Assignment no 6

Aim:
Prepare and implement a sequence model.

Problem Statement:

- Prepare Sequence Model
- Identify at least 5 major scenarios (sequence flow) for your system. Draw Sequence Diagram for every scenario by using advanced notations using UML2.0
- Implement these scenarios by taking reference of design model implementation using suitable object-oriented language.

Objective:

- To study the use of communication.
- Draw sequence diagram
- To implement a sequence diagram.

Theory:

Sequence Diagrams
A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction.
A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects.

Interaction frames
In sequence diagrams and communication diagrams, an interaction frame provides a context or boundary to the diagram in which you create diagram elements, such as lifelines or messages, and in which you observe behavior.
The frame and its contents represent an interaction in a sequence diagram or communication diagram. The heading label of the interaction frame is the name of the interaction that the diagram represents. In sequence diagrams, a frame can

represent combined fragments, which represent scenario constructs, and interaction uses, which represent an interaction within an interaction.

Lifelines in UML diagrams

In UML diagrams, such as sequence or communication diagrams, lifelines represent the objects that participate in an interaction. For example, in a banking scenario, lifelines can represent objects such as a bank system or customer. Each instance in an interaction is represented by a lifeline.

Messages in UML diagrams

A message is an element in a Unified Modeling Language (UML) diagram that defines a specific kind of communication between instances in an interaction. A message conveys information from one instance, which is represented by a lifeline, to another instance in an interaction.

Types of messages

A message specifies a sender and receiver, and defines the kind of communication that occurs between lifelines. For example, a communication can invoke, or call, an operation by using a synchronous call message or asynchronous call message, can raise a signal using an asynchronous signal, and can create or destroy a participant.

You can use the five types of messages that are listed in the following table to show the communication between lifelines in an interaction.

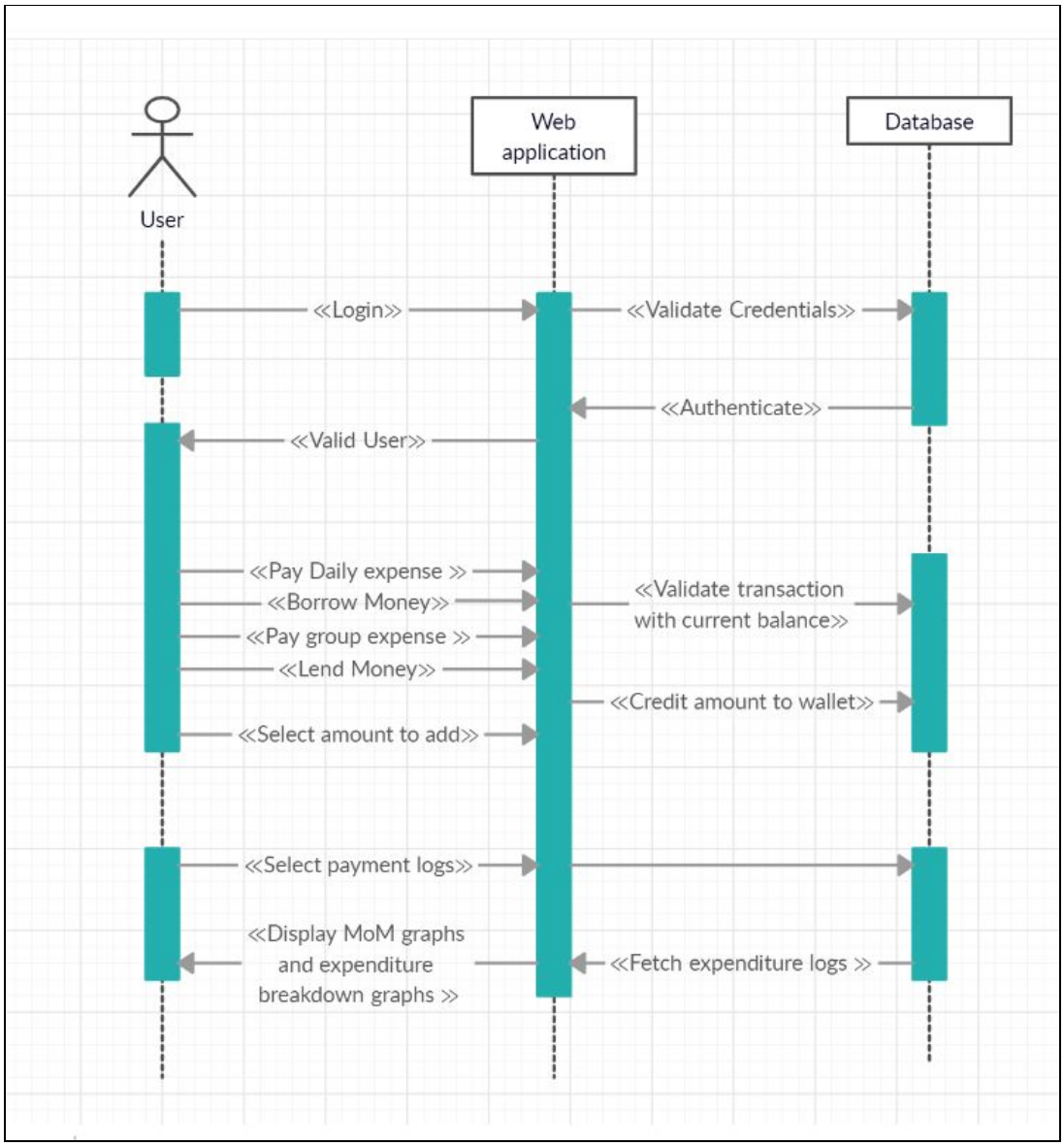
Message type	Description
Create	A create message represents the creation of an instance in an interaction. The create message is represented by the keyword «create». The target lifeline begins at the point of the create message.
Destroy	A destroy message represents the destruction of an instance in an interaction. The destroy message is represented by the keyword «destroy». The target lifeline ends at the point of the destroy message, and is denoted by an X.

