Department of Computer Science and Software Engineering

Model Driven Software Engineering
COEN-6312

Deliverable - 4

Submitted by:
TEAM-BLEED BLUE

Supervised by:
Dr. Abdelwahab Hamou-Lhadj

Table of Contents

Table of Figures

State Diagram:

A state diagram describes the behavior of the objects of a *class.* It can also describe the behavior of a system. . A state diagram, also called a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language (UML). We are only interested in the behavior of objects of a class. At any given point in time, the object is in a certain state, it moves to the next state when it received a signal (a method call, an internal or external signal)

States

A state of an object represents a set of values of its attributes at certain point of time. An object remains in a state until it receives events; it then moves to another state if needed. A state is represented by a rounded rectangle containing the name of the state. Each class has up to one state which describes the behavior of all objects of this class.

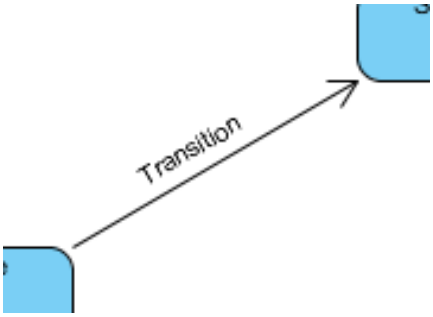
Initial State

An initial state represents a default vertex that is the source for a single transition to the default state of a composite state. There can be at most one initial vertex in a region. The outgoing transition from the initial vertex may have a behavior, but not a trigger or guard.

Final State

A state that represents the enclosing region is completed. If the enclosing region is directly contained in a state machine and all other regions in the state machine also are completed, then it means that the entire state machine is completed.

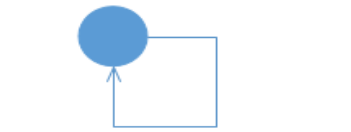
Transition



A transition represents a change of state in response to a signal (or event). It is considered to occur instantaneously. The label on each transition is the signal that causes the change of state.

Reflexive transition

The transition into the same state.



As per our class diagram for ***Online Real Estate System,*** we can have following state diagrams for different classes of the system:

