., 2024).

Emilys withdrawal transection scenario key components were identified namely:

Objects

Objects are referred to as entities in the scenario and are represented as vertical lines in the requested sequence diagram, the vertical lines in sequence diagram represent objects lifelines.

Each identified object interacts with one another by sending and receiving messages.

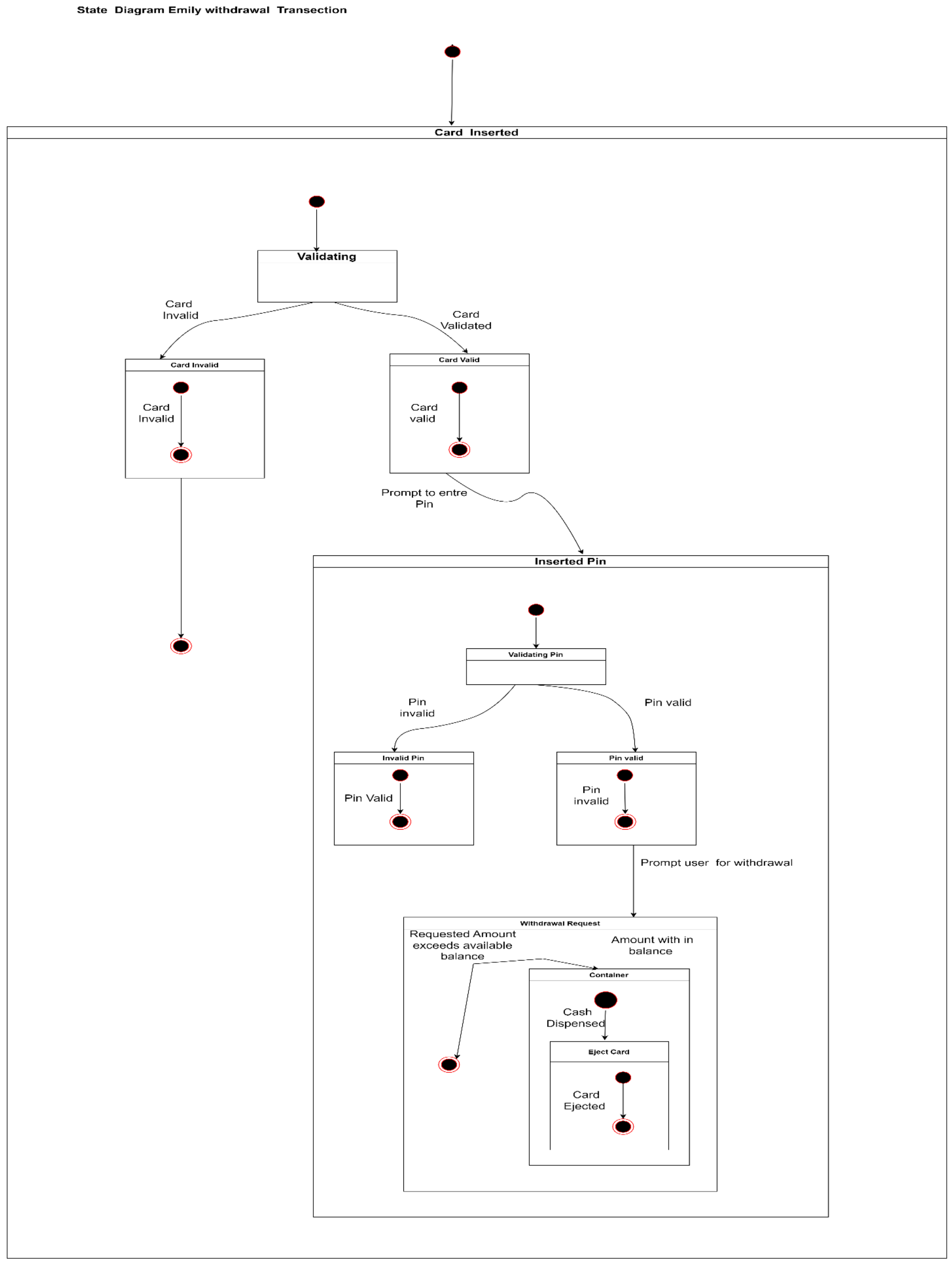
Key objects

- “Emily”-interacting with the system.
- “ATM”-Processing the input transaction.
- “Bank Emilly uses”-validating and updating transactions.

