## Introduction

Software design is procedure to renovate user requirements into certain suitable form, that helps programmer to implement and to do coding. Once the necessities have been composed and analyzed. It is essential to identify in detail how the system will be constructed to perform necessary tasks. Design phase is determined on the requirements, system creation, and the interface creation.

Design has its own importance such as modularity, maintainability, performance, portability, usability, even in deployment. Design must deliver information about what the software deliverables are, and where they be sited in deployment. This intelligence is very very important for our customers over and above integration as they need to know what our software is all about.

The main purposes of software design are services and products, user interfaces, visual and sensory, technology, environment and so on.

## 

## Behavior modelling

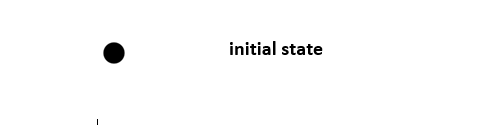
A behavior modelling shows up the collaboration between object to produce some specific system behavior that is stated as use-case. The main task of the project team is to offer a solution that is successfully in use and approaches proper functionality. It is the modelling of in what way the proposed system will interact. The first step in creation of system behavior modeling is requirements gathering. The diagram illustrates hierarchical structure of requirements. Other parts are unified by the suppression association. The specifications are bunched into numerous clusters. Clusters should be friendly, durable, having media capacity.

# Activity diagram

An activity diagram is important diagram to define dynamic aspect of the system. It basically a flowchart characterizes one activity to another activity. It can be portrayed as an operation of the system. Activity diagrams are not only for picturing the dynamic nature of a system, but also for construction of executable system. The only missing part in activity diagram is message part. Moreover, it shows different flows, such as parallel, branched, concurrent, and even single.

* Start point

It denotes beginning of a procedure or else workflow in an activity diagram. It can be used with a note symbol that explains starting point.



* Activity

Activity symbol grants activities that make up a modeled process. The sign includes short description within shape. It is the main building block of an activity diagram.



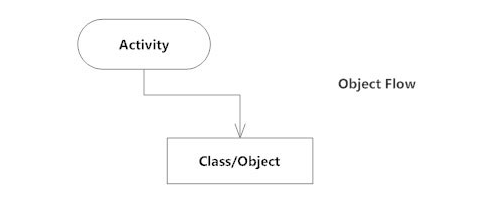
* Action flow

An action flow or connector symbol, demonstrates guiding flow. An arriving arrow starts a step of an activity, after completion of a step the flows continues with outing arrows.



* Object flow

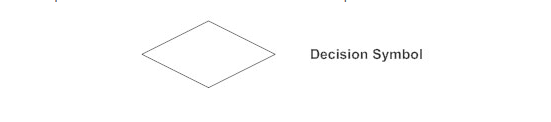
An object flow is a route along with objects or data can be conceded.



* Decisions and branching

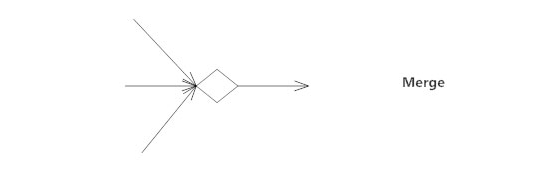
Decision activities are made known to support conditionals in activities. It

Signifies a decision and always has two paths branching out with condition text to permit users to observe options.



* Merge event

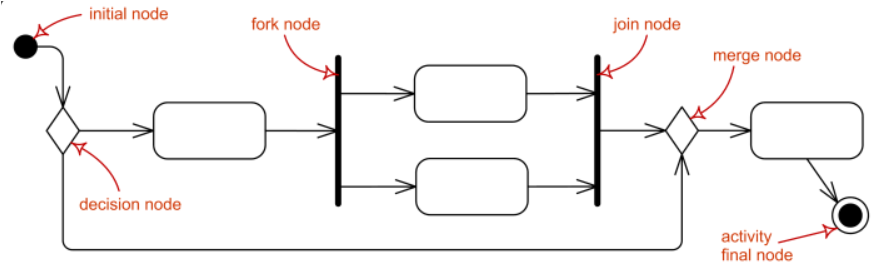
Merge is an association with two packages to specify that content of one package is prolonged by the subjects of another packages.



