

BINUS University

Academic Career: <i>Undergraduate / Master / Doctoral *)</i>	Class Program: <i>International/Regular/Smart Program/Global Class*)</i>	
<input checked="" type="checkbox"/> Mid Exam <input type="checkbox"/> Final Exam <input type="checkbox"/> Short Term Exam <input type="checkbox"/> Others Exam : _____	Term : Odd/ Even / Short *)	
<input checked="" type="checkbox"/> Kemanggisan <input checked="" type="checkbox"/> Alam Sutera <input type="checkbox"/> Bekasi <input type="checkbox"/> Senayan <input type="checkbox"/> Bandung <input type="checkbox"/> Malang	Academic Year : 2021 / 2022	
Faculty / Dept. : School of Computer Science	Deadline	Day / Date : Thursday / Nov 25 th , 2021 Time : 17:00
Code – Course : COMP6115001 – Object Oriented Analysis and Design	Class : All Classes	
Lecturer : Team	Exam Type : Online	
*) <i>Strikethrough the unnecessary items</i>		
<i>The penalty for CHEATING is DROP OUT!!!</i>		

Learning Outcomes:

LO 1 : Conceive the basics concepts of object-oriented analysis and design.

LO 2 : Use the knowledge to develop documentation for object-oriented software analysis and design using Unified Modelling Language.

LO 3 : Analyze any problem in any software application and find out the alternative solutions using object-oriented analysis and design approach.

LO 4 : Manage the software process and build software development teams based on objectoriented analysis and design approach.

I. Essay (20%)

1. **[LO 1 & LO 2, 5 points]** In object-oriented analysis and design paradigm, we can use the ***Unified Process*** methodology. Describe comprehensively the ***phases*** and the ***workflow*** of this method. Give some examples of its implementation.

There is 4 phases in Unified Process :

- **Inception**

Inception phase is the first phase where we prepare the basis of the proposed system (project). We preparing the it's business case, project scope, key requirement, risk, and many thing in this phase. Basically we prepare everything here before we start making the project. *For example: Discussing with the team what are the system goal and also its requirements.*

- **Elaboration**

Elaboration phase is the phase where the team is expected to understand or catch the majority of system requirements. So the team can start making it's design and architecture, analyze it's risk, and create the plan for the project for the next phase. *For example: Create a plan for the system with the team, like schedule, estimated cost, risk, and milestones need to be achieved.*

- **Construction**

Construction phase is the longest phase where the design of the system is finalized and refined (from elaboration phase). In this phase the construction phase is also divided into many iterations. In the last iteration, a fully finished system will be formed and will be used in the last phase, the transition phase. *For example: Making the system with the team based from what we do in Inception and Elaboration phase.*

- **Transition**

Transition phase is the phase where a full finished system will be delivered to the end user. *For example: A brand new system that already been made by the team will be published to the end user.*

And from that phases there is 5 workflow from Unified Process :

- **Requirement**

The primary activity from this workflow is making a use case model from the proposed system that makes it more easy to understand the system requirement. The model and diagram is also useful to reach agreement with the stakeholders.

- **Analysis**

The primary activity from this workflow is making a analysis model from the proposed system to change or refine the functional requirement from the use case model that have been made.

- **Design**

The primary activity from this workflow is making a design model from the proposed system that gives the real realization from the use case model and analysis model.

- **Implementation**

The primary activity from this workflow is making a implementation model from the proposed system which describes how the elements of the design model are packaged into software components.

- **Test**

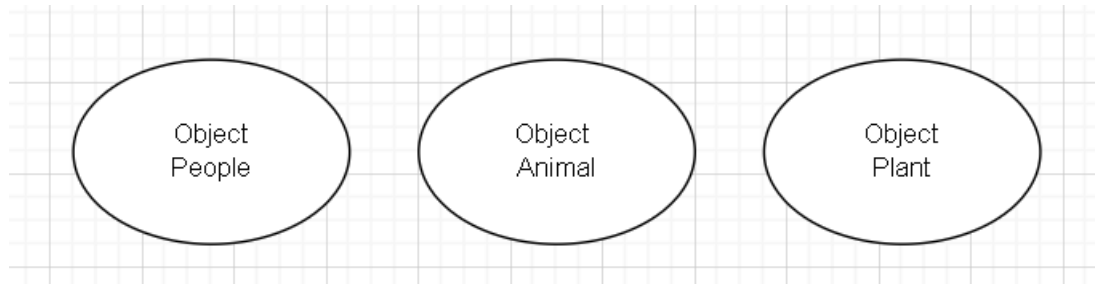
The primary activity from this workflow is making a test model from the proposed system where the team will do some system test from the implementation workflow. The test model also describe on how the team will perform those test.

