

Independent Work App



INDEPENDENT WORK APP

**SUBMITTED
TO
GUJARAT UNIVERSITY
&
C.U. Shah Institute Computer Application**

Independent Work App

A PROJECT REPORT

On

Independent Work App

SUBMITTED TO

GUJARAT UNIVERSITY



For partial Fulfillment Degree of

Bachelor of Computer Application (BCA)

SEM – 5 Software Development Project – 1

Submitted By :

KOLIPATEL PRAGNESH VINODBHAI

KUMBHANI KRUSHALI DINESHBHAI

RATHOD KRUPESH SANJAYBHAI

C.U. SHAH INSTITUTE OF COMPUTER APPLICATION

ASHRAM ROAD, AHMEDABAD – 380014

Website: <https://www.cushahbca.ac.in>

Contact No: +91635-925-9669

Year 2022-2023

C.U SHAH INSTITUTE OF COMPUTER APPLICATION

Near Gujarat Vidyapith, Income Tax Circle, Ashram Road, Ahmedabad-382330

Email : cushahbca@gmail.com

Contact no : +635-925-9669

Certificate of Completion

This is Certify that the following student of C.U. SHAH Institute of Computer Applications, Ahmedabad-14 from BCA semester V(Ty BCA) have satisfactory completed software development project PART 1 on

“Independent Work App”

As An Academic Requirement of BCA Semester-V as a partial fulfillment of the requirement for the degree of Computer Application (BCA) for the Academic year 2022-23.

Name of Student

Roll No.(College)

Seat No.(University)

KOLIPATEL PRAGNESH V.

3061

KUMBHANI KRUSHALI D.

3064

RATHOD KRUPESH S.

3152

Prof. Nilam Shrimali

(Internal guide)

Dr. Brijesh Bhatt

(I/C Principal)

(External Examiner)

Date: 07/07/2022

To,
C. U. Shah Institute of Computer Application,
Near Gujarat Vidhyapith, near income tax cross road,
Shreyas Colony, Navrangpura, Ahmedabad, Gujarat 380014

Subject: Confirmation letter of internship.

Respected Sir,

We are pleased to inform you that we have granted permission to the students of your college to work on their project on "**Independent Work App**" at our premises **BroadSy Technologies Pvt. Ltd.** They will work on Java along with Android Application.

We will provide best guidance from our side and expect good outcomes from students as well. They will work 6 hours for learning, 4 hours for Assignments and 4 hours for project work, totaling 14 hours per week. We will issue a completion certificate once they complete the project.

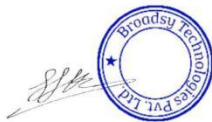
Name of Students:

No.	Name	Roll No.
1	Pragnesh Koli Patel	3061
2	Krushali Kumbhani	3064
3	Krupesh Rathod	3152

Sincerely,

Sandip Kumbhani

CEO



BroadSy Technologies PVT LTD

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

*We respect and thank **Prof. Nilam shrimali** for providing me an opportunity to do the project work in **C.U. SHAH INSTITUTE OF COMPUTER APPLICATION** and giving us all support and guidance, which made me complete the project duly.*

*We owe my deep gratitude to our guide **Prof. Nilam shrimali** who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.*

Last but not the least we are also thankful to our friends, project, partners, colleagues and parents for support and understanding the provided us during the project work. We are also very much thankful to all who directly or in directly helped us.

KOLIPATEL PRAGNESH V. [3061]

KUMBHANI KRUSHALI D. [3064]

RATHOD KRUPESH S. [3152]

INDEX

❖ Project Profile	[01]
❖ Company Profile	[02]
❖ Existing System	[03]
❖ Proposed System	[04]
❖ Development Tools and Technologies	[06]
❖ System Flow Diagram	[07]
❖ UML Diagrams	[11]
✓ Use case Diagram	[13]
✓ Activity Diagram	[19]
✓ Sequence Diagram	[30]
✓ Class Diagram	[39]
❖ ER Diagram	[42]
❖ Data Dictionary	[45]
❖ References	[54]

Project Profile

Project Name	Independent Work App
Objective	Any One Can Get Online Temporary Work Or Worker.
Group No	33
Group Member	3
Team Members	KOLIPATEL PRAGNESH V. KUMBHANI KRUSHALI D. RATHOD KRUPESH S.
Platform	Android Application
Institute	C.U. SHAH Institute of Computer Applications

Company Profile

Company Name	Broadsy Technologies Pvt. Ltd.
Address	B- 1009, 1010, Ratnakar Nine Square, Judges Bungalow Road, Ahmedabad-380015.
Owner Name	Sandip Kumbhani
Phone Number	+91 90336 02129

Existing System

- There is no any Existing System, All the Task Done manually On WhatsApp.
- Whenever Someone Need Temporary Work/Worker They Need More contacts/WhatsApp Groups.
- There are So Many WhatsApp Groups For This kind Of Works and So Many Group Gives Same Works But Different Payments For That Work And It Creates the Confusion to the user.
- Some time It is very Difficult to Manage All the Group Members Or Their Data.

Proposed System

❖ This System We Can Create Only One Part :

● Mobile Application :

- In this system both admin and customer will access different mobile applications.

❖ Following The Rules Of Users :

● Admin :

- Admin can Manage Detail About Worker / Organizers Or Events.

● Worker / Organizer :

- Organizer can Post Events(Work) and Worker Can apply to Join Event(Work).
- It Make Easy and Convenient Way to Find Work / Worker.

Advantages & Disadvantages

❖ Advantages :

- ✓ Easy to Find Work/Workers.
- ✓ Better Payment For Work.
- ✓ Less cut of commission.
- ✓ No Different WhatsApp Groups.
- ✓ Direct Interaction Between Organizer and Worker.
- ✓ More Secure Way rather than Whatsapp.

❖ Disadvantages :

- ✓ Risk of Fraud.
- ✓ You don't know exactly what work you are getting.

Development Tools & Technologies

❖ Tools

● Android

- Android Studio 2021.3.1

❖ Technologies

● Front - End

- Mobile Application
 - ✓ Android(Android Studio Dolphin 2021.3.1)

● Back - End

- Mobile Application
 - ✓ MySQL(Database)
 - ✓ SQLite(Database)

● Other Tools

- WPS Office (For Documentation , Presentation)
- draw.io (For Diagrams)

System Flow Diagram

- A System Flow Diagram is a way to show Relationship between a Business and its components, such as Customer(According to IT Toolbox.
- System flow diagram also known as “**process flow diagram**” or “**data flow diagram**”.

❖ What is system flow ?

- System flow are systems modes that show the activities and decisions that systems execute.





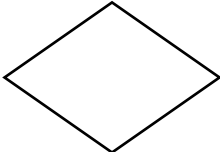

❖ What is system flow diagram ?

- A system flow diagram is basically a graphical and sequential representation of the major steps involved in a systematic process.

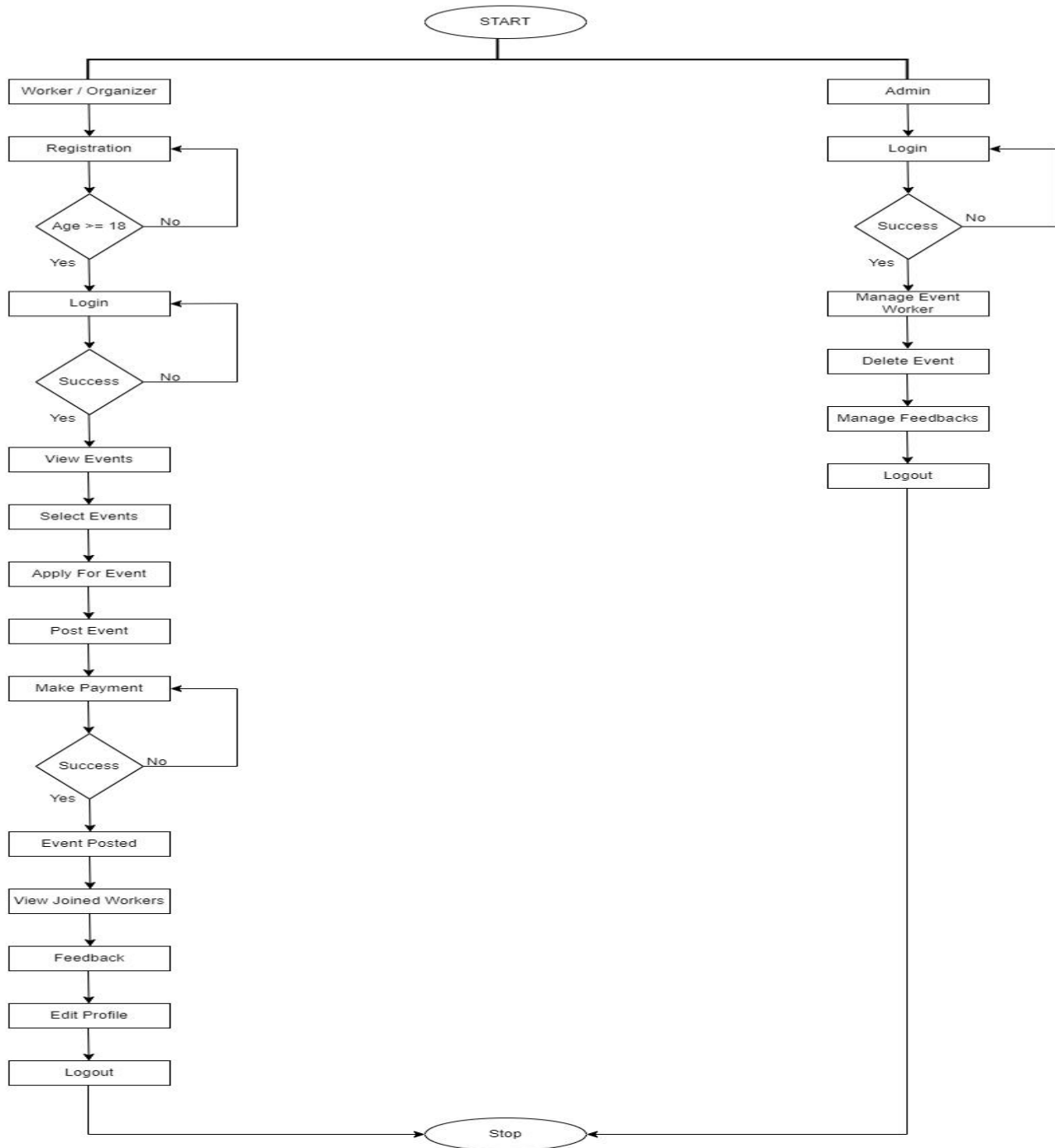
Symbols Using for System Flow Diagram :

Oval	An End or a Beginning.The oval is used to Represent the Start and End of a process.
Rectangle	A Step in the Following Process.The Rectangle is your do-to Symbol.
Arrow	Directional Flow.
Diamond	Call for a Decision.