* Join and fork node

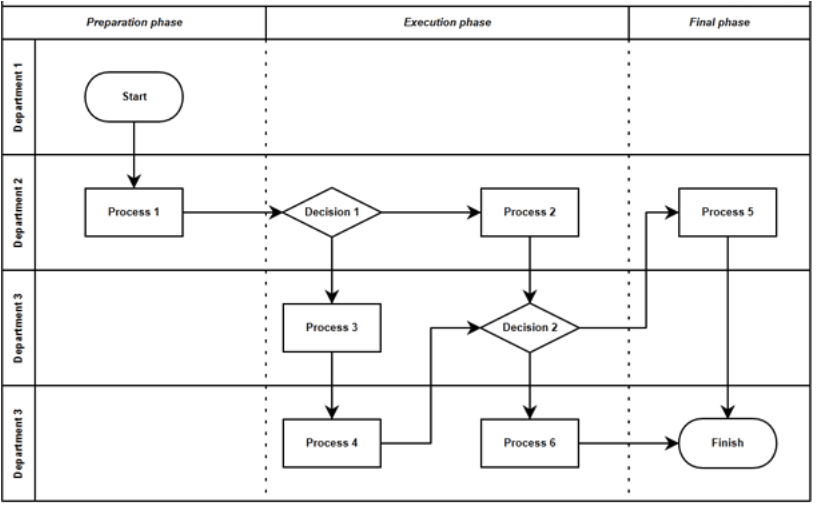
Join symbol or synchronization bar associates two concurrent activities and restores them to a flow where only one activity arises.

Fork symbol splits a single activity flow into two concurrent activities.



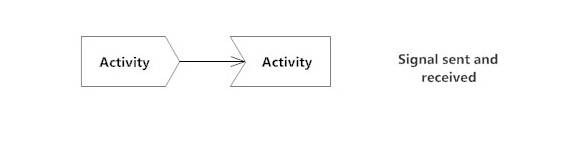
* Swim lanes

A swim lane diagram delivers job sharing and responsibilities for substitute process of a business process. Via metaphor of lanes in a pool, a swim lane diagram delineates clarity and accountability.



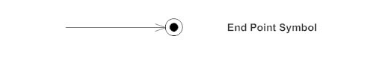
* Send and Receive signals

Send signal symbol demonstrate that a signal is being sent to receiving activities. Receive signal symbol specifies acknowledgement of an event. After event is received the flow originates from this action is completed.