2. **[LO 1 & LO 2, 5 points]** Object-oriented systems concepts show some important *features* including *object identity*, *classes*, *inheritance*, and *polymorphism*. Give detail explanations and examples or illustrations for those features.

- **Object identity**

An object is an instantiation of a class. An object is person, place, event, or anything that we want to capture its information. In conclusion object identity is what distinguishes one object from another object.

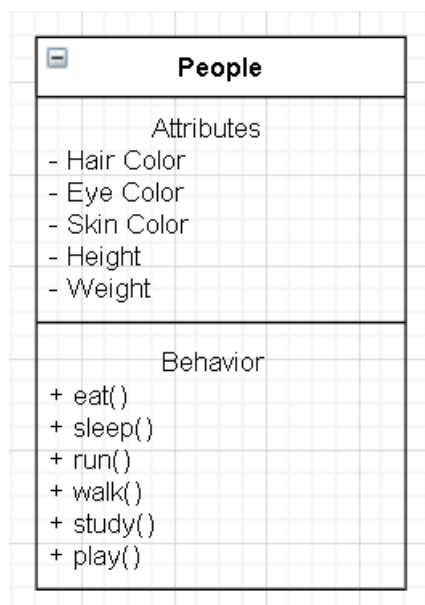
Example :



- **Classes**

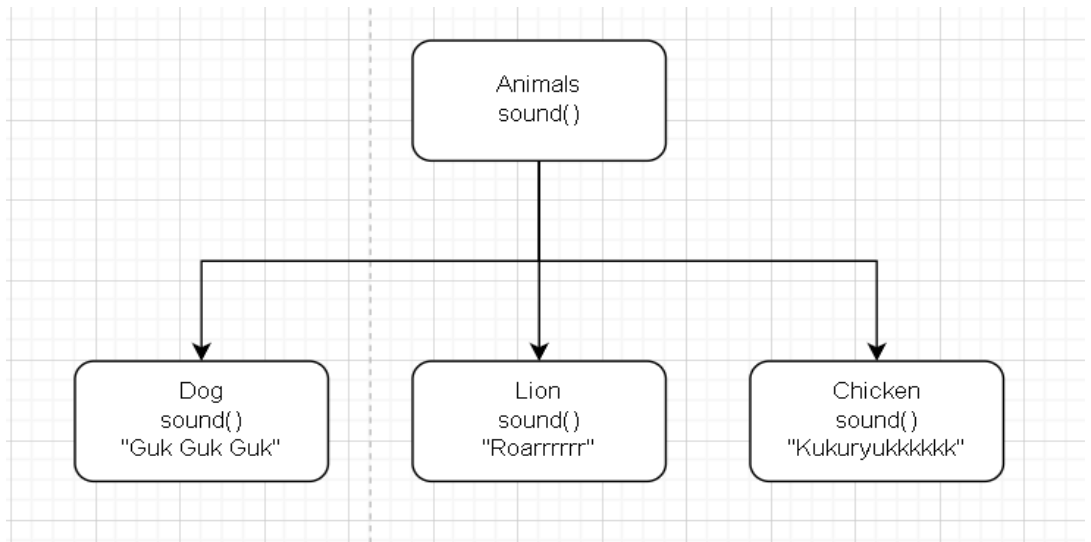
Classes are the template used to defined and create specific object.

Example :