Events of Emily’s withdrawal transection

1. Emily inserts a Bank card in to designated ATM.
2. The ATM validates the inserted ATM card.
3. If Emily’s card is valid, a prompt for entering ATM pin will be shown to Emily.
4. The actor’s object Emily enters Pin.
5. Automatic Tiller Machine transfers entered pin to bank for validation.
6. The Bank approves entered pin.
7. Once the pin is approved, prompt for amount for withdrawal will be displayed.
8. Object actor Emily enters requested amount.
9. ATM processes and sends the withdrawal request to the bank, withdrawal amount requested is authorized and validated to funds available.
10. Funding Bank verifies requested amount for withdrawal is with in margin of current account balance.
11. Requested amount is less or equal to available funds ATM dispenses available notes and forms corrections to Emilys Account balance.
12. ATM displays updated state of bank balance.
13. Client’s card is Ejected.

1.2 State Diagram

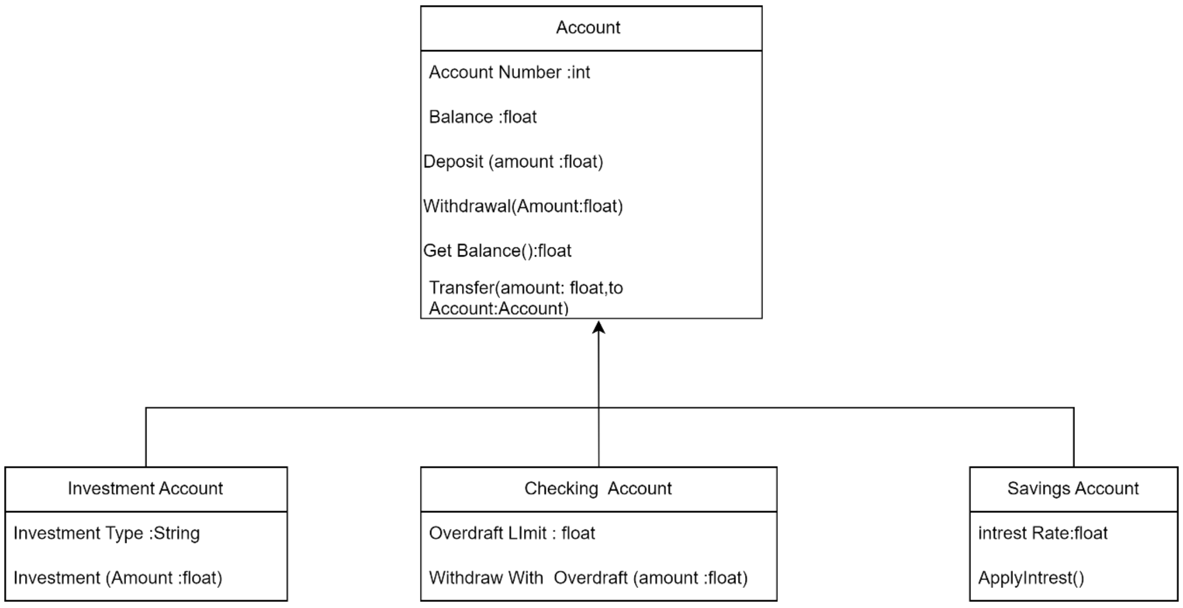


A state machine diagram is used to illustrate how a system, or a portion of a system is operating at specific point of time. State diagrams are behavioral diagrams that make use of state transitions to depict the behavior of the system (Vasiliu, et al., 2024).

The function of state machine diagram is to illustrate different class as well as state changes, hence not commands or processes that are responsible for change (Lincke, 2024).

State diagrams are just like visual aids that help to facilitate the observation and comprehension of a system's behaviour under various operational conditions (Górski, 2024).