I. State Diagram for class Account

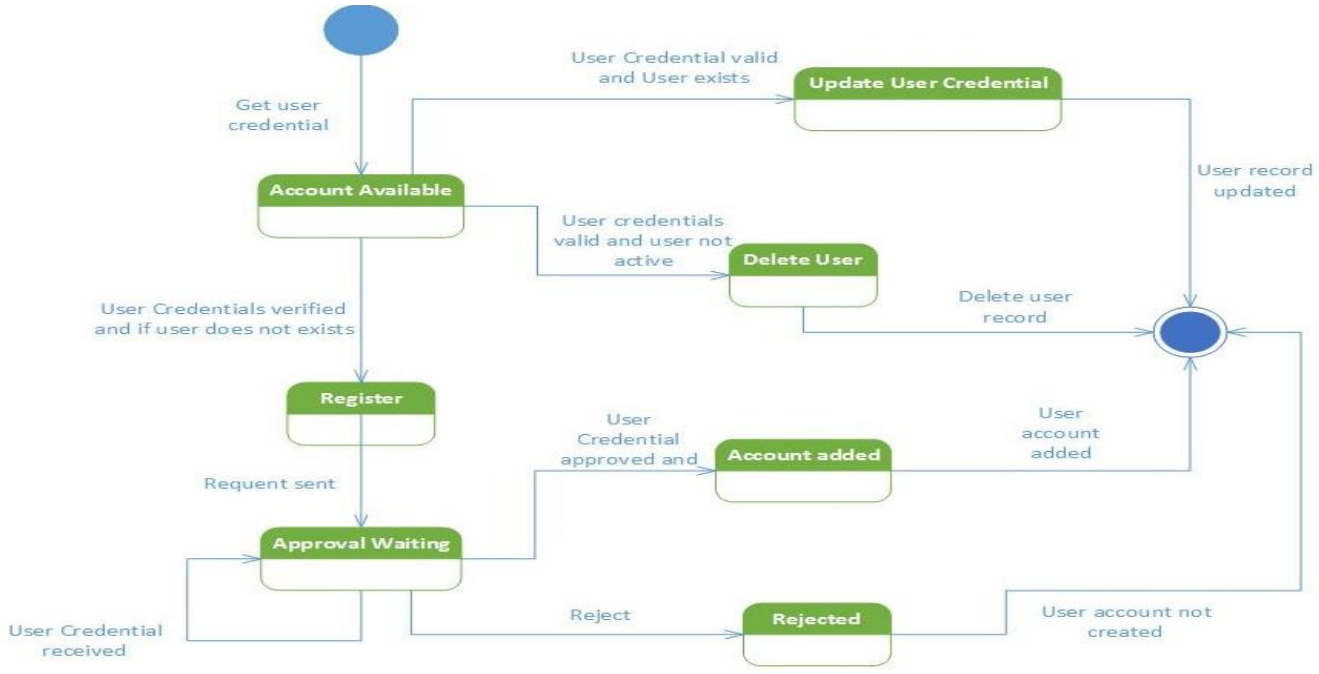


Figure 1: State diagram for "Account"

Description

System will be in *Account Available* state initially. *The Account Available* has three scenarios:

Scenario 1.a: Update User Credential

This state is achieved if the user's credentials are valid and if user exists, the system will allow user to update User Credential and then will reach to final termination.

Scenario 1.b: Delete User

This state is achieved if the user credentials are valid and user is no more active or participating, the system will allow admin to delete the user's record and thereby reach to final termination.

Scenario 1.c: Register

This state is achieved after performing the action of verifying user credential and if user's account does not exists, the system will instantly move to the *Register state*. In the Register state, the user will enter the User profile credentials to register the account in the system and till any further action is taken the system will remain in the *Approval Waiting state*. The *Approval Waiting state* gives rise to two scenarios:

Scenario 2.a: Account Added state

After reaching the *Approval Waiting state*, a condition will be checked to determine whether the user's credential entered are correct. If the condition is true then the system add the account and will thereby leading to *Termination* state.

Scenario 2.b: Rejected state

This state will be achieved if the user's credentials entered are invalid and thereby leading to state of *Termination.*

