Project Title: UrbanGarden

Mohib Abbas Sayed – 2103158

Hamza Sayyed – 2103159

Om Shete – 2103163

EXPERIMENT. NO: 06

Aim: Develop Use Case Diagram for the project (Smart Draw, Lucid Chart).

Theory:

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

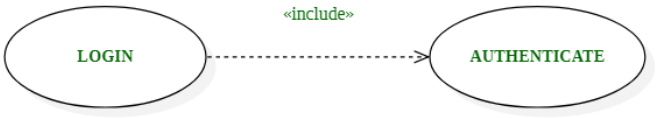
1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.

Difference between <<**include**>> and <<**extend**>> in **Use Case Diagram:**

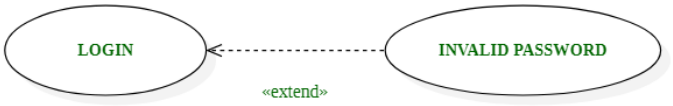
- <<***include***>> extends **Base Use Case** and it specifies that an **Included Use Case** must run successfully to complete Base Use Case. The Base Use Case is incomplete in the absence of an Included Use Case. The Included Use Case can be Base Use Case itself or it might be shared by a number of distinct Base Use Cases.

- <<***extend***>> on the other end , is used to add an **Extended Use Case** which extends the **Base Use Case.** Base Use Case can run successfully even without invoking/calling extended use case called Optional Use Case. The Base Use Case is complete in itself but under certain conditions it would require to refer to extension condition.

The representation of <<include>> and <<extend>> is as below :-



LOGIN is the BASE USE CASE and AUTHENTICATE is the INCLUDED USE CASE.

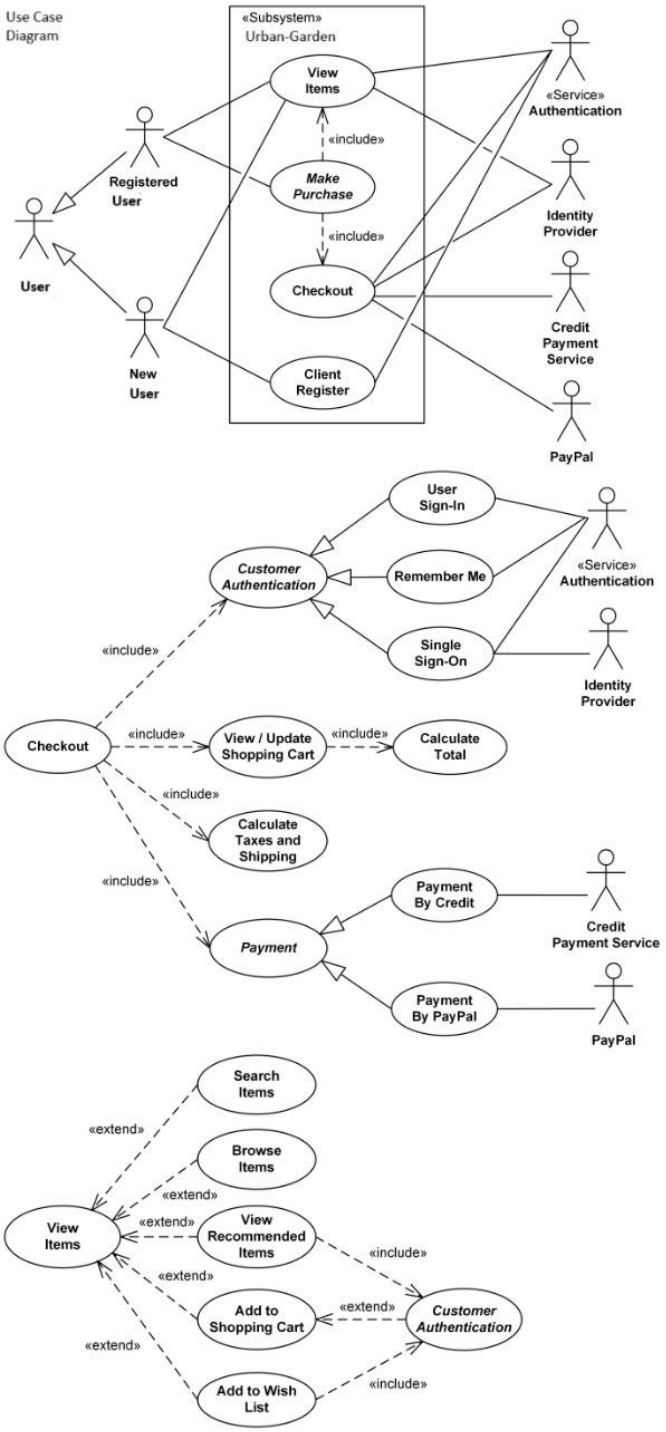


ILOGIN is the BASE USE CASE and INVALID PASSWORD is the EXTENDED USE CASE.

How to plan use case?
Following example will illustrate on how to plan use cases:

Use Case: What is the main objective of this use case. For eg. Adding a software component, adding certain functionality etc.
Primary Actor: Who will have the access to this use case. In the above examples, administrators will have the access.
Scope: Scope of the use case
Level: At what level the implementation of the use case be.
Flow: What will be the flow of the functionality that needs to be there. More precisely, the work flow of the use case.
Some other things that can be included in the use cases are:

- **Preconditions**
- **Postconditions**
- **Brief course of action**
- **Time Period**


Use Case Diagram
«Subsystem»
Urban-Garden
View Items
Make Purchase
Checkout
Client Register
«include»
«include»
Registered User
User
New User
«Service» Authentication
Identity Provider
Credit Payment Service
PayPal
User Sign-In
Customer Authentication
Remember Me
Single Sign-On
«Service» Authentication
Identity Provider
«include»
Checkout
«include»
View / Update Shopping Cart
«include»
Calculate Total
«include»
Calculate Taxes and Shipping
«include»
Payment
Payment By Credit
Credit Payment Service
Payment By PayPal
PayPal
Search Items
«extend»
Browse Items
«extend»
View Items
«extend»
View Recommended Items
«include»
«extend»
Add to Shopping Cart
«extend»
Customer Authentication
«extend»
Add to Wish List
«include»