Symbols Using For System Flow Diagram :

Symbol	Name
	Start
	Arrows
	Input / Output
	Process
	Decision
	End

System Flow Diagram



UML Diagrams

UML Diagram

❖ What is UML and what is used for?

- A UML is a Diagram based on the UML(Unified Modeling Language) with the purpose of visually Representing a system along with its main Actors,roles,actions,artifacts or classes, in order to better understand,alter,maintain or Document information about System.
- UML used by Software Developers.
- UML can be used to Develop Diagrams and provide users with ready-to-use,expressive modeling examples.

❖ Why do we use UML Diagrams?

- UML has Applications beyond Software Development, such as process Flow in Manufacturing.
- It is Analogous to the Blueprints used in other Fields, and Consists of different types of diagrams.
- In the Aggregate, UML Diagrams describe the Boundary, Structure and the behavior of the system and the objects within it.

❖ Why are UML Diagrams Important?

- The Important of using UML for Modeling.
- UML is a powerful tool that can greatly improve the quality of your System Analysis and Design, and it is hoped that the improved practices will translate into higher quality System.

❖ What is the need of UML Diagrams?

- “Importance of UML Diagrams in Software Development”.
- The Unified Modeling Language (UML) is a Standard Language for Specifying Visualizing, Constructing and Documenting the Artifacts of Software for System as well as for Business modeling and other Non-Software System.

❖ Advantages :

- Provides standard for Software Development.
- Reducing of Costs to Develop Diagrams of UML using supporting tools.
- Development time is reduced.
- Development time is reduced.
- The past faced issues by the Developers are no longer exists.

❖ Types of UML Diagram :

1. Object Diagram
2. Use Case Diagram
3. Activity Diagram
4. Sequence Diagram
5. Class Diagram

Use Case Diagram

Use Case Diagram

❖ Purpose of Use case Diagram :

- Use Case Diagram is one of them and its specific Purpose is to gather System Requirements and Actors.
- Use case Diagrams Specify the events of a system and their flows.
- But Use case Diagram never describes how they are implemented.

❖ A Use case describes a sequence of actions that provide something of measurable value to an Actor and is drawn as a Horizontal ellipse.

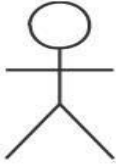


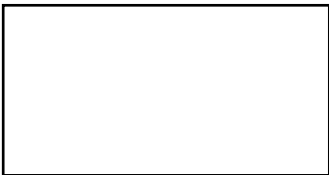
❖ Importance of Use cases :

- Use cases are important because they are in a tracking format. Hence they make it easy to comprehend about the functional Requirements in the system and also make it easy to identify the various interactions between the Users and the System within an Environment.

Use Case Diagram Notations :

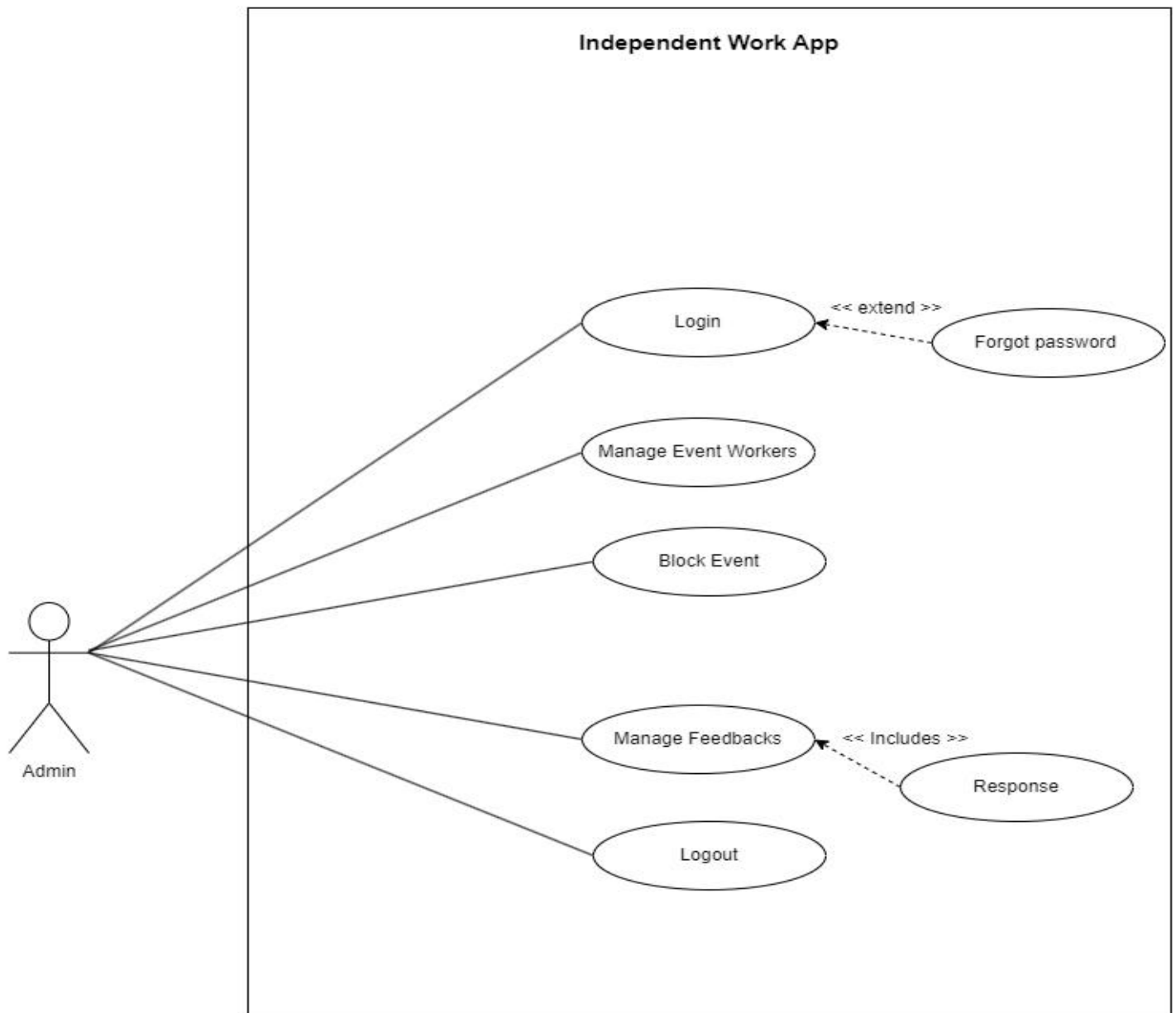
Actor	Actor are usually individuals involved with the system defined According to their roles.
Use Case	A Use case describe how Actor uses a system to accomplish a particular goal.
Relationship	The Relationships between and among the Actor and the Use cases.
System Boundary	Boundary of inside whole system But Actor are Outside of the System Boundary.

Symbols Of Use Case Diagram :

Symbol	Name	Description
	Actor	Include all user roles that interact with the system and system components only if they responsible for initiating/triggering a use case
	Use Case	System function (process – automated or manual).
	Relationship	Connection between Actor and Use Case.
<<include>>	Include	Include relationship between Use Cases(one UC must call another; e.g., Login includes User Authentication UC).
<<extend>>	Extend	Extend relationship between Use Cases(one calls Another under certain condition; think of if-then decision points).
	System Boundary	Boundary of system

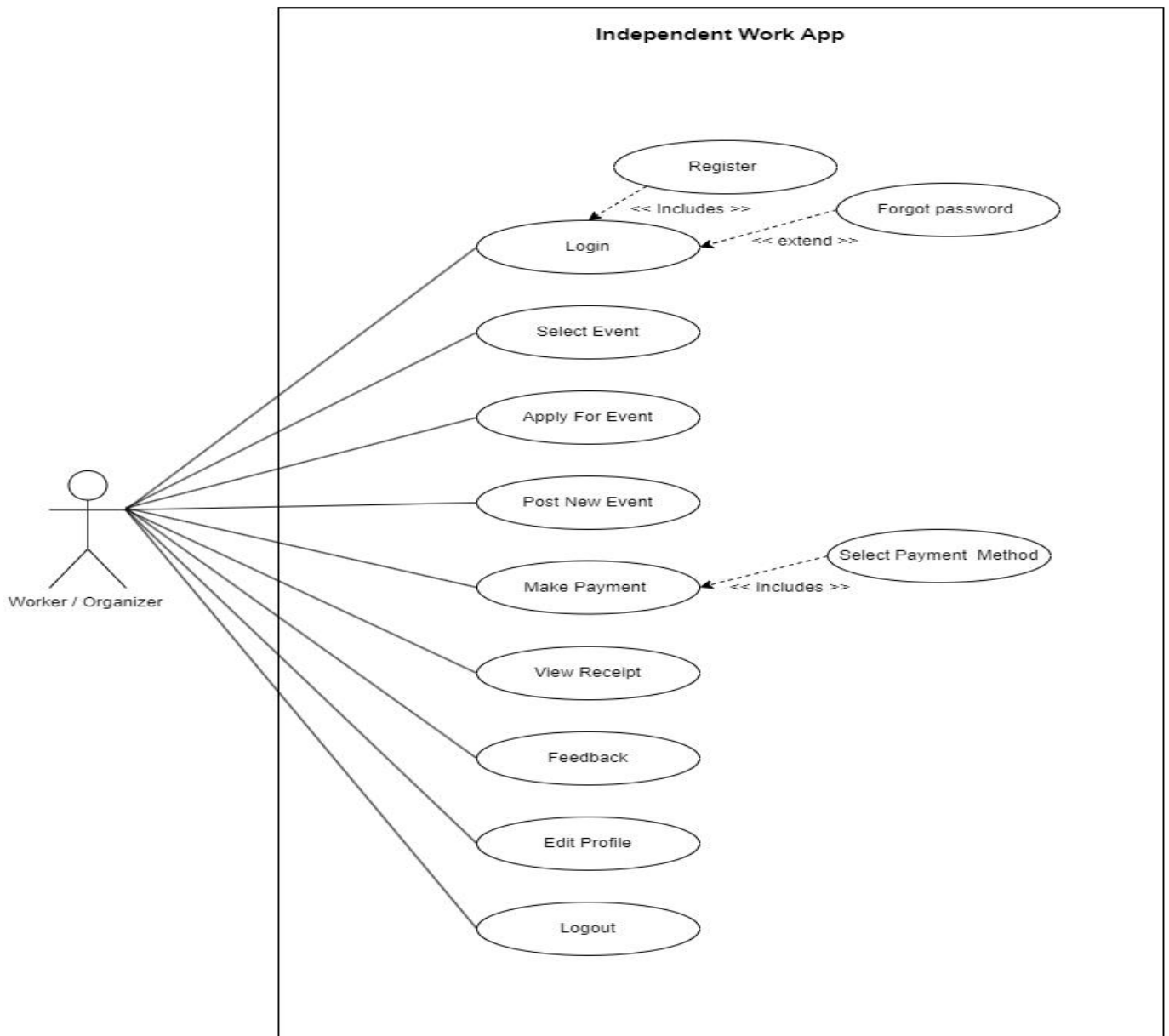
Use Case Diagram :

➤ Admin Side :



Use Case Diagram :

➤ User Side :



Activity Diagram

Activity Diagram

❖ What is Activity Diagram Definition?