1.3 Concept of inheritance Diagram



The concept of inheritance takes a superclass at higher level and divides it into sub classes, sub classes inherit attributes from the parental super class, the parental class identified as super class is used as a blueprint templet which can both extend and feather modify the functionalities of super class (Bian, et al., 2024).

The attributes and methods are inherited by sub classes as well as the parental behavior (AlRababah, 2024).

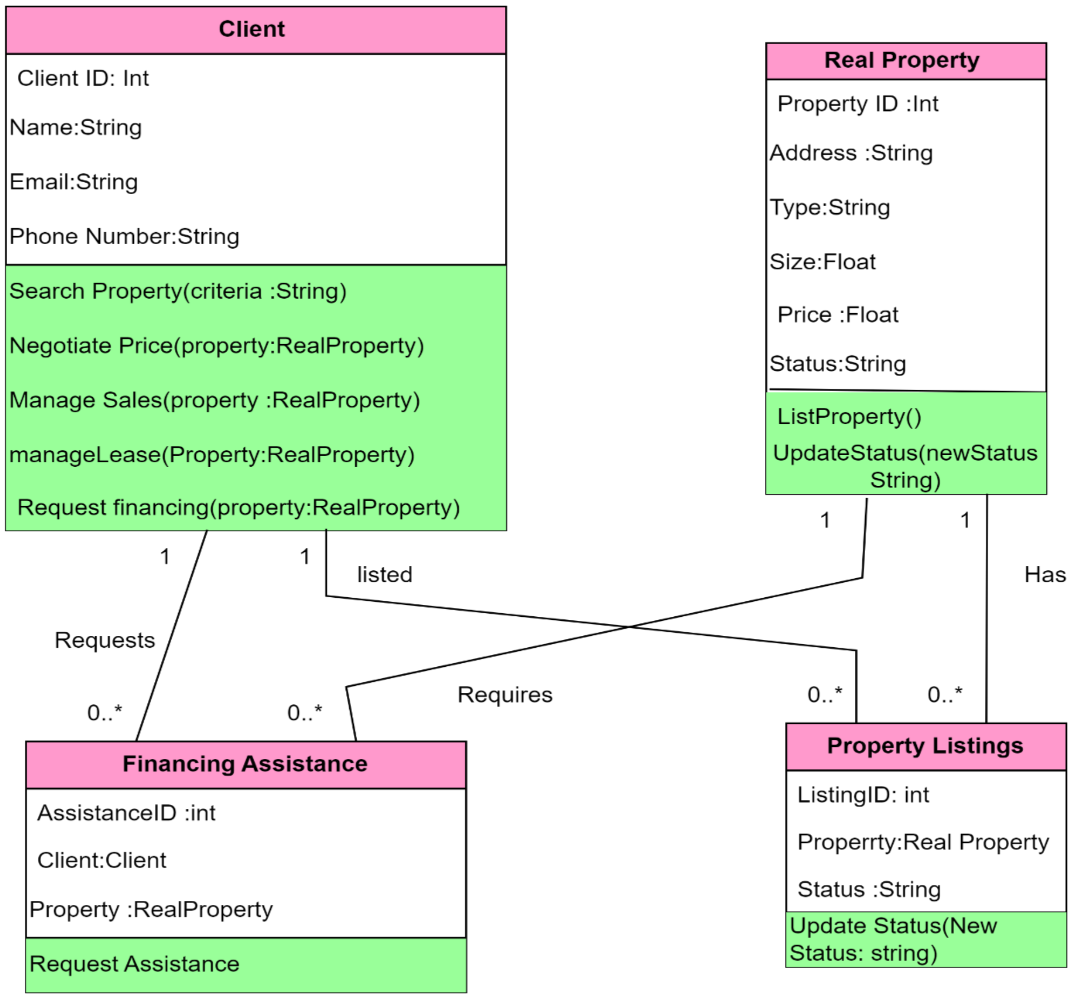
The benefits of concepts of inheritance are code reusability, by smaller sub classes (Ali, Hamza, & Rashid, 2023).

