Chirag Khatri - 1902070 - C21 DATE □□□□□□□□

Experiment - 5

Aim :- Identify all scenarios & develop use case diagram for the project.

Software used. Draw.io

Theory :-

Use case ~~digr~~ diagram

Actors

Actors represent anyone or anything that interact with the system. An actor may.

• Only input info to a system.

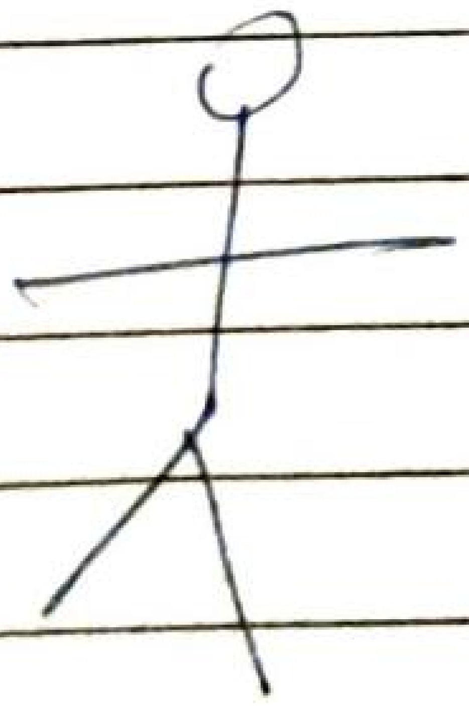
• Only retrieve info from a system.

• Both input and actors are found in the problem state. Typically, the actors are found in the problem State, and also from conversation with the customers a domain enterp.

There are 3 types of actors:

1. Users of the system
2. external application systems, and
3. external devices that can independe interact with the system.

In UML, an actor is represented stickma symbol, as shown below:-



Use case:

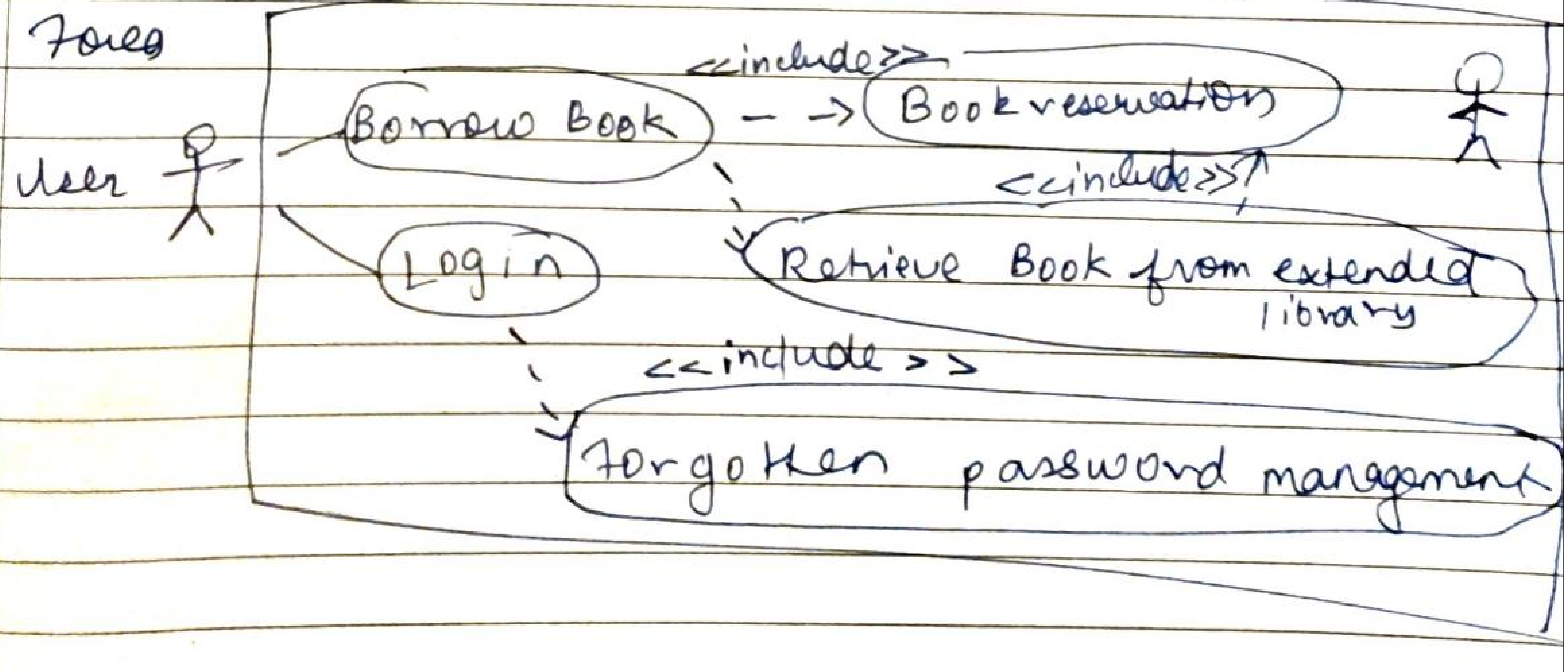
Use cases eventually map to the

Scanned by TapScanner

menu options. Use cases represent the funtionality provided by the system. Each individual funtionality provided by a system is capture as a use case.

A use case diagram thus represents a dialog between an actor and the system. A ~~va~~ collection of use cases for a system reflects all the defined ways in which a system can be used. A use case can be defined as a sequence of transactions performed by a system, that yields a measurable result of values for a partic actor. In UML, a use case is represented as an oval, as shown

A use case diagram is an interaction wise of a some or all the actors, use cases & their interactions identify for a system. Each system typically has a ~~man~~ main use case diagram, which is a picture of the system boundary & the major functionality provided by the system. Other use case diagram as needed.