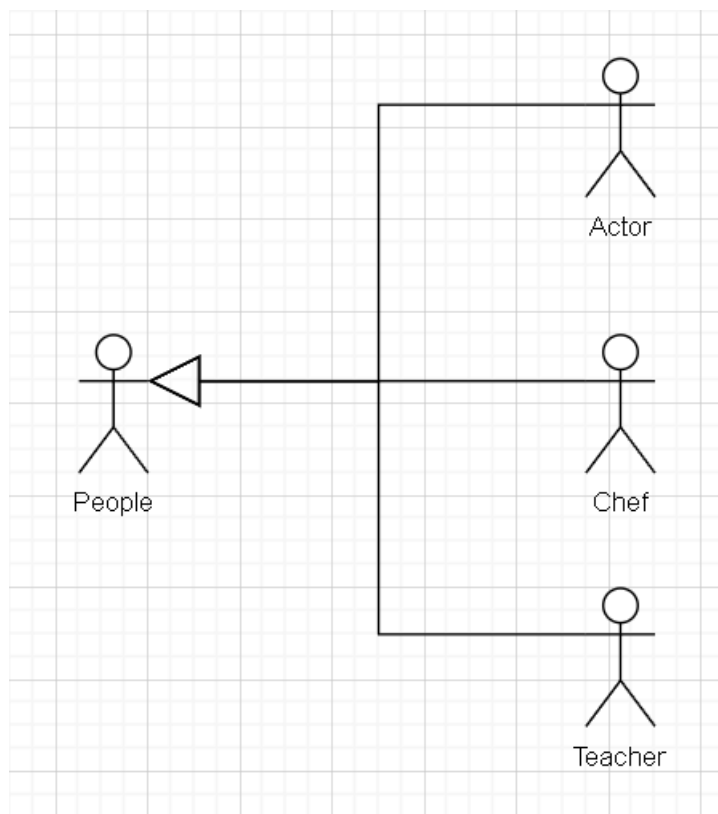
- **Polymorphism**

Polymorphism is the ability to take object on many different type. It applies the objects but also the operations. A polymorphic object is the one who true type hides within a super or parent class.



- **Inheritance**

Inheritance is a feature that can allows to create a children class (sub-classes) from a parent class by inheriting the attributes and the method that parent class have.



3. [LO 1 & LO 2, 5 points] In developing system requirements, the initial effort of the development is the process of compiling **analysis domain/boundary**. What does it mean by the **domain analysis** and what are the purposes? (What are the advantages if there was a domain analysis document)?

- Domain analysis is the process analyze system background information. The more information we have, the easier it will be to understand the system that been working on and make it easier to make decisions later in other software development process.

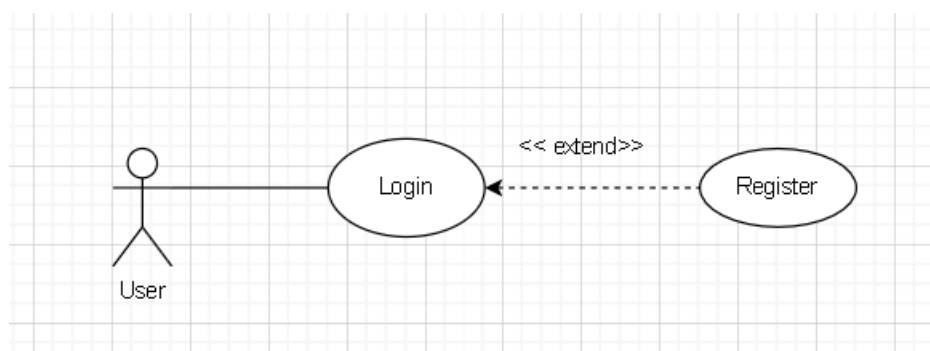
There are some advantages from using Domain Analysis :

1. Faster development
We can make system development more faster because we can communicate with the stakeholders more effektivly with domain analysis.
2. Better System
Of course by digging a lot of background information the system will help us to make better system later.
3. Anticipation of extensions
With the use of domain analysis, it will be easier for us to see trends and make our system more adaptable in the future.

4. [LO 1 & LO 2, 5 points] In UML Use Case diagram, there are **extension stereo type** (<<extend>>) and **inclusion stereo type** (<<include>>). What do these two things mean? Give some examples of their use.

- **Extension stereo type (<<extend>>)**
Extend relationship is a relation that shows a use case extend some use case (other function) if a certain condition occurs or we can say it's optional. For example, if certain conditions are met, the extended use case will work.

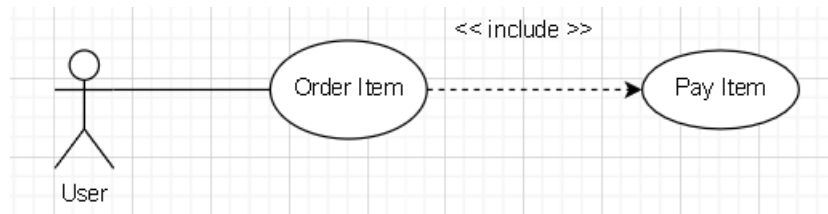
Example :



As we can see use case Login is extended the Register use case. It's because if the user already register the account they don't need to register anymore, instead if the user is not registered yet they need to register first before they do login.