II. State Diagram for class **Inquiry**

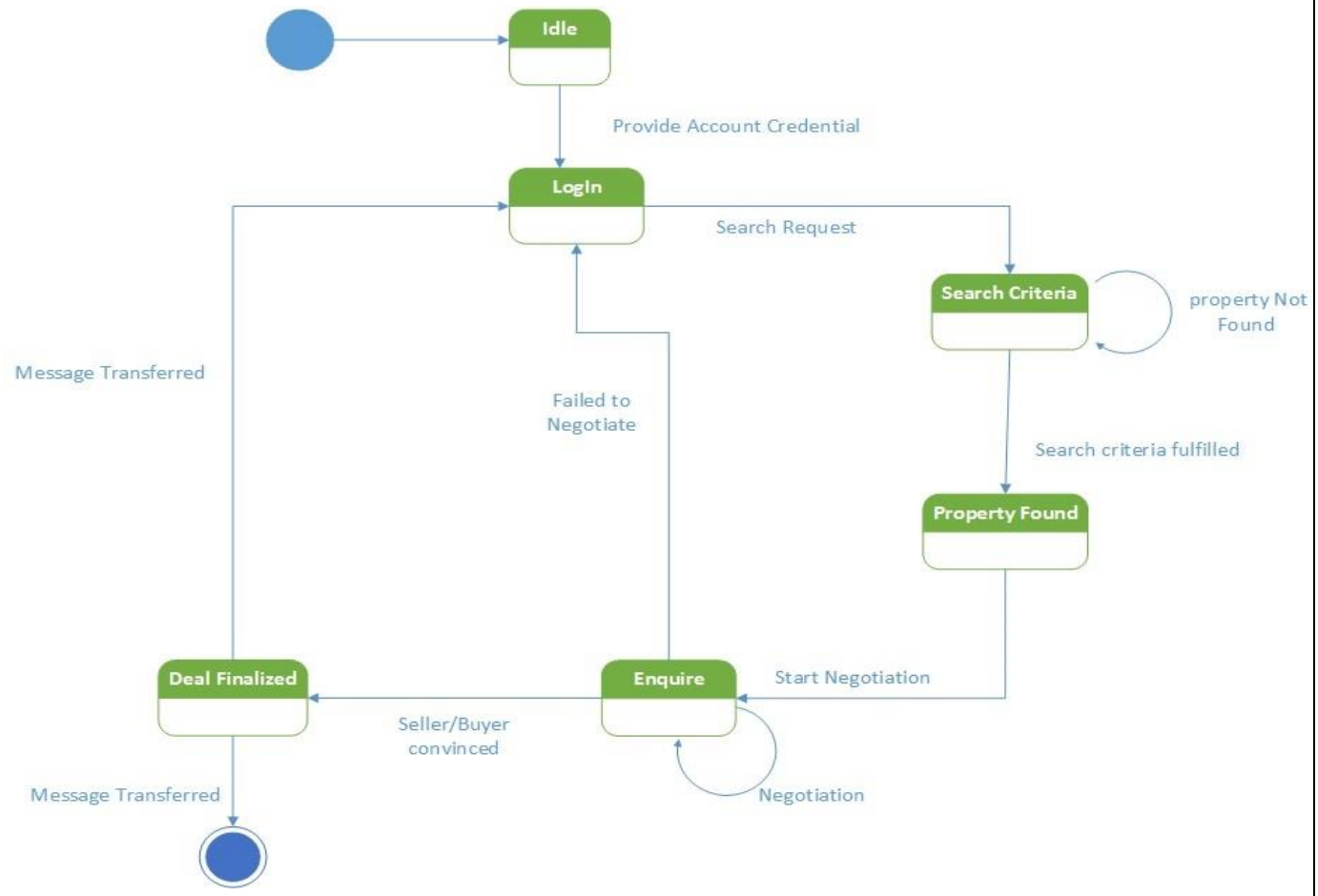


Figure 2: State diagram for "Inquiry"

Description

In this state diagram, we are considering that user has successfully create an account or already has an account. User enter account credentials to *login* and then generates a search request, If the search criteria is fulfilled user selects the property which sooths them else keep on searching. The user sends an *enquiry* message to the property seller and tries to negotiate. Once the deal is finalized the user is returned to *login* state.

Scenario 1: Deal Finalized

This state will be reached only if the buyer finds a property and generates an *enquiry* towards the seller and successfully negotiates with the seller

Scenario 2: Enquire

After reaching the state *property found* the user (buyer) has to generate an enquiry request towards the seller. If they fail to negotiate user will return to login state