DATE □□ □□ □□□□

Appendix 1) Stereotypes :- Defines a new model element in terms of another masil elemen

<< stereotypes >>

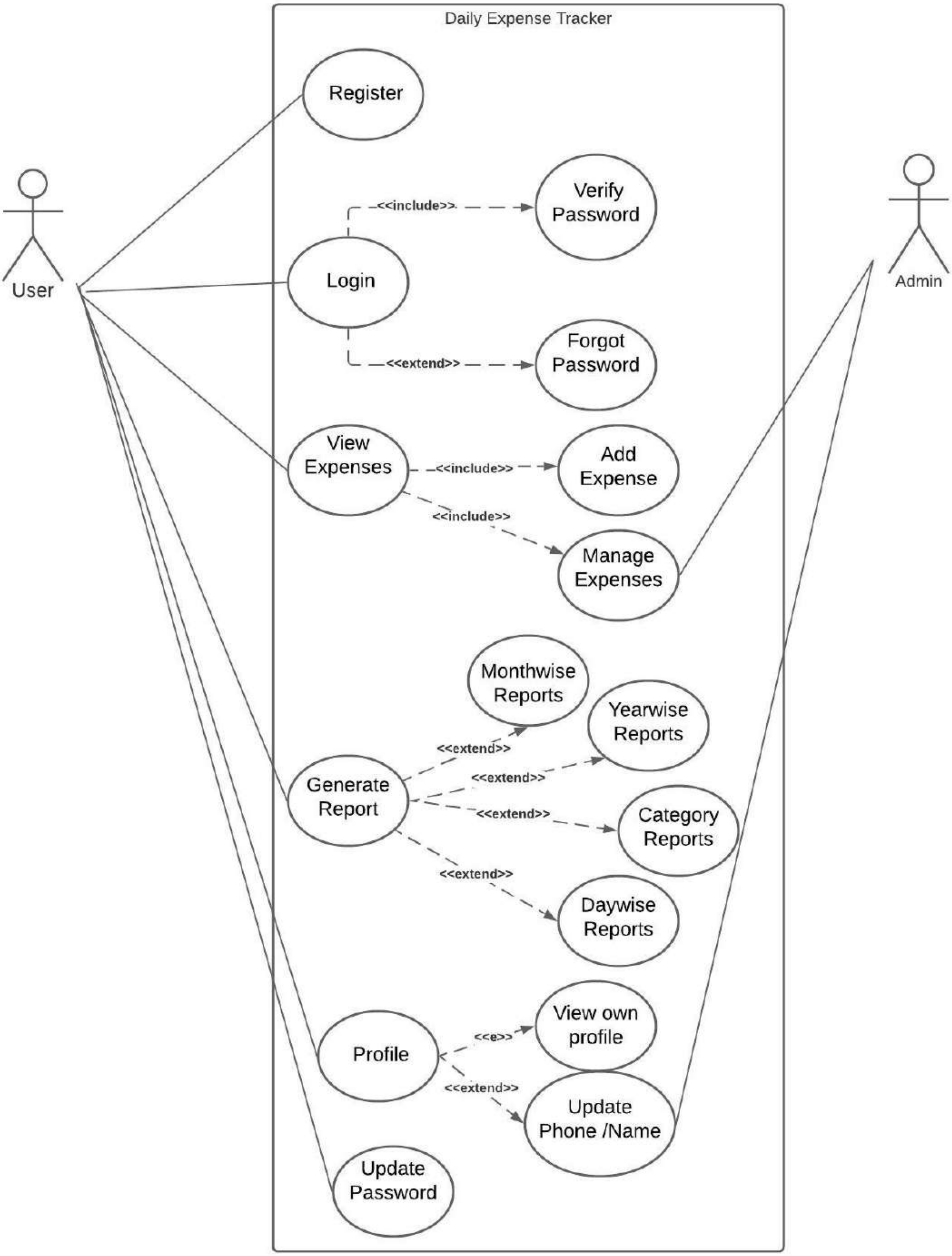
2) system boundary boxes :- Use can draw a vectangle ground, the uses cases, called the system boundary box.

3) Abstract use case :- Inherited use case are are called abstract use case which is directly inherited by actor is called connect use ase.

4) Concrete use case :- Use case, which is directlu inherited bis ator is callea concrete case

Conclusion : thus use case diagrams are understood.

Scanned by TapScanner


Daily Expense Tracker
User
Admin
Register
Login
<<include>>
Verify Password
<<extend>>
Forgot Password
View Expenses
<<include>>
Add Expense
<<include>>
Manage Expenses
Monthwise Reports
Yearwise Reports
Generate Report
<<extend>>
<<extend>>
<<extend>>
Category Reports
<<extend>>
Daywise Reports
Profile
<<e>>
View own profile
<<extend>>
Update Phone /Name
Update Password

Actors

1. User
2. Admin

Use cases

1. Register
2. Login
3. View Expenses
4. Generate Report
5. Profile
6. Update Password

Use case description

1. User- A person who is using the website to manage their expenses
2. Admin- admin can update user details and transactions when required
3. Register- user has to register to use the web app
4. Login- user can login through this section
5. View Expenses-In this section user can manage the expenses (add/delete).
6. Generate Report- : In this section, user can view expenses on day wise basis, month wise basis and year wise basis according to periods of time
7. Profile-In this section, user can update his/her profile.
8. Update Password- In this section, user can change his/her passwords