- **Inclusion stereo type (<<include>>)**
Include relationship is a relation that represents a use case is included in the main use case. For example, if one use case runs, the other use case will also run.

Example :



As we can see use case pay item is included to order item use case. That means that when user order an item, the user need to pay the item also so the item can be ordered and send to the user.

II. Cases (80%)

To work on the following case, please pay attention to the last digit of your NIM (Student Number). Choose the system application theme that matches to the last digit of your NIM. Mistakes in choosing a theme will result in all answers being considered wrong. Example:

- If your NIM is: 2301922070, that means the last number of your NIM is 0, then you have to analyse for the **pharmacy application system**.
- If your NIM is: 2301922071, that means the last number of your NIM is 1, then you have to analyse for the **hospital application system**.

And so on.

Select one of the following business areas according to the last digit of your NIM as the scope of the project to be worked on: 0 = **pharmacy application system**,

1 = **hospital application system**,

2 = **human resources application system**,

3 = **tourism application system**,

4 = **product sales application system**,

5 = **service sales application system**,

6 = **property application system**,

7 = **entertainment application system**,

8 = **manufacturer application system**,

9 = **school application system**

My NIM is 2301882306, so I will choose property application system.

Your tasks are:

1. [LO 1, LO 2, LO 3, & LO 4, 20 points] Determine the **functional** and **non-functional requirements** that may exist in the scope of selected applications. Set the analysis domain/boundary according to your assumptions.

Functional Requirement :

- Login
Before enter the application user need to login to their account to access all application features.
- Register

If the user doesn't have account before they need to register their account and the system will add new account to the database. So later user can use that account to access the application features.

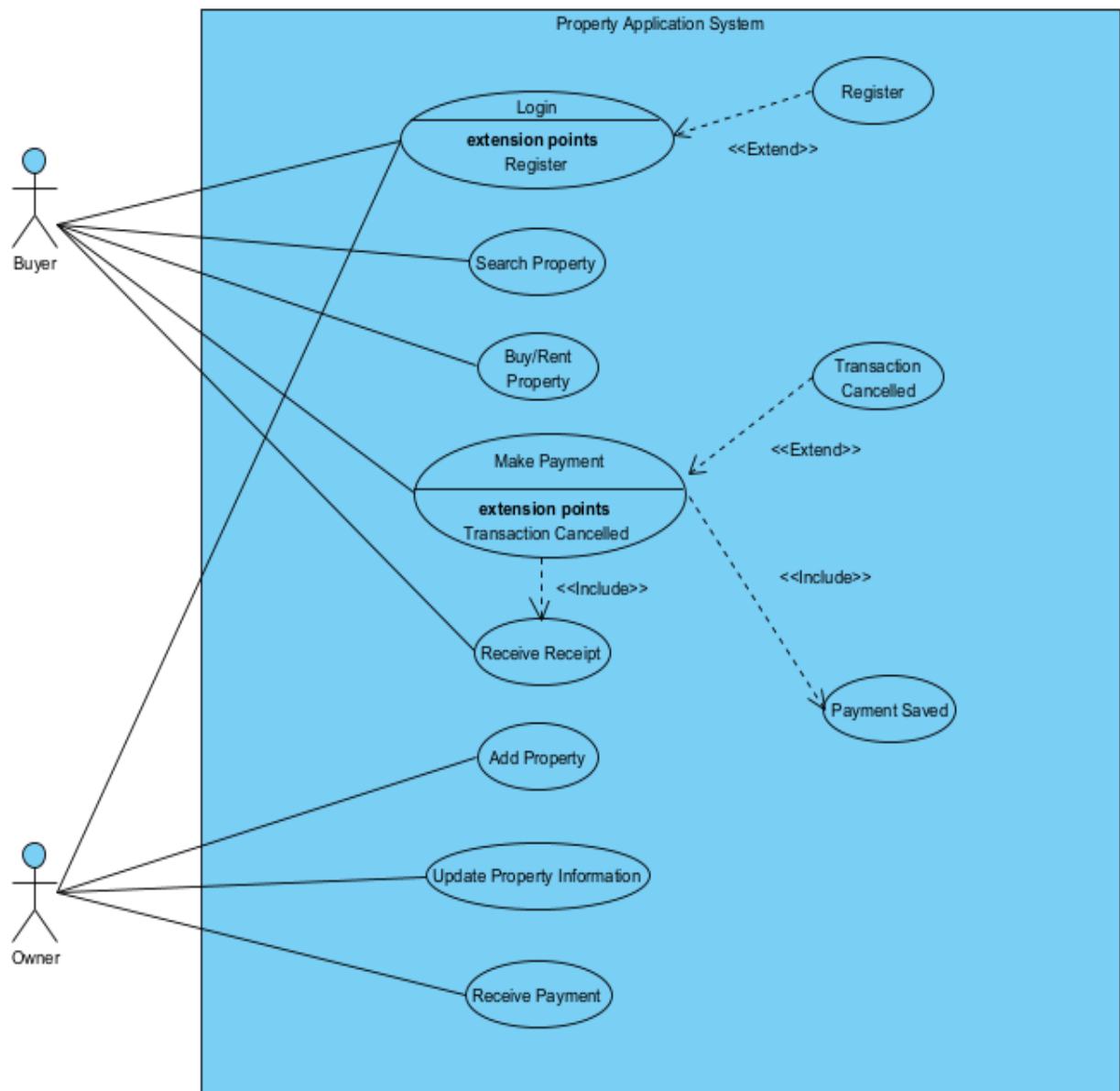
- **Search Property**
Because this is property application system, user can search and view property detail in this application.
- **Buy/Rent Property**
After user find the desired property, they can buy or rent the property.
- **Payment Saved**
The system will receive the payment and will later distribute the money to the property owner if the transaction is successful.
- **Receive Receipt**
After a successful transaction, user will receive their receipt as the proof from buying/renting a property.
- **Add Property**
Not only buying/renting the apartment, the user who have the property can rent or sell their property by add and promote the property through the application.
- **Update Property**
User also can update their property info, so that buyers can see clearly the info and details of the property being marketed.
- **Receive Payment**
After a successful sale user will receive their payment.