Extensibility And Maintainability -the smaller changes been made are reflected in sub classes involved, this eliminates code duplications hence subclasses extend functional functionalities of super classes (Boltz, et al., 2024).

Account is the supper class with 3 sub classes which are namely investment account, checking account and savings account, in simpler terms a mere client could open an account and use it for a variety of use cases and or use to withdraw funds could be of 3 types and each of the 3 accounts could be linked to a single user to withdraw funds from.

Question 2

2.1 class diagram



One client can have many listed property's **client** "1"—0..* **Property listing .**

One client can have a request for multiple financial assistance **client "1"- - "0..*" financial Assistance.**

One real property can require multiple financial assistance **Real Property "1"- - "0..*" Financial assistance**

One real property can have multiple property listings **Real Property "1"- - "0..*" Property listings**

Classes and attributes

Client class has attributes listed as, ClientID, Name, Email, phone Number.

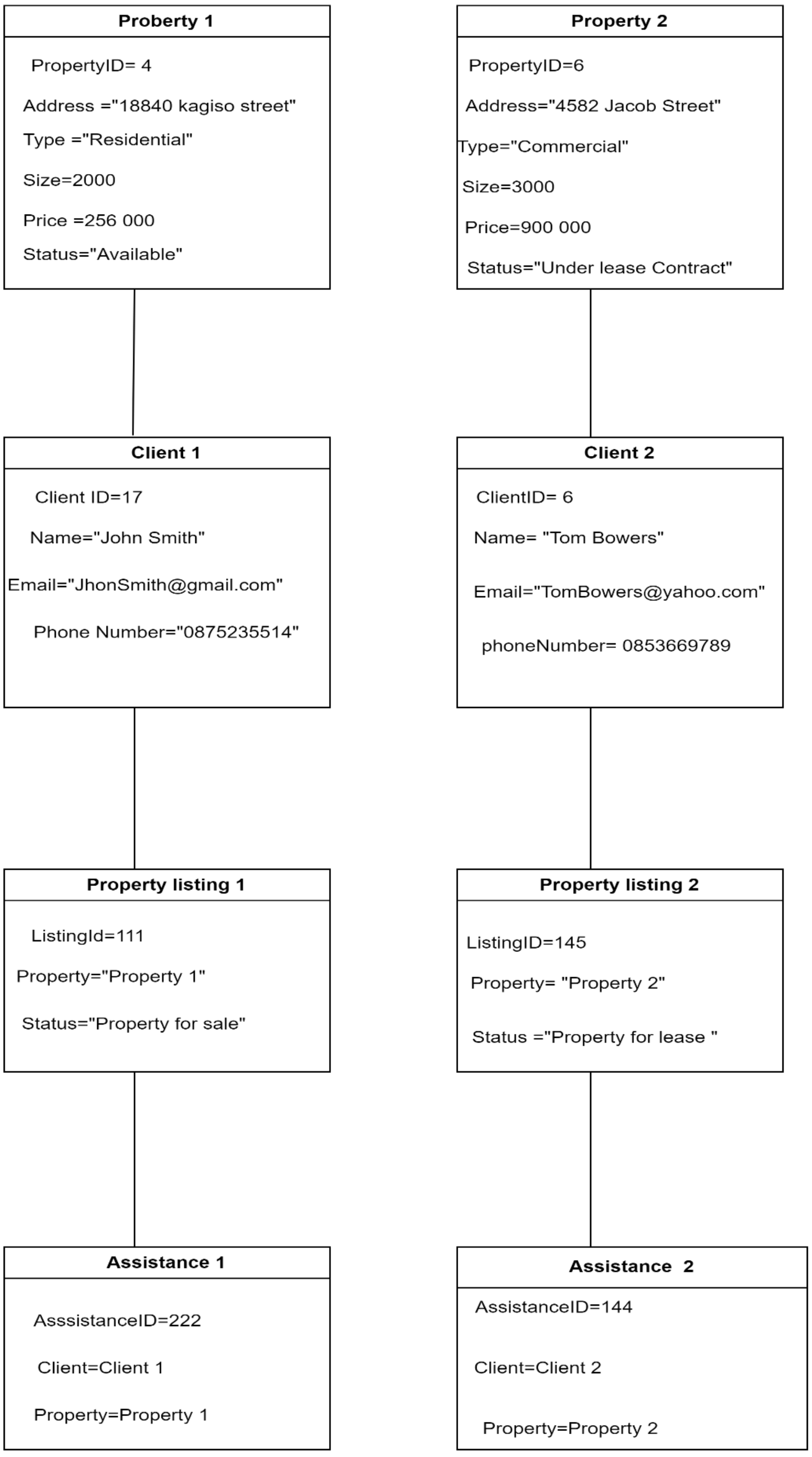
Real property has attributes listed as Property, Address, Type, Size, Price, Status.