Message type	Description
Synchronous call	Synchronous calls, which are Synchronous calls, which have a send and a receive message. A message is sent from the source lifeline to the target lifeline. The source lifeline is blocked from other operations until it receives a response from the target lifeline.
Asynchronous call	Asynchronous calls, which are associated with operations, typically have only a send message, but can also have a reply message. In contrast to a synchronous message, the source lifeline is not blocked from receiving or sending other messages. You can also move the send and receive points individually to delay the time between the send and receive events. You might choose to do this if a response is not time sensitive or order sensitive.
Asynchronous signal	Asynchronous signal messages are associated with signals. A signal differs from a message because no operation is associated with the signal. A signal can represent an interrupt condition or error condition. To specify a signal, you create an asynchronous call message and change the type in the message properties view.
Lost and found	A lost message is a message that has a known sender but the receiver is not known. A found message is a message that does not have a known sender but has a receiver.

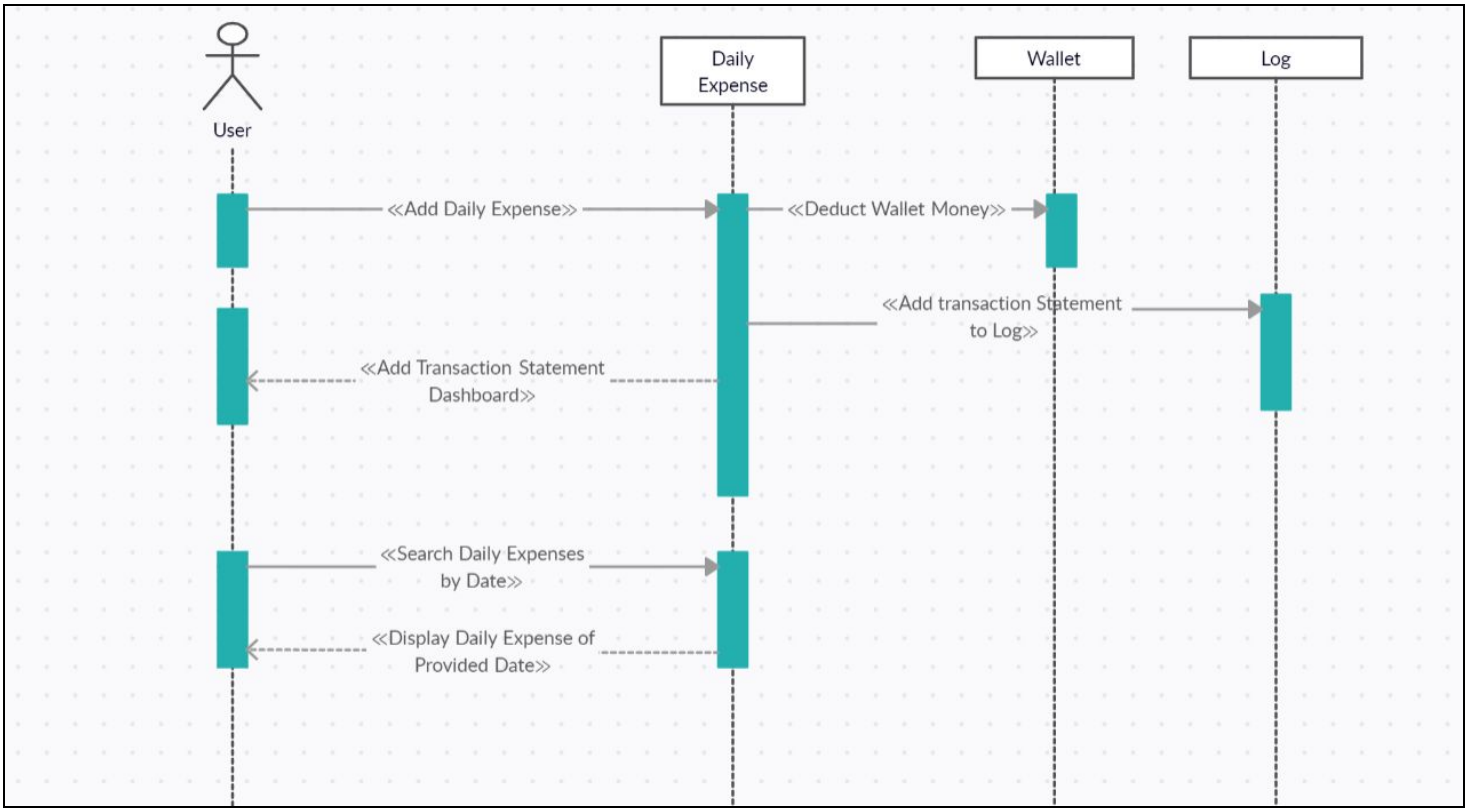
An asynchronous message is the only message type for which you can individually move the sending and receiving points. You can move the points of an asynchronous message to manipulate the time delay between the sending event and the receiving event; the result is called a skewed message. You can create an asynchronous message with or without a behavior execution specification.