- In Unified Modelling Language (UML), an Activity Diagram is a Graphical Representation of an executed Set of Procedural System.
- Activities and considered a State Chart Diagram variation.
- Activity Diagrams Describe Parallel and Conditional Activities Use cases and System Functions at a Detailed Level.

❖ Why do we use Activity Diagram?

- The Basic Usage of Activity Diagram is Similar to other Four UML Diagrams. The Specific Usage is to Model the Control Flow from one Activity to another.
- This Control Flow does not include Messages.
- Activity Diagram is Suitable for Modelling the Activity Flow of the System.

❖ How do You Write an Activity Diagram?

- In Each of these Cases, Here's How to Draw An Activity Diagram Form the Beginning.
 - **Step 1** : Figure out the Action steps From the Use case.
 - **Step 2** : Identify the Actor who are involved.
 - **Step 3** : Find a Flow among the Activities.
 - **Step 4** : Add Swim lanes.

❖ What are the Elements of Activity Diagram?

- Fundamental Elements of the Activity are Actions and Control Elements :
 - Decision
 - Division
 - Merge
 - Initiation etc...
- Elements are connected by so-called “**Activity Edges**” and From the “**Control Flow**”, which can also be casually called “**Flow**”.
- The execution of an Activity can contain Parallel Flows.




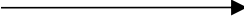
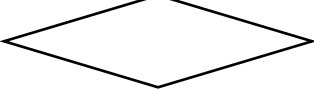
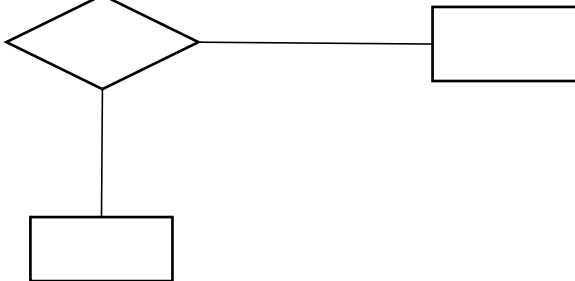
❖ What is Swim lane Diagram?

- Swim lane From Wikipedia, the Free Encyclopedia.
- A Swim lane Diagram is used in process Flow Diagram of Flowcharts, that visually Distinguishes Job Sharing and Responsibilities for Sub-Processes of a Business Process.
- Swim lanes may be arranged either “**Horizontally**” or “**Vertically**”.

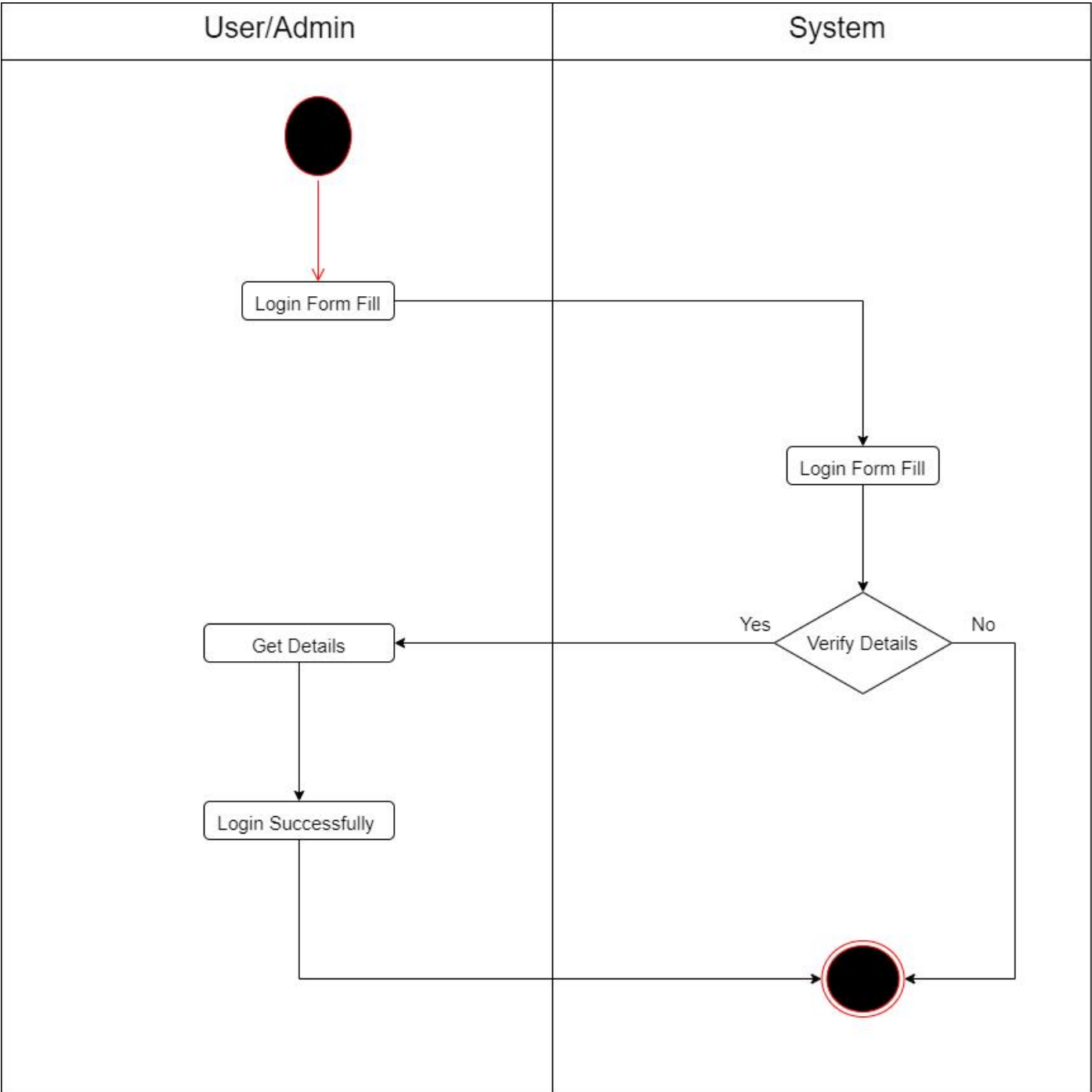
❖ What is Swim lane in Activity Diagram?

- A Swim lane Diagram is a type Of Flowchart that Delineates who does what in a Process.
- Using the Metaphor of Lanes in a Pool.
- A Swim lane Diagram Provides **Clarity** and **Accountability by Process Steps** within the **Horizontal** or **Vertical** “**Swim lanes**” of a Particular Employee Work Group or Department.

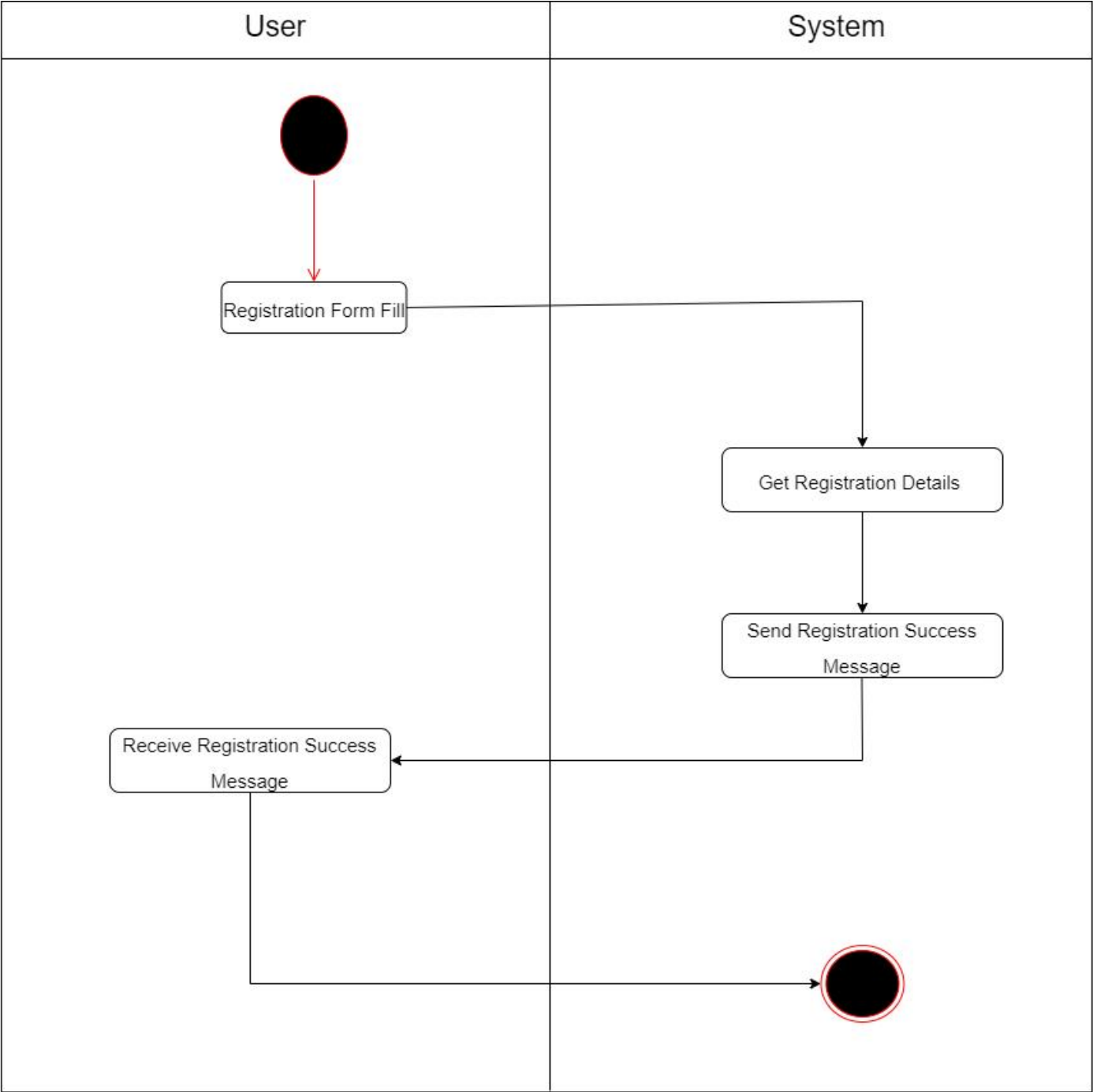
Symbols Of Activity Diagram :

Symbol	Overview
Start point/initial State 	A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swim lanes, make sure the start point is placed in the top left corner of the first column.
End point symbol 	An arrow pointing to a filled circle nested inside another circle represents the final action State (Ending Point).
Activity 	An action state represents the non-interruptible action of objects. You can draw an action state in Smart-Draw using a rectangle with rounded Corners.int could Activity.
Action flow 	Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.
Decision symbol 	A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."
Guard system 	In UML, guards are a statement written next to a decision diamond that must be true before moving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.