Finance Assistance has attributes listed as Assistance, Client, Property.

Property has listed attributes Property.

Property listings has the following attributes, ListingID, Property, Status.

2.2Object Diagram for real Property Estates



Object diagrams represent, Specific instances as well as their relationships, they offer proper insight on how particular objects in the scenario interact with a particular system at a particular moment of time.

Question 3

3.1use case diagram for Eduvos online system

A use case diagram is a type of UML diagram that shows all functional requirements of designed system. evident from the user's perspective, use case diagrams shows interactions of system with external actor's identified as users. Use case Diagram shows a high-level view of implemented systems functionalities and how users interact with the implemented system (Jayadi, et al., 2024).

In the scenario stipulated the are few elements identified namely

Actors

The Eduvos online system has external entities as actors which are.

- Academic support staff
- Student
- Financial Officer

Use Case

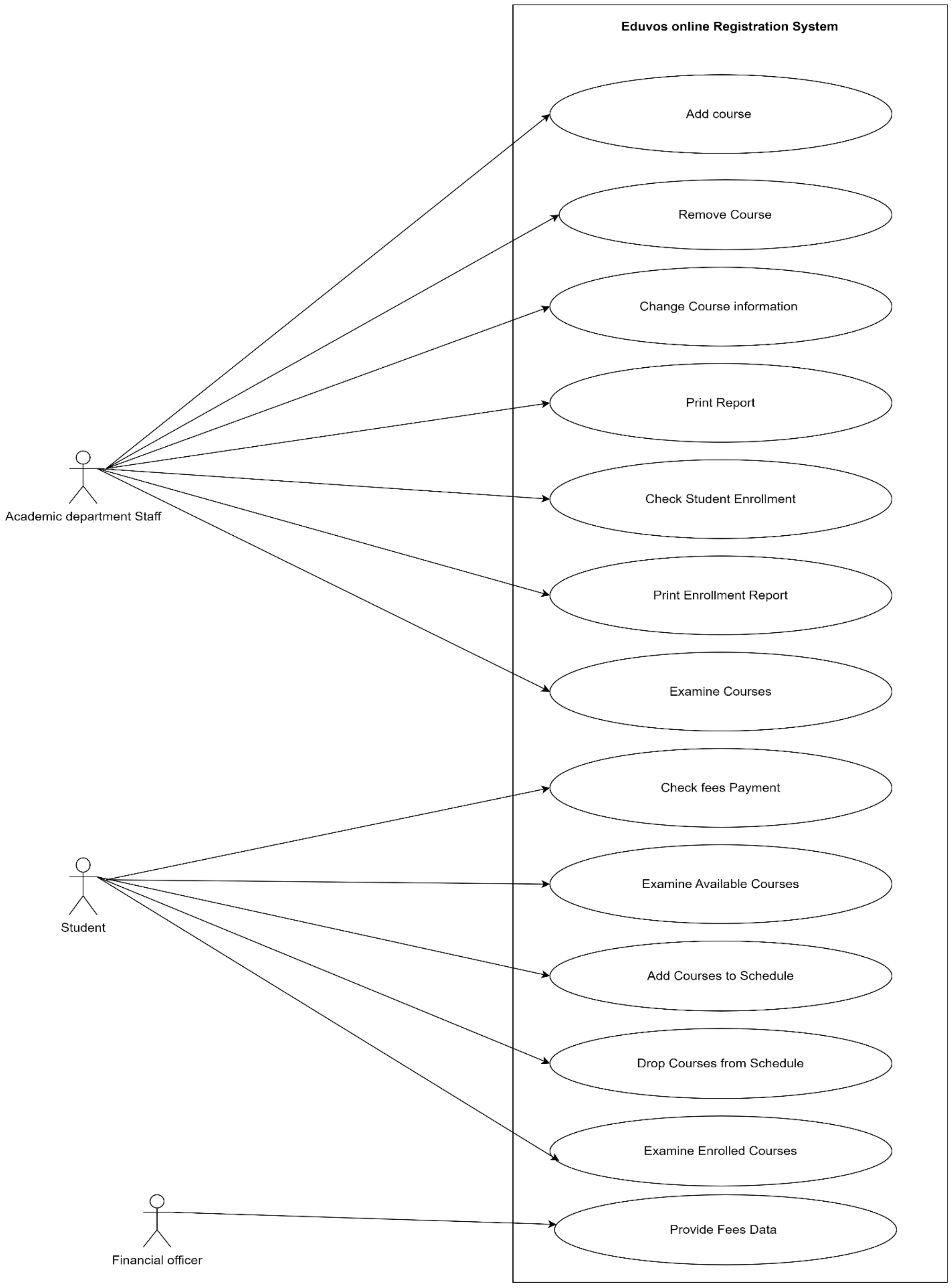
Every system has functionalities which users can use as use cases (Bower, et al., 2024).