Non Functional Requirement

- The size of the application is not more than 150mb.
- Application can be accessed by internet.
- Data is stored securely in the database.
- Easy and convenient application for user access.
- The process of moving between pages does not take more than 5 seconds.

2. [LO 1, LO 2, LO 3, & LO 4, 20 points] Define the use case modelling components at the analysis level such as *actor*, *use case scenario*, and all *use cases* (processes) according to the chosen business theme. Draw the **Use Case Diagram** and create the **use case description** for each *use case*!

Use Case Diagram



Use case description

Use Case : Login

Actor : Buyer, Owner

Goals : To access Property Application System

Overview :

Before access the application feature, user need to validate their data by login to the application.

Typical course of events :

Actor action

System action

2. User input email and password

1. Display login form

3. Validate user data

4. User successfully enter the application

Alternative courses :

Step 2 - 4 : User input wrong email and password, so user cannot enter to the application or user doesn't have account yet they need to register account first.

Use Case : Search Property

Actor : Buyer,

Goals : To search for the property that buyer want.

Overview:

Buyer can search for the property they want before buy/rent it.

Typical course of events :

Actor action

System action

2. Buyer find the property they want

1. Display all property list

3. The buyer choose the desired property

4. View property information

Alternative courses :

Step 2 - 4 : Buyer didn't find their desired property.

Use Case : Buy/Rent Property

Actor : Buyer,

Goals : To buy/rent property.

Overview:

Buyers can buy or rent the property they want.

Typical course of events :

Actor action

System action

1. Display all property list

2. Buyer find the property they want

3. The buyer choose the desired property

4. View property information

5. Buyer interested with the property.

6. Buyer buy/rent the property.

7. Process the transaction

Alternative courses :

Step 2 - 7 : Buyer didn't find their desired property. So buyer didn't buy/rent the property.

Use Case : Make Payment

Actor : Buyer,

Goals : Pay the property transaction invoice.

Overview :

Buyers need to pay their invoice to process the transaction.

Typical course of events :

Actor action

System action

1. Display transaction invoice

2. Buyer pay the invoice

3. Process the payment

4. Transaction Success

5. Payment saved.

7. User receive receipt

6. System create receipt.

Alternative courses :

Step 2 - 7 : Buyer didn't pay the invoice within the specified time. So the transaction will be automatically canceled.

Use Case : Receive Receipt

Actor : Buyer,

Goals : Buyer receive the transaction receipt.

Overview:

Buyers will receive the receipt when they finish the payment.