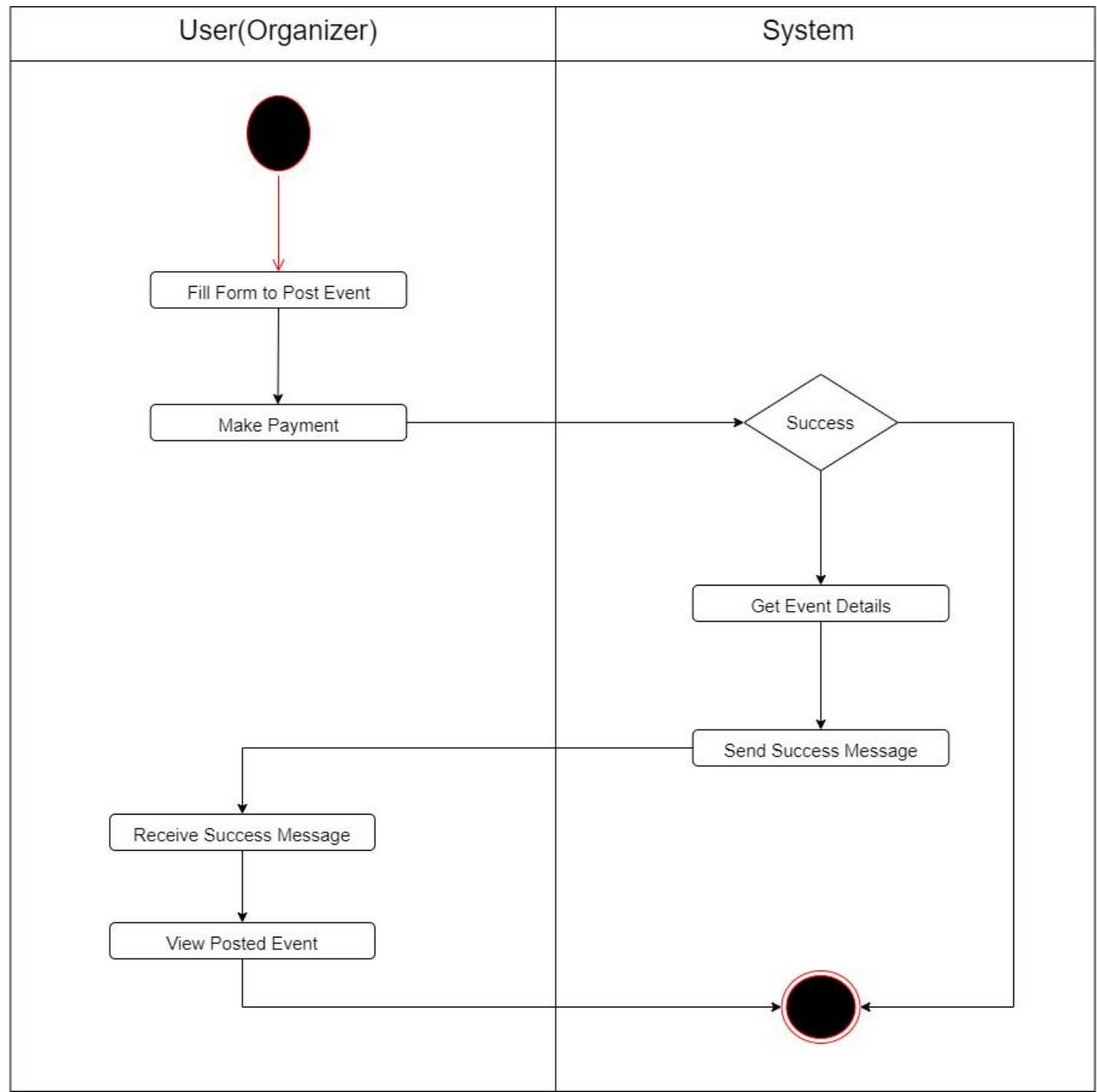
Login



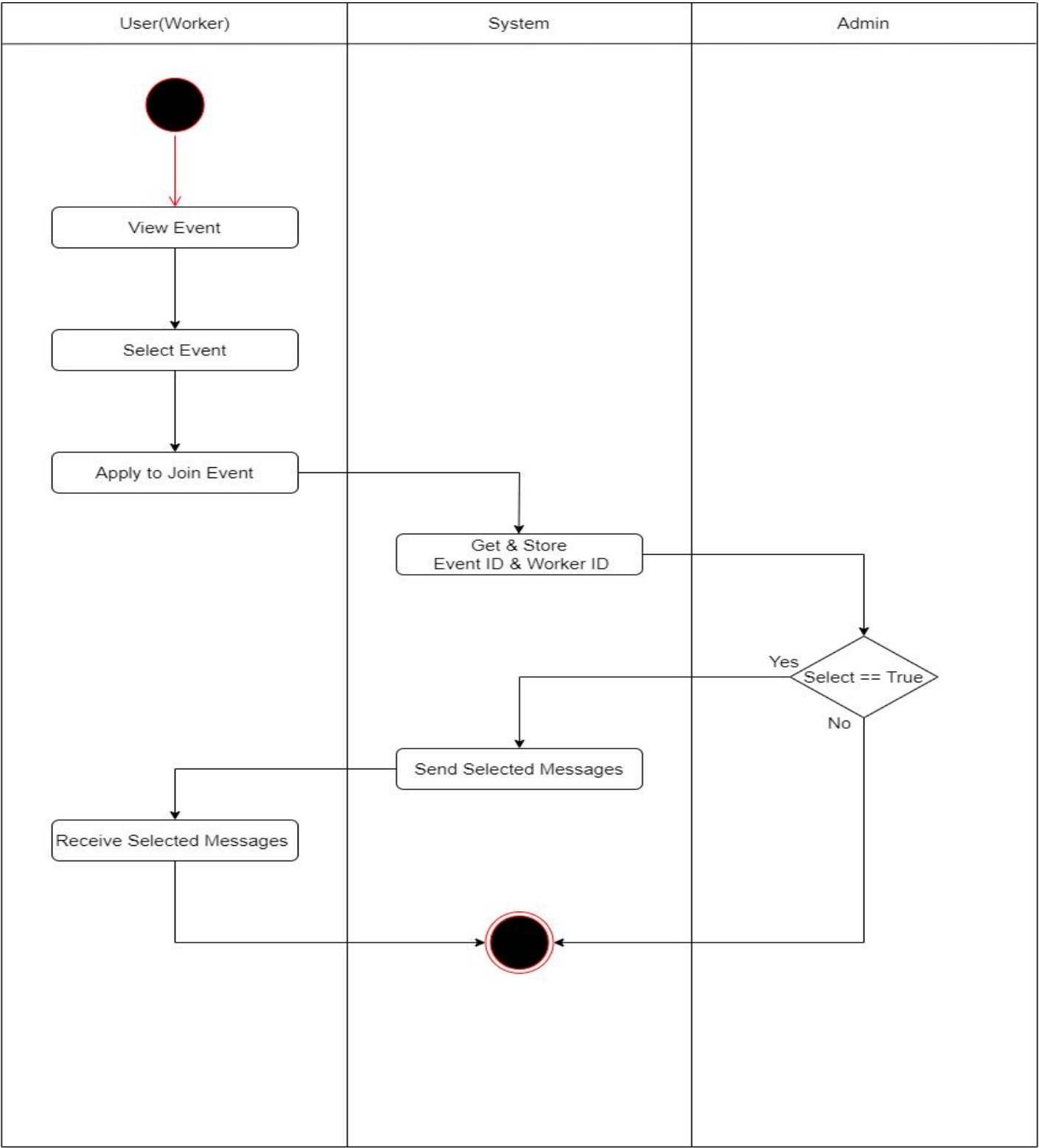
Registration



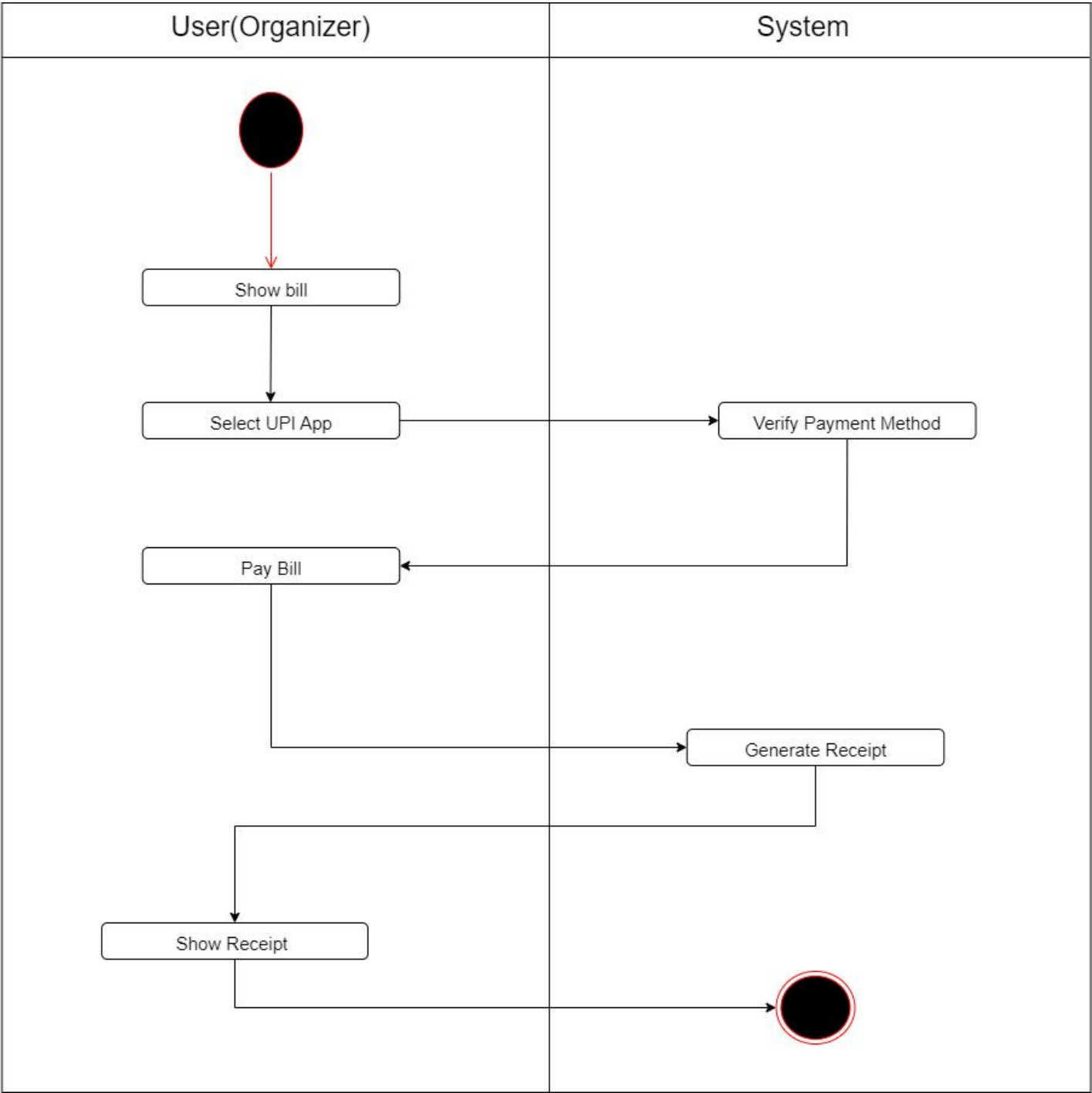
Post Event



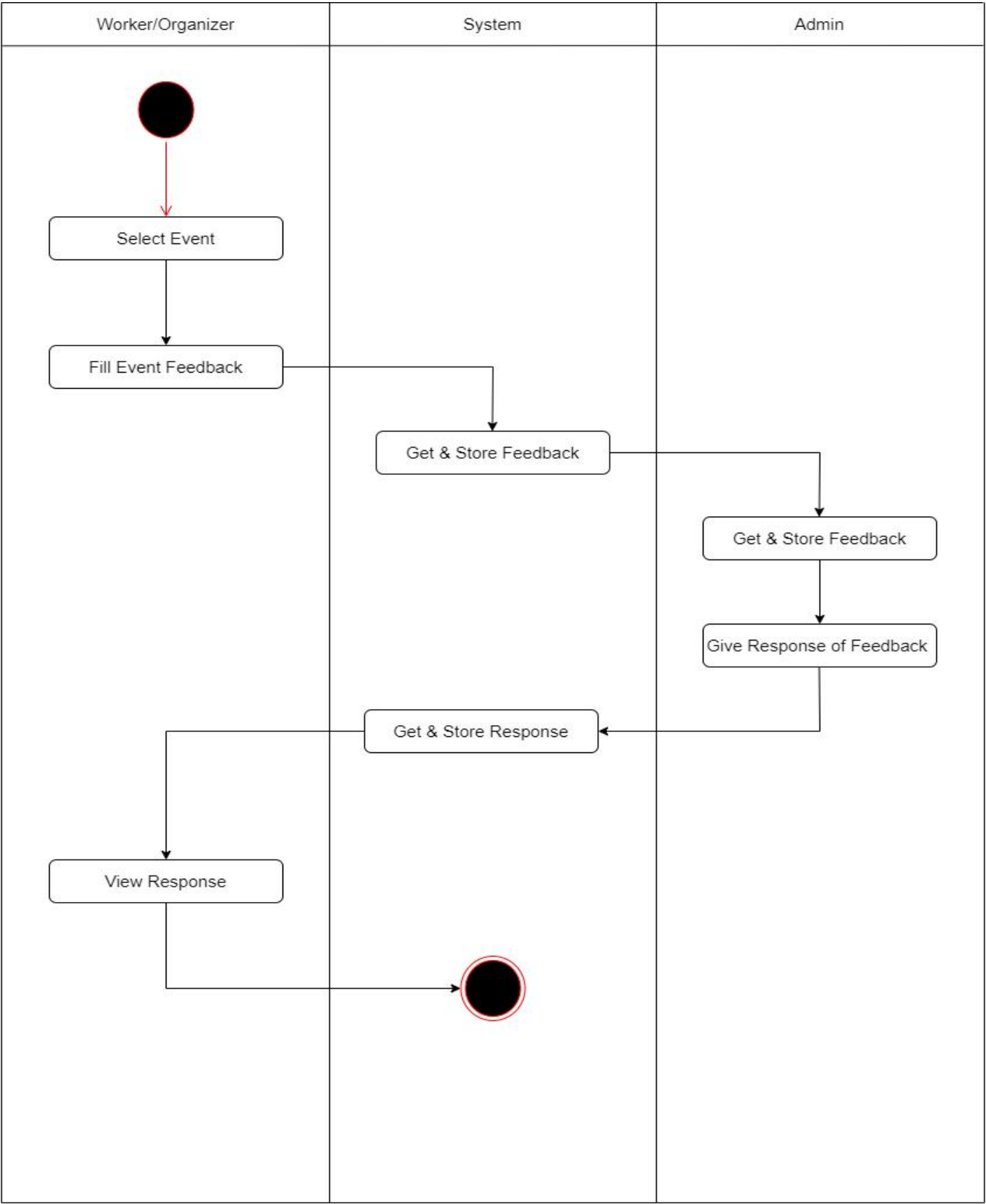
Apply Event



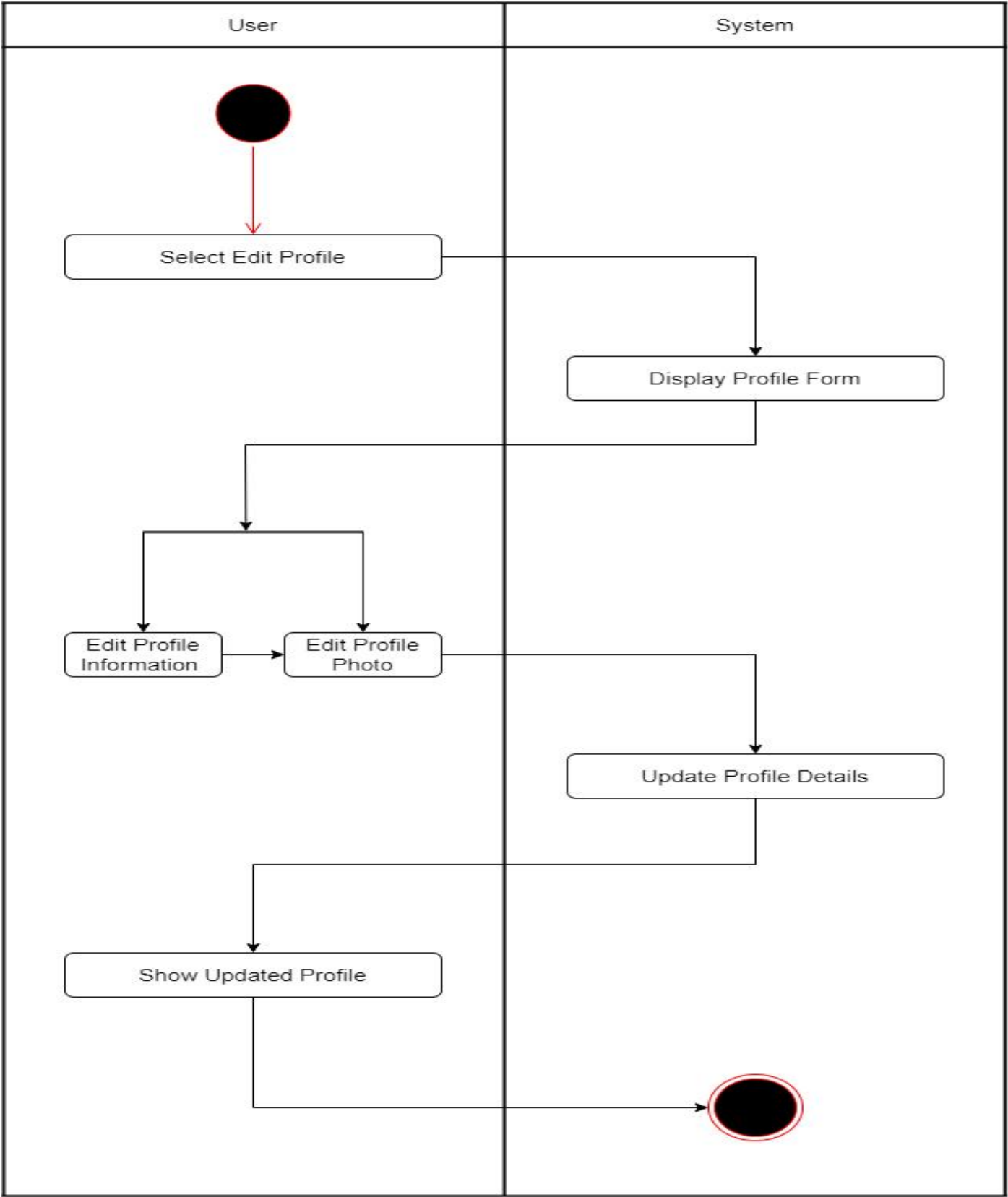
Payment



Feedback



Edit Profile



Sequence Diagram

Sequence Diagram

❖ What is Sequence Diagram?

- Sequence Diagram Tutorial.
- A Sequence Diagram Describes an interaction among a Set of Object Participated in a Collaboration or Scenario, Arranged in a Chronological order.
- It Shows the Objects Participating in the interaction by their “**Lifelines**” and “**Messages**” that they send to each other.

❖ What is Sequence Diagram used for?

- UML Sequence model the flow of Logic within your System in a Visual manner, enabling you both to Document and validation your Logic and are commonly used for both Analysis and Design Purposes.

❖ Why is Sequence Diagram used?