Actor /User	**Use Case**
Academic department staff	Add course
	Remove Course
	Change course Information
	Print report
	Check student Enrolment
	Print Enrolment Report
	Examine Courses

Actor/User	**Use Case**
Student	Check fees payment
	Examine available courses
	Add course to schedule
	Drop courses from schedule
	Examine enrolment schedule

Actor/User	**Use Case**
Financial Officer	Provide Fees Data

3.1 Eduvos Online registration system Use case Diagram



3.2 Activity Diagram Eduvos on-line registration

Activity diagrams are a type of UML diagram which represent the workflow of a system and illustrates how activities are made with the system. Workflows illustrate how activities are made with the current working system (Bower, et al., 2024).

Student

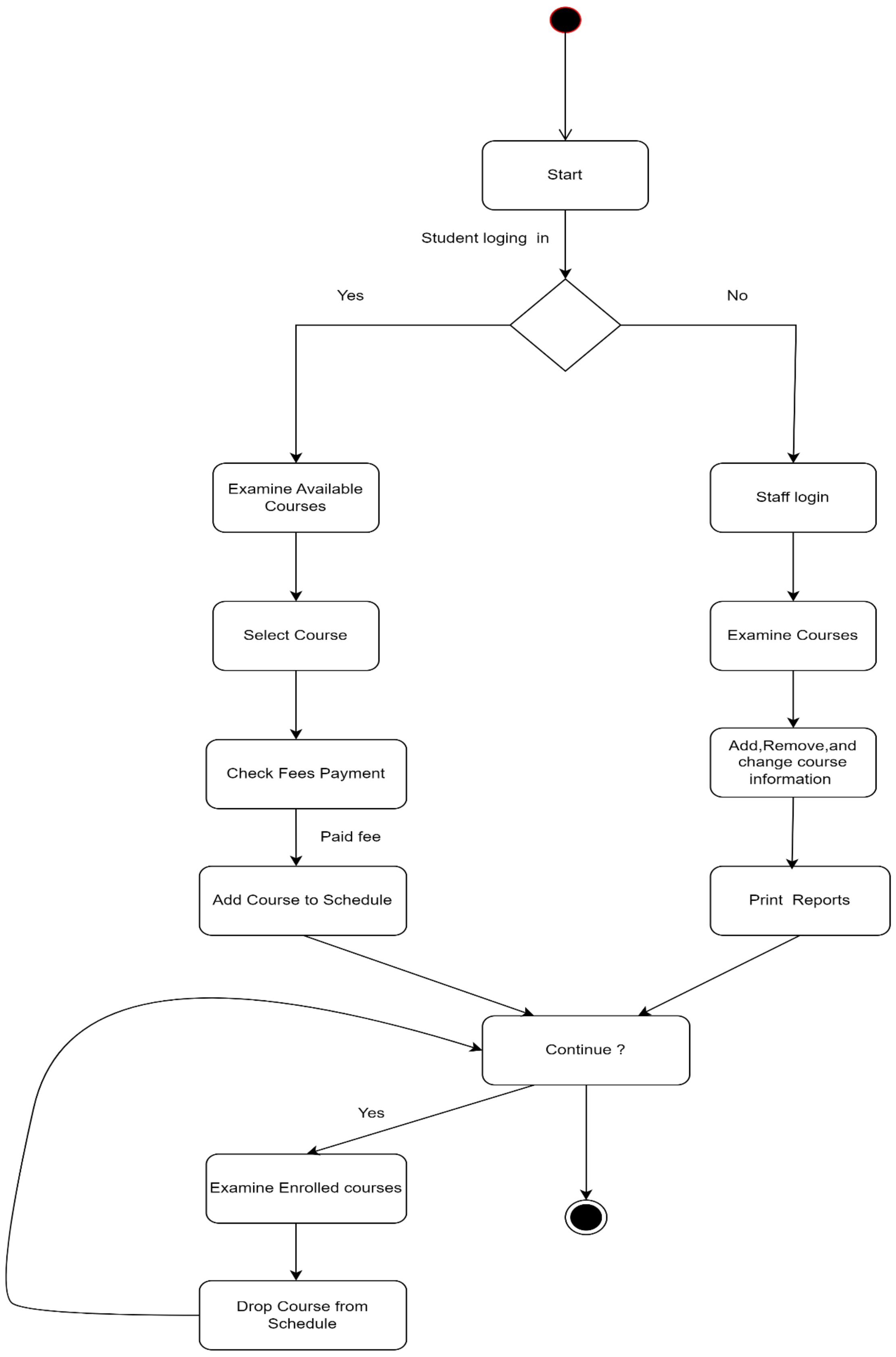
- The process starts with the start node.
- If a candidate is a student, student will examine available courses.
- Choose a course and view fees and payment required.
- The student feather checks payment of fees to make sure they are paid if paid courses are added to their schedule.
- Courses enrolled for are examined and dropped if course is not desired and registration process for student continues till the end node.

Staff member

- If an alternative person if not a student logs in, such as “Staff member”.
- Stuff member will log in, examine student courses feather on add and changes to course information as well as print reports if staff member is happy, they could continue with the process, till registration ends.