Typical course of events :

Actor action

System action

1. Buyer complete the payment

2. Prepare the receipt

4. Buyer receive the receipt

3. Display the receipt

Alternative courses :

Step 1 - 4 : Buyer didn't receive receipt if they not complete the payment.

Use Case : Add Property.

Actor : Owner.

Goals : Add property to the application.

Overview:

Property owner will add their property to the application to promote it.

Typical course of events :

Actor action	System action
	1. Display add property form
2. Owner fill all the property data.	3. Validate property data.
4. Owner successfully add property.	5. Newproperty added to list

Alternative courses :

Step 2 - 5 : When owner input wrong property data, newproperty will not be added.

Use Case : Update Property Information.

Actor : Owner.

Goals : Update property information in the application.

Overview:

Property owner can update their property information through the application.

Typical course of events :

Actor action	System action
	1. Display update property form
2. Owner fill all the property data.	3. Validate property data.
4. Owner successfully update property.	5. Property info updated.

Alternative courses :

Step 2 - 5 : When owner input wrong property data or cancel to update the property, property information will not be updated.

Use Case : Receive Payment.

Actor : Owner.

Goals : Owner will receive the payment from the transaction made.

Overview:

Property owner will receive the money soon as the transaction has reach the deal.

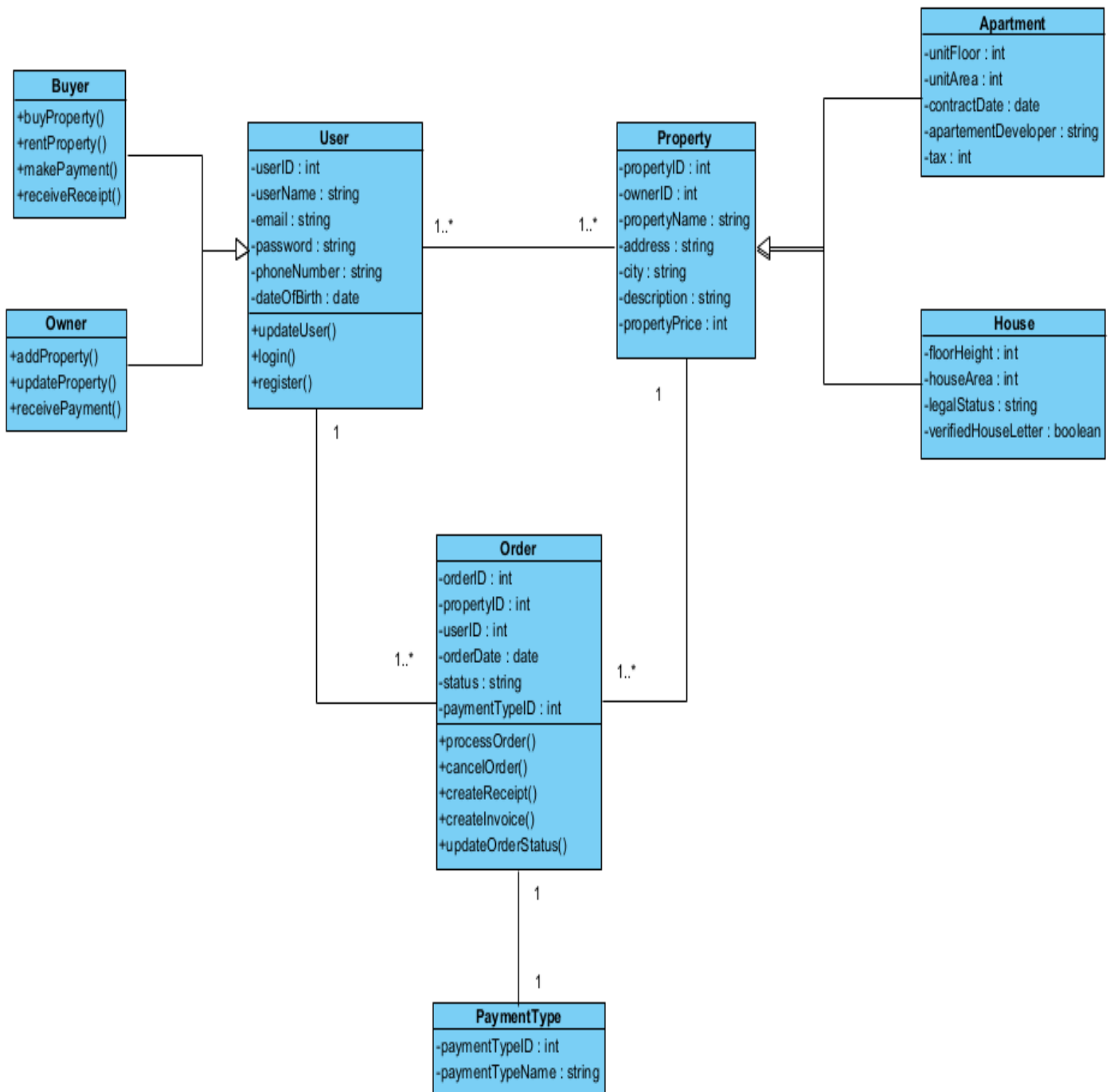
Typical course of events :