- The Sequence Diagram is a good Diagram to use to Document a System’s Requirements and to flush out a System’s Design.
- The Reason the Sequence Diagram is so useful is because it shows the interaction Logic between the Objects in the System in the time order that the in interaction take place.

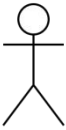
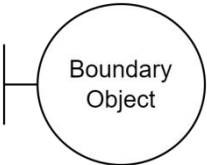
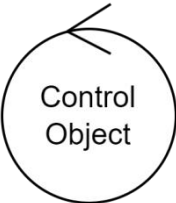
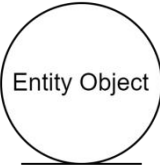
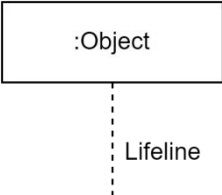
❖ Basic Sequence Diagram Notations :

- Class Roles or Participants. Class roles describe the way an object will behave in Context.
- Activation or Execution Occurrence. Activation boxes Represent the time an Object needs to complete a task.
- Messages.
- Lifeline.
- Destroying Objects.
- Loops.
- Synchronous Message.
- Asynchronous Message.




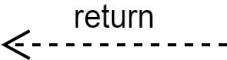
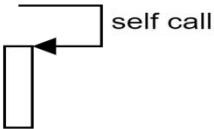
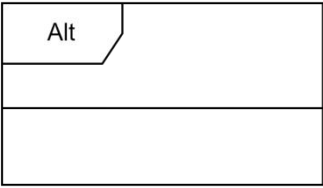
❖ What are the Elements of Sequence Diagram?