3.**2 Activity diagram of Eduvos online registration without swim lane Partitions**

3.2 Activity Diagram Eduvos' online registration with out swimlane Partitions



Question 4

4.1

Classes are blueprints of object instances which have common attribute properties and behavior methods (Planas & Cabot, 2020).

Active classes identified in the scenario.

a) Customer
b) Insurance Policy
c) Vehicle

CRC cards Class Responsibility Collaboration (CRC)

CRC diagrams are used as brainstorming diagrams to analyze software requirements in a faster approach based on agile development (Tuglular & Leblebici, 2020).

a)

Class Name	Responsibilities	Collaborators involved
Customer	• Issue policy numbers. • Issue information concerning other drivers	• Insurance Clerk • Insurance Policy

b)

Class Name	Responsibilities	Collaborators involved
Insurance Policy	• Show basic insurance policy information. • Review policy status (if up to date or not) • Check total coverage in opposition to other ranges. • Update policy with new vehicle information. • Calculate New premiums.	• Customer • Vehicle • Insurance Clerk

c)

Class Name	Responsibilities	Collaborators involved
Vehicle	• Issue vehicle Specifications (Year, Vin number, model, make “manufacture”) • Authenticity of vehicle information validated.	• Insurance Policy • Customer

4.2

Use case description =Adding a new car to existing policy

Actor: customer

Check list preconditions.

a) Customer is in possession of an existing car insurance policy.

b) Consumer has telephonically called the insurance call center.