Actor action	System action
	1. Transaction completed
3. Owner receive the payment money.	2. Send money to the owner.

Alternative courses :

Step 1 - 3 : If transaction not complete, money will not be send to the owner.

3. [LO 1, LO 2, LO 3, & LO 4, 20 points] Draw the **class diagram** covers all possible objects in the system application you have defined. The class diagram includes all aspects such as **object classes**, **attributes**, **relationships**, dan **methods**. As much as possible, also include the aspects of the hierarchical class structure: **inheritances**.



4. [LO 1, LO 2, LO 3, & LO 4, 20 points] In terms of the project management for the application system you are working on, work also on the **use case point estimation** to calculate how much **effort** were needed to complete your system. Use the worksheets sample in your reference books, set the related numbers according to the description in the questions no 2 and 3 above. Assign a value for each technical complexity factors (13 factors) and choose one of the two PHMs (*PersonHours Multiplier*), 20 or 28. If the standard cost of each team member is IDR 1,000,000 per hour, how much will it **cost** to complete your project.

- **Unadjusted Actor Weighting Table**

Actor Type	Description	Weighting Factor	Number	Result
Simple	External system with well-defined API	1	0	0
Average	External system using a protocol-based interface, e.g., HTTP, TCT/IP, or a database	2	0	0
Complex	Human	3	2	6
			Unadjusted Actor Weight Total (UAW)	6

- **Unadjusted Use Case Weighting Table**

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1–3 transactions	5	0	0
Average	4–7 transactions	10	6	60
Complex	>7 transactions	15	2	30
			Unadjusted Use Case Weight Total (UUCW)	90

Unadjusted Use Case Point (UUCP) = UAW + UUCW = 90 + 6 = 96

- **Technical Complexity Factors**

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T1	Distributed system	2.0	0	0
T2	Response time or throughput performance objectives	1.0	4	4
T3	End-user online efficiency	1.0	4	4
T4	Complex internal processing	1.0	1	1
T5	Reusability of code	1.0	3	3
T6	Ease of installation	0.5	4	2
T7	Ease of use	0.5	4	2
T8	Portability	2.0	2	4
T9	Ease of change	1.0	1	1
T10	Concurrency	1.0	0	0
T11	Special security objectives included	1.0	1	1
T12	Direct access for third parties	1.0	0	0
T13	Special user training required	1.0	0	0

			Technical Factor Value (Tfactor)	22
--	--	--	----------------------------------	----

Technical Complexity Factor (TCF) = $0.6 + (0.01 * \text{Tfactor}) = 0.6 + (0.01 * 22) = 0.82$

- **Environmental Factors**

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
E1	Familiarity with system development process being used	1.5	3	4.5
E2	Application experience	0.5	3	1.5
E3	Object-oriented experience	1.0	3	3
E4	Lead analyst capability	0.5	3	1.5
E5	Motivation	1.0	4	4
E6	Requirements stability	2.0	3	6
E7	Part-time staff	-1.0	0	0
E8	Difficulty of programming language	-1.0	3	-3.0
			Environmental Factor Value (EFactor)	17.5

Environmental Factor (ECF) = $1.4 + (-0.03 * \text{Efactor}) = 1.4 + (-0.03 * 17.5) = 0.875$

Adjusted Use Case Points (UCP) = $\text{UUCP} * \text{TCF} * \text{ECF} = 96 * 0.82 * 0.875 = 68.88$

Effort in person-hours = $\text{UCP} * \text{PHM} = 68.88 * 20 = 1377.6$

Cost needed = $1377.6 * \text{Rp. } 1,000,000.00 = \text{Rp. } 1,377,600,000.00$