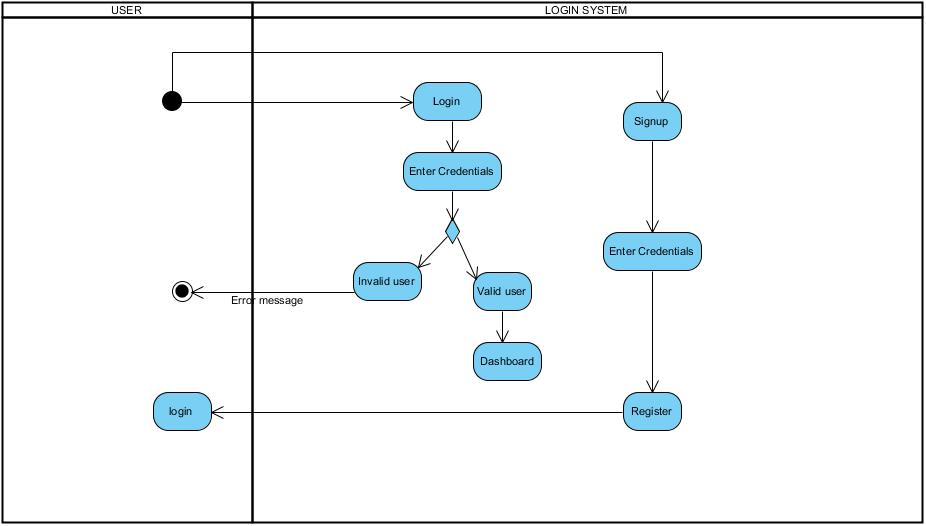


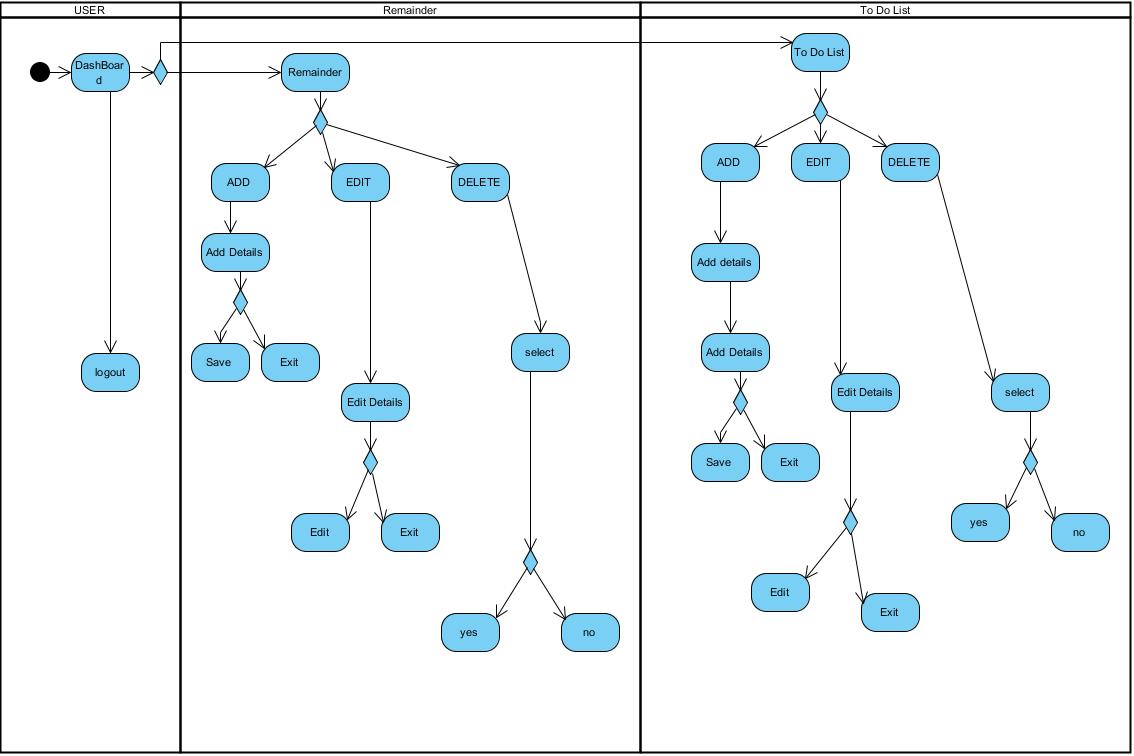
* Final state of end point

End point symbol spots the end state of an activity and signifies the accomplishment of all the flows of process.



The authentic activity diagram is publicized below:





# Sequence diagram

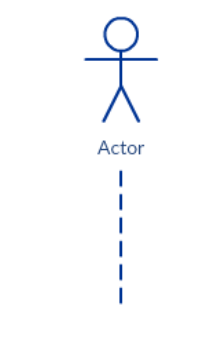
A sequence diagram basically portrays interaction between objects in a sequential order. It is the most common interaction diagram. Similarly, we call sequence diagram event diagrams or event scenarios. These are extensively used by businessmen and software developers to document and understand requirements for new and existing systems.

It is an object collaboration and used primarily for evolution from requirements articulated as use cases to the next and additional official modification.

# Sequence diagram notations

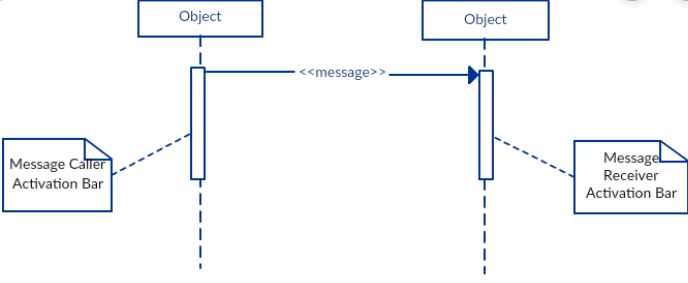
1. Actor

An actor models a type of character played by an entity that interrelates with the subject by exchanging signals and data. A single physical occurrence may play the character of numerous different actors, equally given actor may be played by manifold different instances.



1. Message

A message describes specific communication amongst lifelines of an interaction. It is kind of message that signifies supplication of operation of goal lifeline.