Basic interaction flow

- Client issues policy, number to insurance clerk.
- The clerk captures policy numbers into insurance system.
- Insurance system displays policy information.
- Clerk checks if insurance policy is up to date.
- Specifications of new vehicle are provided by consumer, specification describe vehicle by (VIN Number, Year, Make, Model).
- Authenticity of vehicle information is validated while been entered by the insurance clerk.
- Customer concludes by choosing the desired premium fee and cover type.
- Insurance Clerk captures and records Premium fee and cover type.
- Insurance system validates total coverage against other available ranges.
- Client provides other alternative drivers information.
- Alternative new use cases are invoked if new driver is added.
- Policy is updated with new information and new calculated premiums.
- Updates are created and copy is printed and sent to policy owner.

Post check conditions.

Clients’ policy is updated to include newly listed vehicles.

Client receives an updated policy, as well as premiums.

4.3

An association relationship indicates that instances of one particular, class are connected to or related, relation instances of another class in the context of system development is crucial for defining the structure and its functions.
Object-oriented systems show how classes are related to one another.
The are higher possibilities to find one-to-one, one-to-many, or many-to-many relationship present.
Associations are typical in a class diagram and are represented by simple lines connecting the classes.

Works Cited

Vasiliu¹, L., Roman, D. and Prodan, R., 2024. Check for updates Extreme and Sustainable Graph Processing for Green Finance Investment and Trading.

In *AI, Data, and Digitalization: First International Symposium, SAIDD 2023, Sogndal, Norway, May 9–10, 2023, Revised Selected Papers* (p. 120). Springer Nature.

Lincke, S., 2024. Planning for Secure Software Requirements and Design with UML. In *Information Security Planning: A Practical Approach* (pp. 417-445). Cham: Springer International Publishing.

Górski, T., 2024. Smart contract design pattern for processing logically coherent transaction types. *Applied Sciences*, *14*(6), p.2224.

Dabdawb, M., 2024. Unified Modeling Language Quantitative Measures Based on a Behavioural Model. *Journal of Education & Science*, *33*(1).

Mushtaq, Z. and Wahid, A., 2024. Revised approach for the prediction of functional size of mobile application. *Applied Computing and Informatics*, *20*(1/2), pp.181-193.

Li, B., Wu, W., Tang, Z., Shi, L., Yang, J., Li, J., Yao, S., Qian, C., Hui, B., Zhang, Q. and Yu, Z., 2024. DevBench: A Comprehensive Benchmark for Software Development. *arXiv preprint arXiv:2403.08604*.

Bian, Y., Shen, Y., Zheng, F., Ning, F. and Wang, G., 2024, March. Research on software modeling of aerial electro-optical system based on UML. In *Advanced Fiber Laser Conference (AFL2023)* (Vol. 13104, pp. 634-642). SPIE.

Ali, H.M., Hamza, M.Y. and Rashid, T.A., Exploring Polymorphism: Flexibility and Code Reusability in Object-Oriented Programming.

Boltz, N., Hahner, S., Gerking, C. and Heinrich, R., 2024. An Extensible Framework for Architecture-Based Data Flow Analysis for Information Security. *arXiv preprint arXiv:2403.09402*.

Jayadi, P., Dewi, R.S. and Sussolaikah, K., 2024. Activity-based function point complexity of use case diagrams for software effort estimation. *Journal of Soft Computing Exploration*, *5*(1), pp.1-8.

Bower, M., Howe, C., McCredie, N., Robinson, A. and Grover, D., 2014. Augmented Reality in education–cases, places and potentials. *Educational Media International*, *51*(1), pp.1-15.

Bastos, R.M. and Ruiz, D.D.A., 2002, January. Extending UML activity diagram for workflow modelling in production systems. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 3786-3795). IEEE.

Planas, E. and Cabot, J., 2020. How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus. *Computer Standards & Interfaces*, *67*, p.103363.

Tuglular, T. and Leblebici, O., 2020, October. Automatic Code Generation with Document Responsibility Collaboration Modelling Method. In *2020 Turkish National Software Engineering Symposium (UYMS)* (pp. 1-6). IEEE.