- The Following **Nodes** and **Edges** are typically Drawn in a UML Sequence Diagram :
 - Lifeline
 - Execution Specification
 - Message
 - Combined Fragment
 - Interaction Use
 - State invariant
 - Continuation
 - Destruction Occurrence

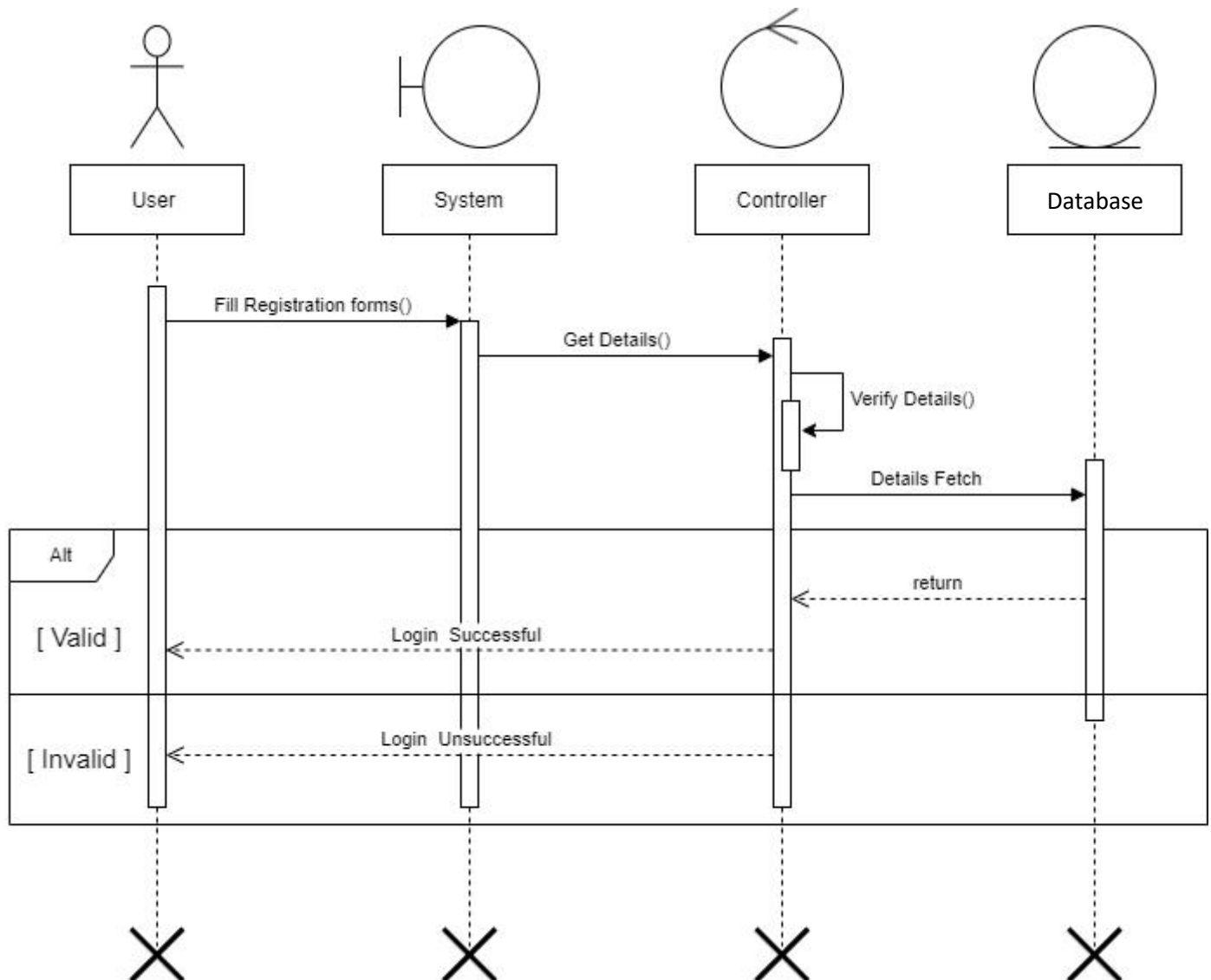
Symbols Of Sequence Diagram :

Symbols	Overview
 <p>Actor</p>	<p>Actors are the entities that interact with a system. Although in most cases, actors are used to represent the user of system, actors can actually be anything that needs to exchange information with the system. So, an actor may be people, computer hardware, other system, etc.</p>
 <p>Boundary Object</p>	<p>Objects that interface with system actors (e.g. a user or external service). Windows, screens and menus are examples of boundaries that interface with users.</p>
 <p>Control Object</p>	<p>Objects that mediate between boundaries and entities. These serve as the glue between boundary elements and entity elements, implementing the logic required to manage the various elements and their interactions. It is important to understand that you may decide to implement controllers within your design as something other than objects - many controllers are simple enough to be implemented as a method of an entity or boundary class for example.</p>
 <p>Entity Object</p>	<p>Object representing system data, often from the domain model.</p>
 <p>Lifeline</p>	<p>A sequence diagram is made up of several of these lifeline notations that should be arranged horizontally across the top of the diagram. No two lifeline notation should overlap each other. They represent the different objects or parts that interact with each other in the system during the sequence.</p>

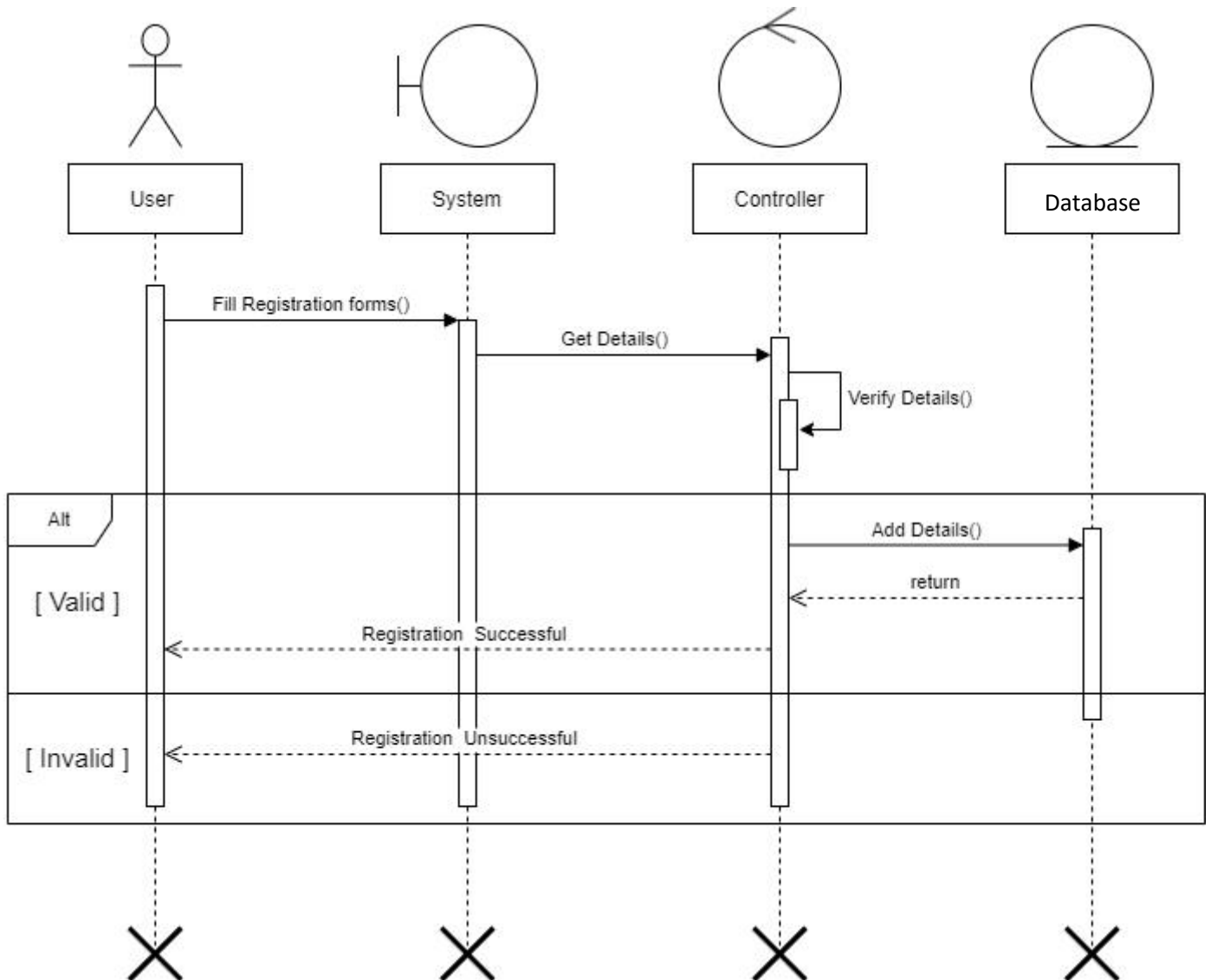
Independent Work App

 <p>Activation</p>	<p>Activation boxes represent the time an object needs to complete a task. when an object is busy executing a process or waiting for a replay message, use a thin gray rectangle placed vertically on its lifeline.</p>
	<p>This Symbol is Known as lifeline end Symbol work with lifeline.</p>
	<p>An asynchronous message is used when the message caller does not wait for the receiver to process the message and return before sending other messages to other objects within the system. The arrowhead used to show this type of message is a line arrow like show in the example below.</p>
	<p>A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline.</p>
	<p>A message an object sends to itself, usually shown as a U shaped arrow pointing back to itself.</p>
	<p>This Symbol is cloud as alternate. This symbol have two parts valid and invalid part. This symbol is check condition of message when condition is true ten pass positive response otherwise pass negative response.</p>