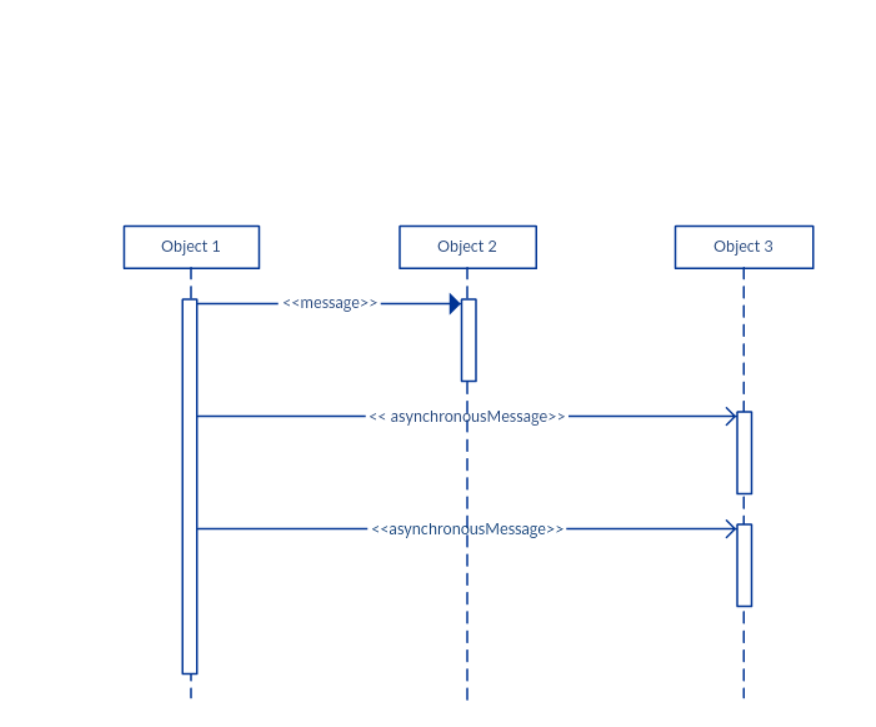
* Synchronous message

A synchronous message involves a prior response of interaction. It is basically sketched using a line with a solid arrowhead pointing from one object to another.



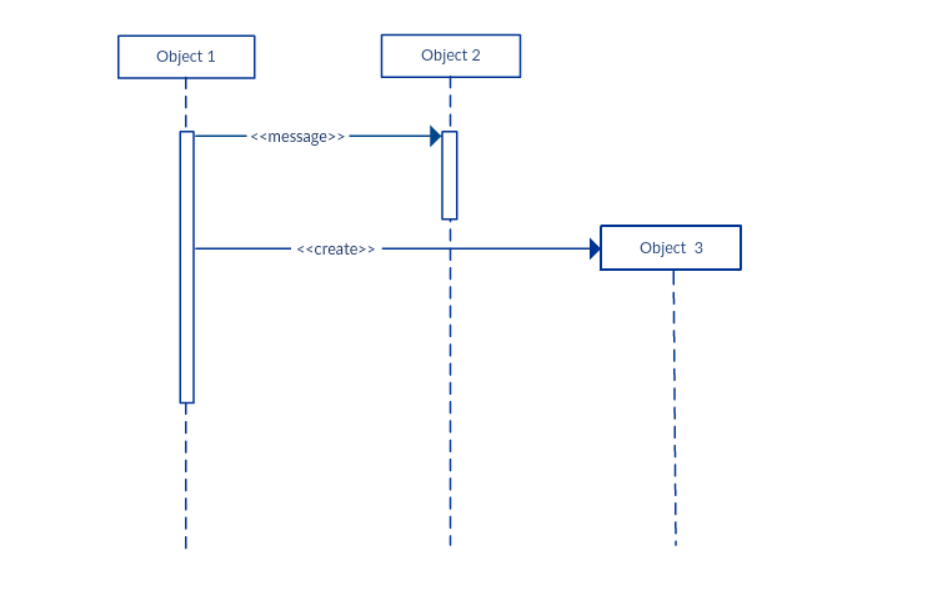
* Asynchronous message

An asynchronous message is sent from an object that will not wait for a response from the receiver before rolling its task.