III. State Diagram for class **User**

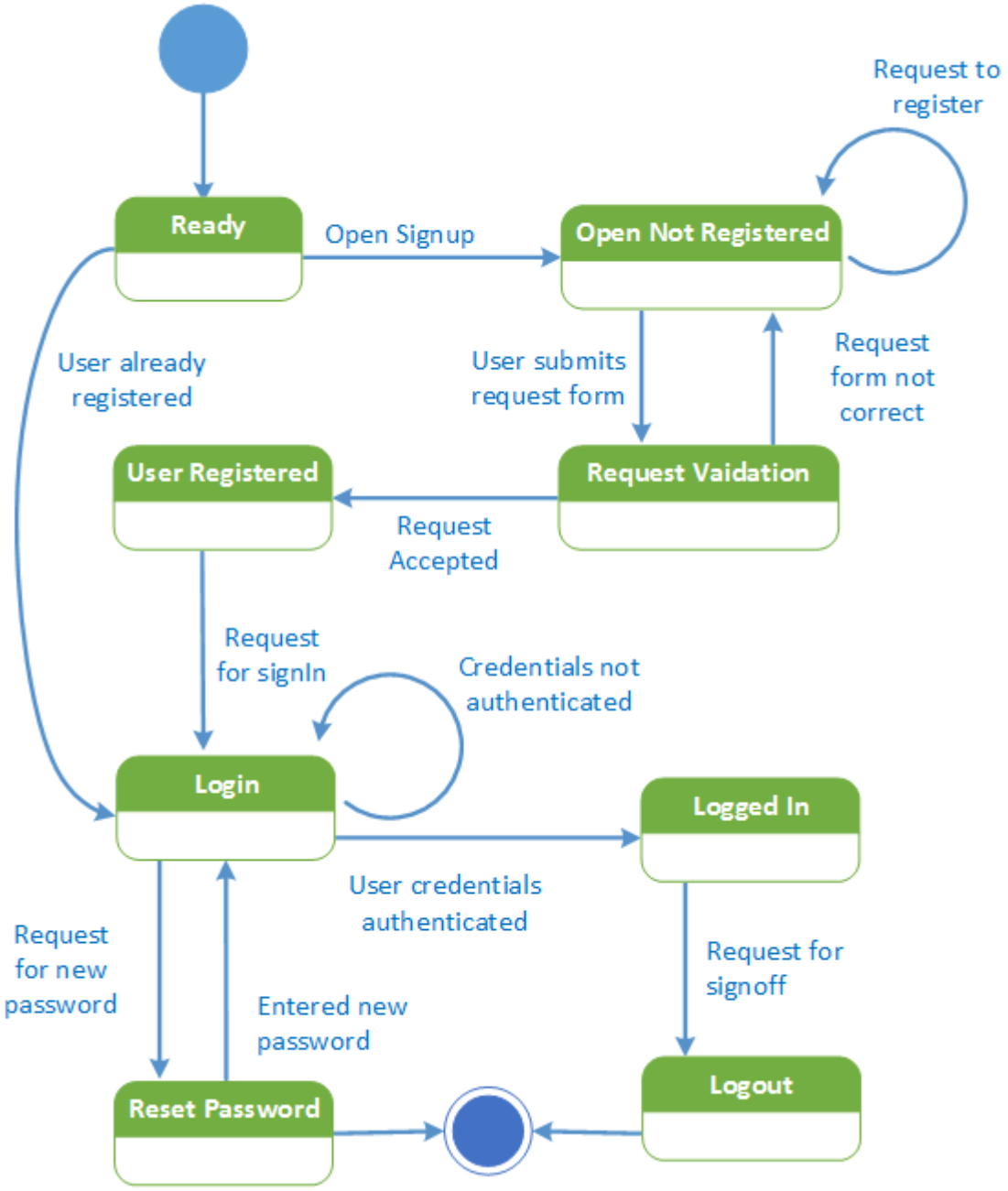


Figure 3: State Diagram for "User"

Description

This State machine diagram illustrates the state transition for class “User”. Initial state ‘Ready’ represents the system ready to accept either user login or new registration request. If user is not requested and upon registration request, system state changed to ‘Open not registered state” where user can raise request by filling user registration form. State won’t change until user fills in the correct form and submits the form. Upon user request to submit the form, system will change state to “request validation” state, where system verifies and validates the form for correctness and completeness. System accepts the registration request as “User registered” form. User can only Login into the system when system would be “Login” state and upon credential authentication, system could to “Logged in” state and would be ready to use. There is another state to reset the password and after user signoff the system state change to logout and won’t allow user to access the system.