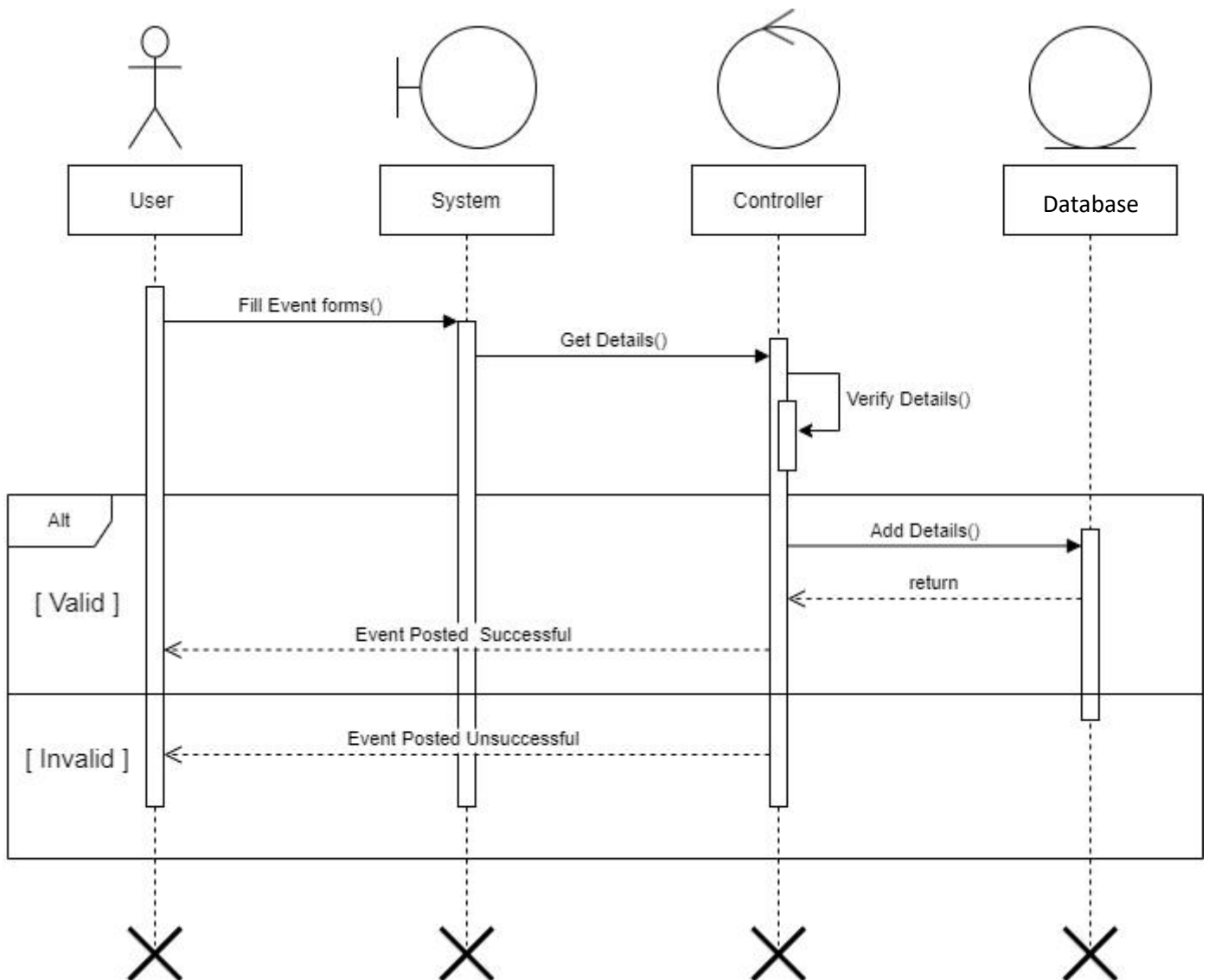
Login



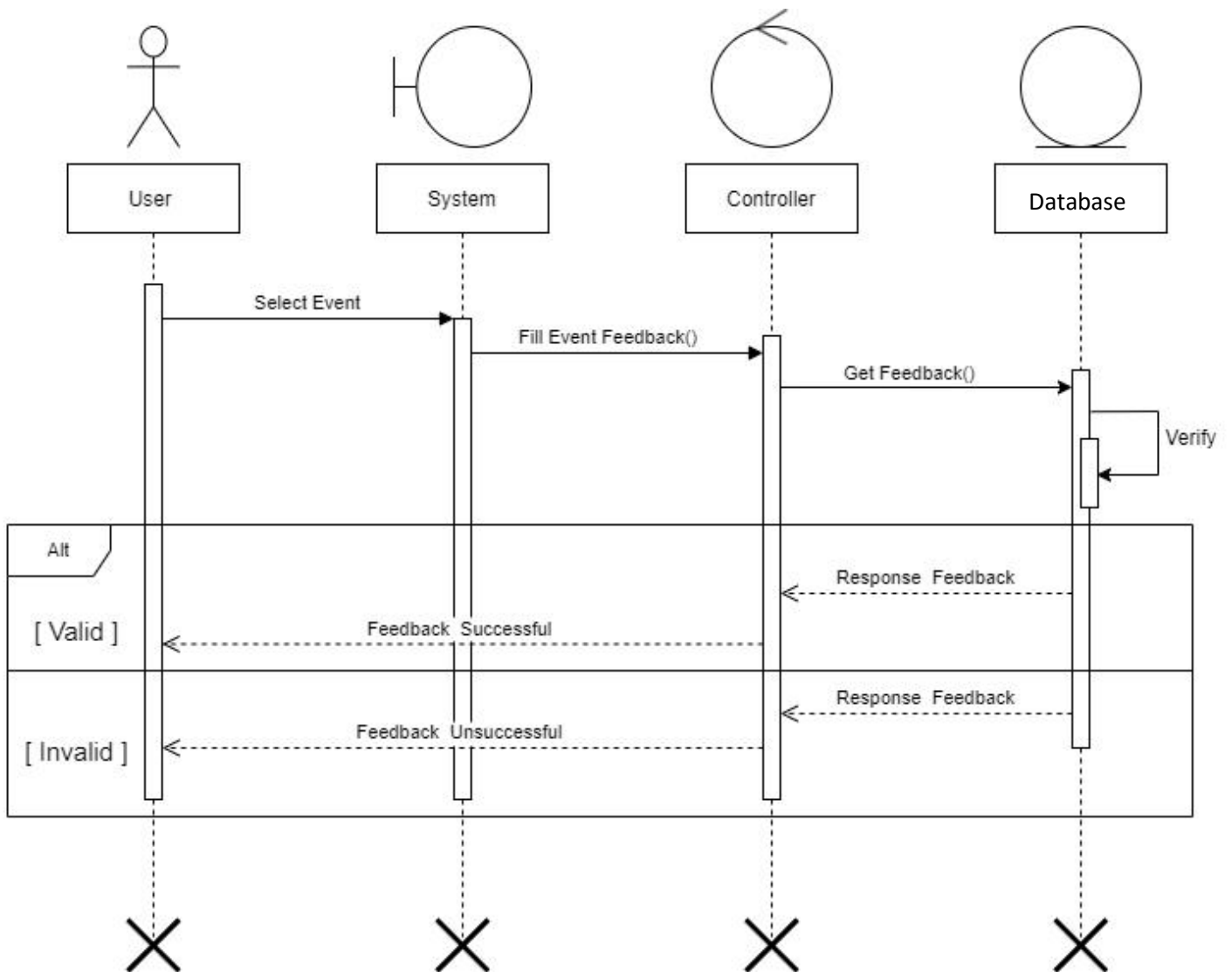
Registration



Post Event



Feedback



Class Diagram

Class Diagram

❖ What is Class Diagram?

- A Class Diagram is an illustration of the Relationship and some code Dependencies among Classes in the Unified Modeling Language(UML).
- In this Context a class Defines the Methods and variables in an Object. Which is a Specific Entity in a Program of the unit of Code Representing the Entity.

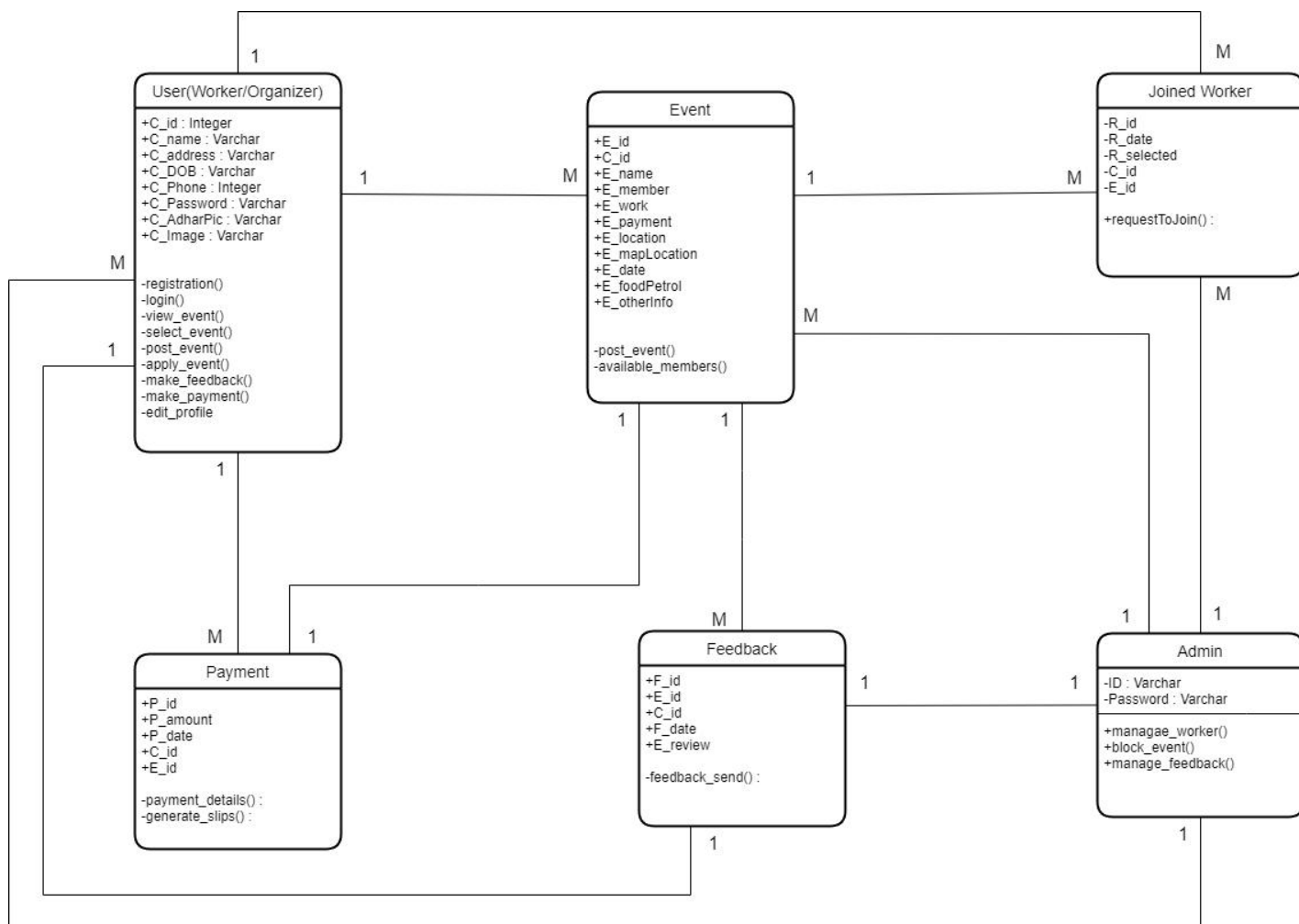
❖ How do you identify Classes in the class Diagram?

- Analysis Activities include.
- Identifying Objects (often from use case as a starting point).
- Identifying Associations between Objects.
- Identifying general Attributes and Responsibilities of Objects.
- Modeling interactions between Objects.
- Modeling how individual objects change state identify Operations.

❖ How do you identify Classes in the class Diagram?

- In Class Diagrams as work with the following elements :
 - Class. A Class represents a Relevant Concept from the Domain, a set of Persons, Object or ideas that are depicted in the IT System.
 - Attribute.
 - Generalization.
 - Association.
 - Multiplicity.
 - Aggregation.