A self-directed message is a message that is sent from the source lifeline to itself. A self-directed message could be a recursive call or a call to another operation or signal that belongs to the same object.
The message that the source lifeline sends to the target lifeline represents an operation or a signal that the target lifeline implements. You can name and order messages. The appearance of the line or arrowhead reflects the properties of the message. The following table shows the graphics that represent messages in UML diagrams.

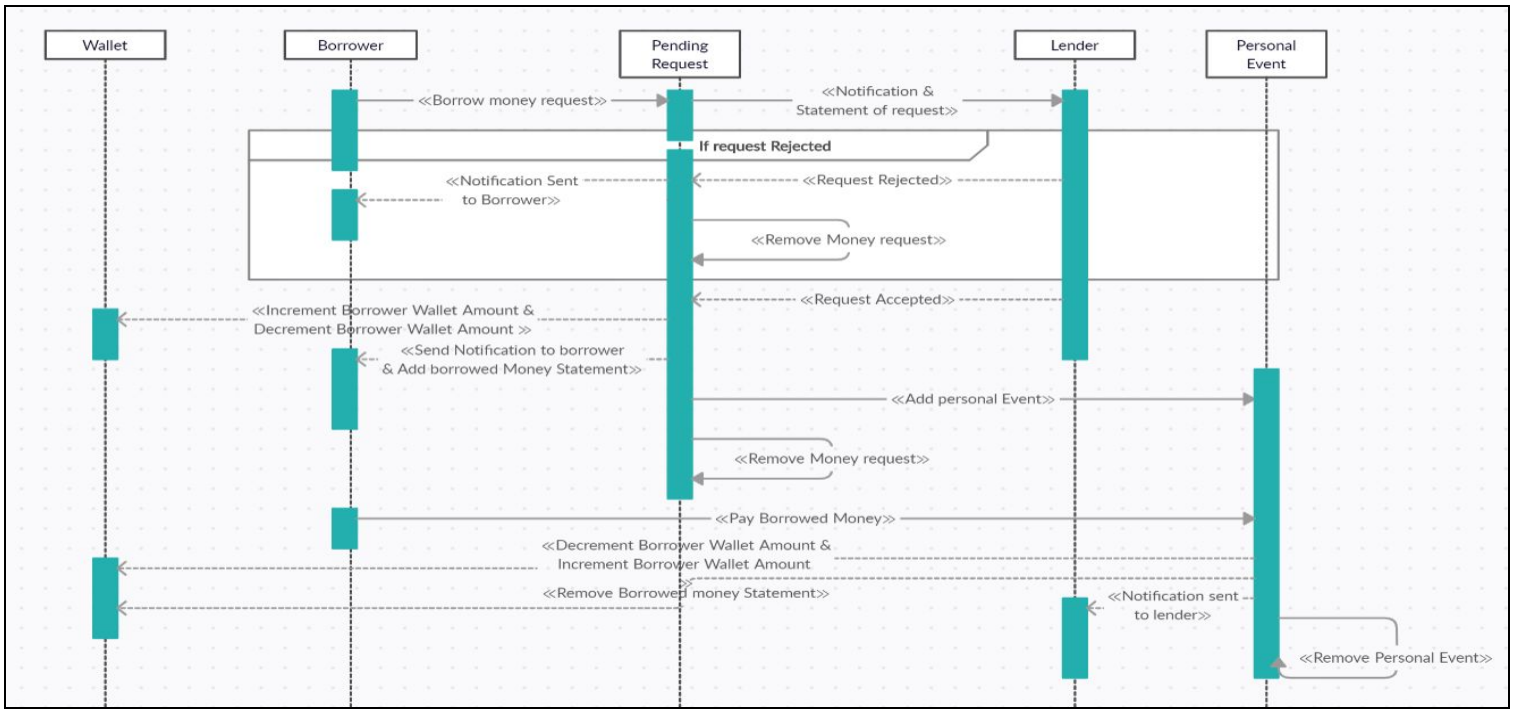
Sequence diagram:



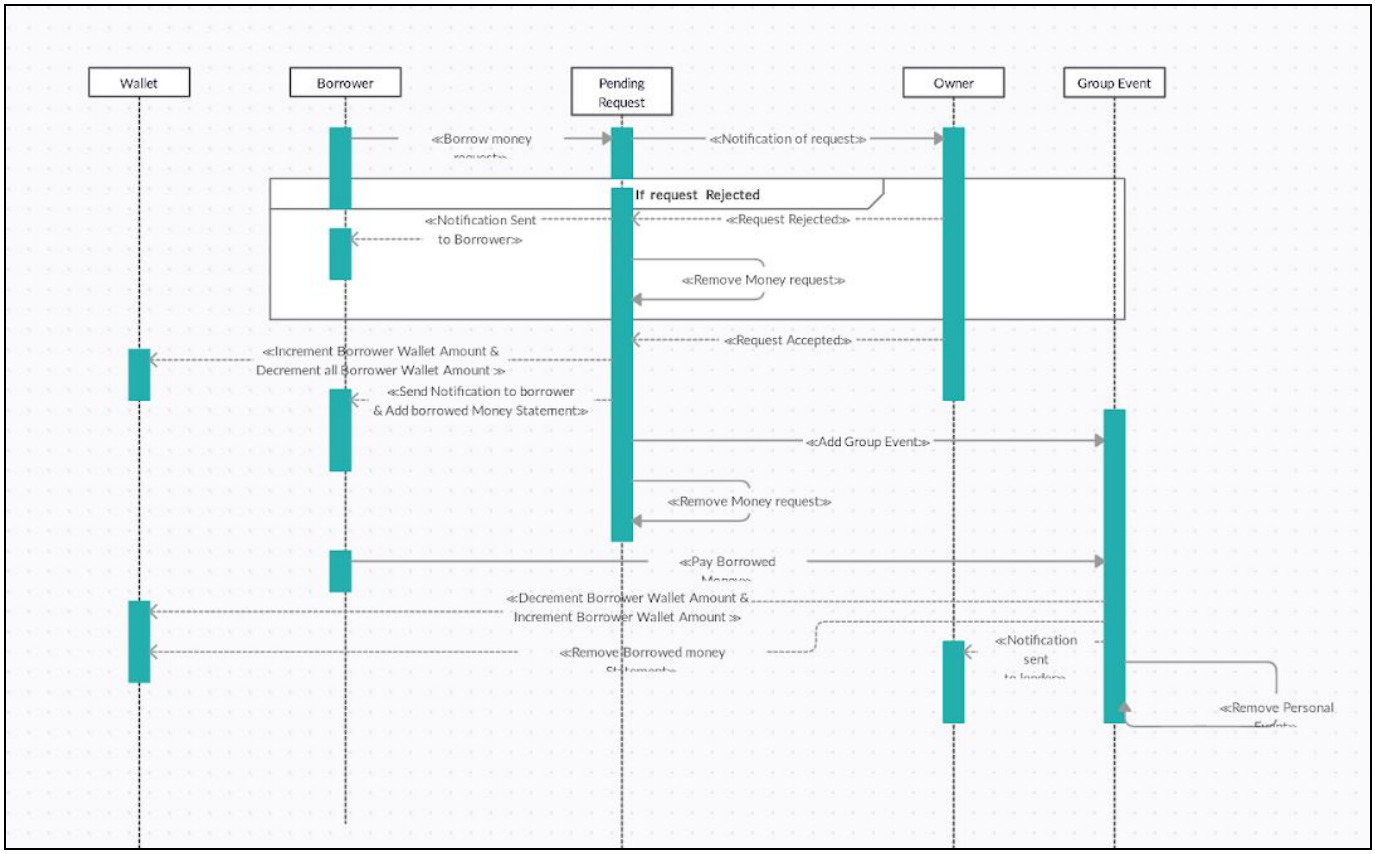
Daily Expense :-



Personal Expense :-



Group Event :-



<u>Conclusion</u>:

We prepared and implemented a sequence diagram successfully.