IV. State Diagram for class **Transaction**

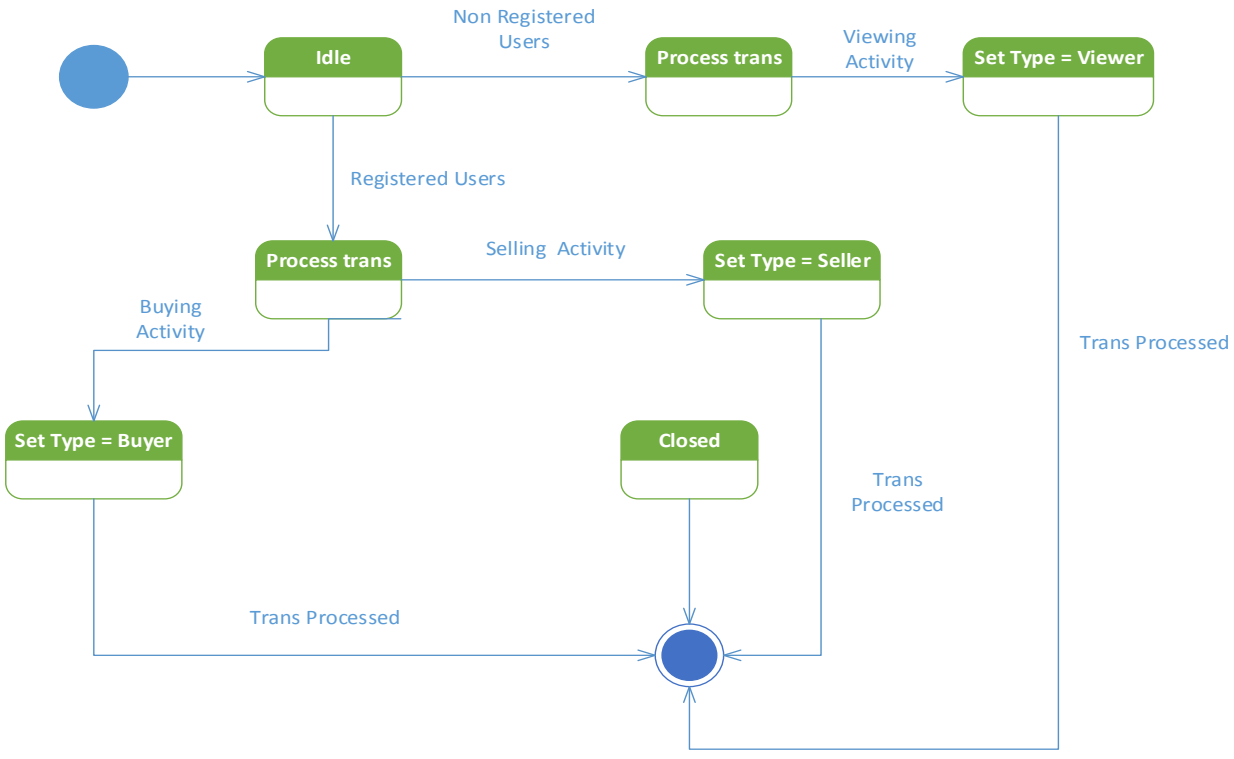


Figure 4: State Diagram for "Transaction"

Description

The Transaction state diagram shows the transaction types of the model. The user's transaction type is changed when his activity differs. The users involve in the state diagrams are **Seller, Buyer, and Viewer**. The registered users are sellers and buyers and Viewer is a Non Registered user. At the start system is in idle state and transaction is set to Viewer when the viewer is using the application to view the list of properties he intended to buy. This is done through the viewing activity after the Processed trans state is executed. Similarly, trans type is set to buyer and seller after buying and selling activity is processed. The system comes to the closed state after the user has perform its transactions.

Action Specification

UML relies action semantics for UML in late 2001.Actions semantics provide the specification of actions. An action language, called ALF (Action Language for Foundational UML) was later defined in late 2013. There are many languages that comply with the action semantics specification

1. Object Action Language (OAL) from BridgePoint
2. ALF (Action Language for Foundational UML)
3. Action Specification Language (ASL) from Kennedy Carter
4. SMALL (Shlaer – Mellor Action Language)
5. TALL (That Action Language)

Actions semantics provide the specification of actions. There are many languages that comply with the action semantics specification. The Action Specification for the operation of "Online Real Estate System" are specified as follows:

1. Creating a new Account and deleting the instance of the class Account.

Create object instance newAccount **of** Account; // creates a new instance of the class Account called newAccount

newAccount.name = "Adam";
newAccount.emailid= "Adam.gmail.com";
newAccount.Login_date= "1st January 1956";

x = newAccount.name; // x gets the value of the name
y = newAccount.emailid; // y gets the value of the email
z = newAccount.Login_date; // z gets the value of the login date

delete object instance newAccount // deletes the object newAccount

2. Adding a Property

Create object instance newProperty **of** Property; // creates a new instance of the class Property called newProperty

newProperty.Property_type = “Residential”;
newProperty.Property_id= “***”; // “***” is a system generated id
newProperty.Location = “Downtown”;
newProperty.Location = “Laval”;
newProperty.Location = “Lasalle”;
newProperty.Area = “1500”; // Area is measured in square feets’

x = newProperty.Property_type; // x gets the value of the name
y = newProperty.Property_id; // y gets the value of the id
z = newProperty.Location; // z gets the value of the location
w = newProperty.Area; // w gets the value of the area

3. Property

a. List all the properties where location is “Laval”.

select any Property **from instances of** Property **where** Property.Location == “Laval” ;

b. Returns all Property of the class Property whose area is 1500 sqfts or larger.

select many notSelectedArea **from instances of Property where** selected.area <= 1500;

if empty NewProperty // No such Property

4. Return all users who logged in to the system on 25th December 2015.

Select many user_name from instances of Account where Login_date == “25th December 2015”;

5. Increasing the Price of property whose area is between 2000 sqfts and 5000 sqfts by 15%.

select many notSoNewProperty from instances of Property
where selected.area < 5000 and selected.area > 2000;

for each Property in notSoNewPropertys
property.Price = property.Price * 1.15; // increase the price by 15%
end for;

References

[1] State Diagram: http://en.wikipedia.org/wiki/State_diagram
[2] State Diagram: http://web.engr.illinois.edu/~mfleck/building-blocks/version- 1.0/state-diagrams.pdf
[3] Action Semantic: Executable UML: A Foundation for Model-Driven Architecture" by S. Mellor and M. Balcer- Chapter7
[4] UML State chart Diagram: http://www.tutorialspoint.com/uml/uml_statechart_diagram.htm
[5] COEN6312 Winter 2016. Lecture notes