Class Diagram



ER Diagram

ER Diagram

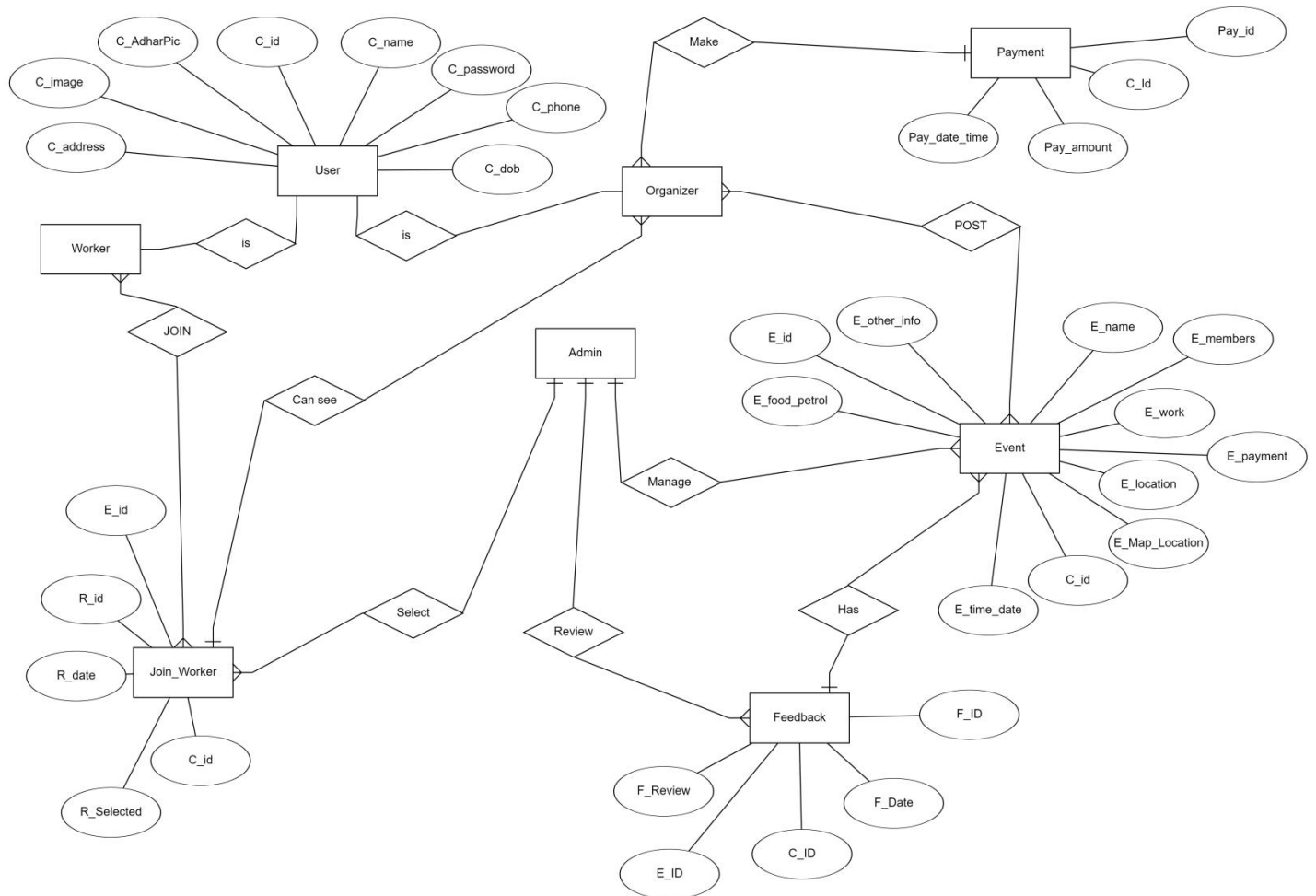
❖ What is ER Diagram?

- An Entity is an Object or Component of Data.
- An Entity is represented as Rectangle in an ER Diagram.

❖ Why we used ER Diagram?

- ER Diagrams are most often used to Design or Debug Rational Database in the Fields of Software Engineering , Business information System, Education and Research.
- ER Diagram also are often used in conjunction with Data Flow Diagram (**DFDs**), which map out the Flow of Information for Processes or System.

ER Diagram



Data Dictionary

Data Dictionary

❖ What is Data Dictionary?

- A data dictionary is a repository of metadata. Metadata is data about data.
- Some example of what might be contained in an organization's data dictionary include : The data types, e.g., integer,real,character, an image of all fields in the organization's database.

❖ What are the benefits Data Dictionary?

- There are number of advantages of using Data Dictionary in computer system analysis and design.
- The advantages are : consistency,clarity; reusability; completeness; increase in sharing and integration; and ease of use for the developer.

❖ Why we need a Data Dictionary?

- A successful data dictionary can improve the reliability and dependability of an organization's data , reduce redundancy, improve documentation and control, and make it easier to analyze data and use it to make evidence-based care decision like those common in accountable care organizations.

Admin : Store the information About Authentication

- This Table Shows the Details of the Admin.

Primary Key : ---

TABLE : Admin

Foreign Key : ---

Field Name	Datatype (Size)	Collection	Constraints	Description
ID	Varchar (15)	Utf8_unicode_ci	Unique / Not Null	Generated by Admin
Password	Varchar (30)	Utf8_unicode_ci	Not Null	Generated by Admin

- This Table Shows the Details of the Admin (ID,Password).
- ID is Unique and Password is Not Null in this Table.

User : Store the information About User

- This Table Shows the Details of the User.

Primary Key : C_Id

TABLE : User

Foreign Key : ---

Field Name	Datatype (Size)	Collection	Constraints	Description
C_Id	Integer (6)	-	Primary Key / Auto Inc.	Auto Generated by the Database
C_Name	Varchar (50)	Utf8_unicode_ci	Not Null	Generated by the End User
C_Password	Varchar (30)	Utf8_unicode_ci	Not Null	Generated by the End User
C_Address	Varchar(200)	Utf8_unicode_ci	Not Null	Generated by the End User
C_Phone	Integer (10)	-	Not Null	Generated by the End User
C_DOB	Varchar (10)	Utf8_unicode_ci	Not Null	Generated by the End User
C_AdharCard	Varchar (15)	Utf8_unicode_ci	Not Null	Generated by the End User
C_Image	Varchar (15)	Utf8_unicode_ci	Not Null	Generated by the End User

- This Table Shows the Details of the Customer (C_Id, C_Name, C_Password, C_Address, C_Phone, C_DOB, C_AdharCard, C_Image).
- C_Id is Primary key In this Table.
- Value of Primary Key is Auto Increment.

Event : Store the information About Posted Event

- This Table Shows the Details of the Posted Event.

Primary Key : E_Id

TABLE : Event

Foreign Key : C_Id

Field Name	Datatype (Size)	Collection	Constraints	Description
E_Id	Integer (6)	-	Primary Key / Auto Inc.	Auto Generated by the Database
C_Id	Integer (6)	-	Foreign key	Fetch Data From the user table
E_Name	Varchar (20)	Utf8_unicode_ci	Not Null	Generated by the End User
E_Members	Varchar (5)	Utf8_unicode_ci	Not Null	Generated by the End User
E_Work	Varchar (200)	Utf8_unicode_ci	Not Null	Generated by the End User
E_Payment	Integer (5)	-	Not Null	Generated by the End User
E_Location	Varchar (100)	Utf8_unicode_ci	Not Null	Generated by the End User
E_MapLocation	Varchar (100)	Utf8_unicode_ci	Not Null	Generated by the End User
E_Date	Varchar (10)	Utf8_unicode_ci	Not Null	Generated by the End User
E_FoodPetrol	Varchar (50)	Utf8_unicode_ci	Not Null	Generated by the End User

E_OtherInfo	Varchar (200)	Utf8_unicode_ci	Not Null	Generated by the End User
-------------	---------------	-----------------	----------	---------------------------

- **This Table Shows the Details of the Customer (E_Id, C_Id,E_Name, E_Member, E_work, E_Payment, E_Location, E_MapLocation, E_Date, E_FoodPetrol, E_OtherInfo).**
- **E_Id is Primary key and C_Id is Foreign Key in this Table.**
- **Value of Primary Key is Auto Increment.**
- **Foreign Key Fetch the Data From User Table.**

Joined_Worker : Store the information About Joined Worker

- This Table Shows the Details of the Workers Who Apply to Join the Event.

Primary Key : E_Id

TABLE : Joined_Worker

Foreign Key : C_Id,E_Id

Field Name	Datatype (Size)	Collection	Constraints	Description
R_Id	Integer (6)	-	Primary Key / Auto Inc.	Auto Generated by the Database
C_Id	Integer (6)	-	Foreign key	Fetch Data From the user table
E_Id	Integer (6)	-	Foreign key	Fetch Data From the Event table
R_Date	Varchar (10)	Utf8_unicode_ci	Not Null	Date is Auto Generated When User Put Item in the Cart.
R_Selected	tinint (1)	-	Not Null	Generated by the End User

- This Table Shows the Details of the Customer (R_Id, C_Id,E_Id, R_Date, R_Selected).
- R_Id is Primary key and C_Id & E_Id is Foreign Key in this Table.
- Value of Primary Key is Auto Increment.
- Foreign Key Fetch the Data From User Table and Event Table.

Feedback : Store the information About Feedback

- This Table Shows the All the Feedback From the User and New Suggestion From Admin.

Primary Key : E_Id

TABLE : Joined_Worker

Foreign Key : C_Id,E_Id

Field Name	Datatype (Size)	Collection	Constraints	Description
F_Id	Integer (6)	-	Primary Key / Auto Inc.	Auto Generated by the Database
C_Id	Integer (6)	-	Foreign key	Fetch Data From the user table
E_Id	Integer (6)	-	Foreign key	Fetch Data From the Event table
F_Date	Date	-	Not Null	Date is Auto Genereted When User Put Item in the Cart.
F_Review	Varchar (200)	Utf8_unicode_ci	Not Null	Generated by the End User

- This Table Shows the Details of the Customer (F_Id, C_Id,E_Id, F_Date, R_Review).
- F_Id is Primary key and C_Id & E_Id is Foreign Key in this Table.
- Value of Primary Key is Auto Increment.
- Foreign Key Fetch the Data From User Table and Event Table.

Payment : Store the information About Payments.

- This Table Shows the All the Payments.

Primary Key : P_Id

TABLE : Payments

Foreign Key : C_Id,E_Id

Field Name	Datatype (Size)	Collection	Constraints	Description
P_Id	Integer (6)	-	Primary Key / Auto Inc.	Auto Generated by the Database
C_Id	Integer (6)	-	Foreign key	Fetch Data From the user table
E_Id	Integer (6)	-	Foreign key	Fetch Data From the Event table
P_Date	Date	-	Not Null	Date is Auto Generated When User Put Item in the Cart.
P_Amount	Integer (10)	-	Not Null	Generated by the End User

- This Table Shows the Details of the Customer (P_Id, C_Id,E_Id, P_Date, P_Amount).
- P_Id is Primary key and C_Id & E_Id is Foreign Key in this Table.
- Value of Primary Key is Auto Increment.
- Foreign Key Fetch the Data From User Table and Event Table.

References Link

- [1] www.draw.io
- [2] www.tutorialspoint.com
- [3] www.w3schools.com
- [4] www.visio.com
- [5] www.stackoverflow.com

Thank You...