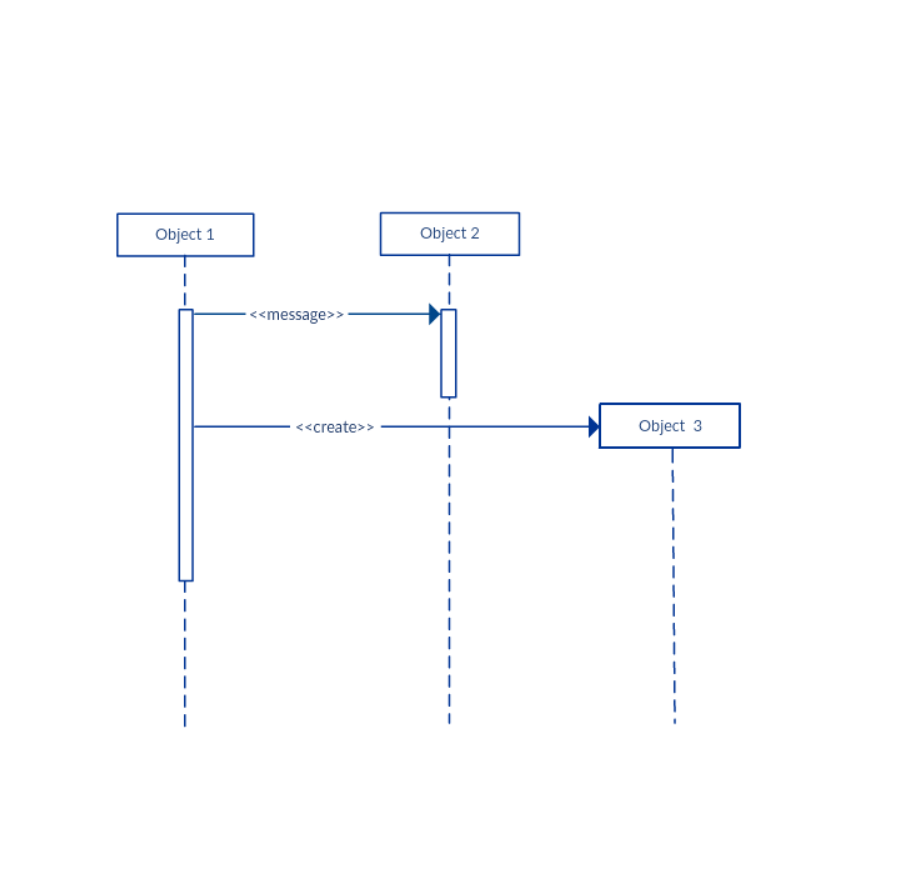
* Destroy message

A destroy message is the destruction of an instance in the interaction. It is styled by the keyword <<destroy>>.



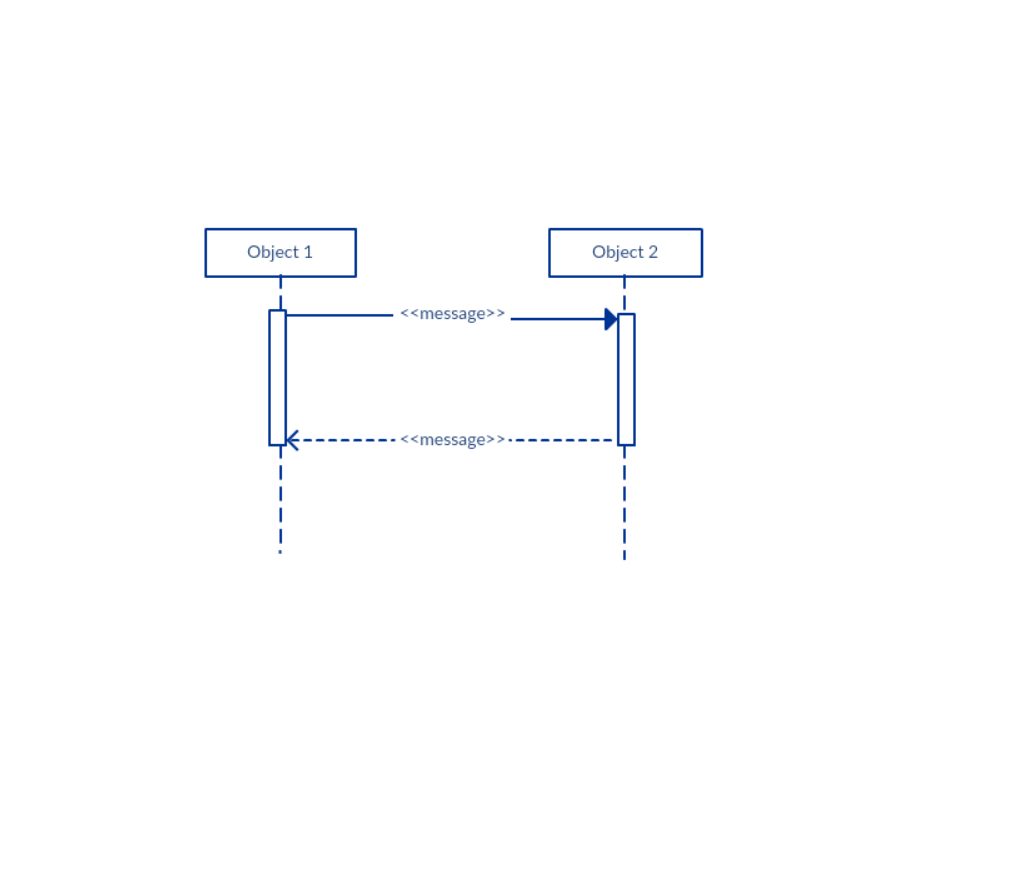
* Create message

A create message signifies the creation of an instance of an interaction. It is represented by keyword <<create>>. For example, in a banking scenario, a bank manager starts a credit check on a client by sending a create message to the server.



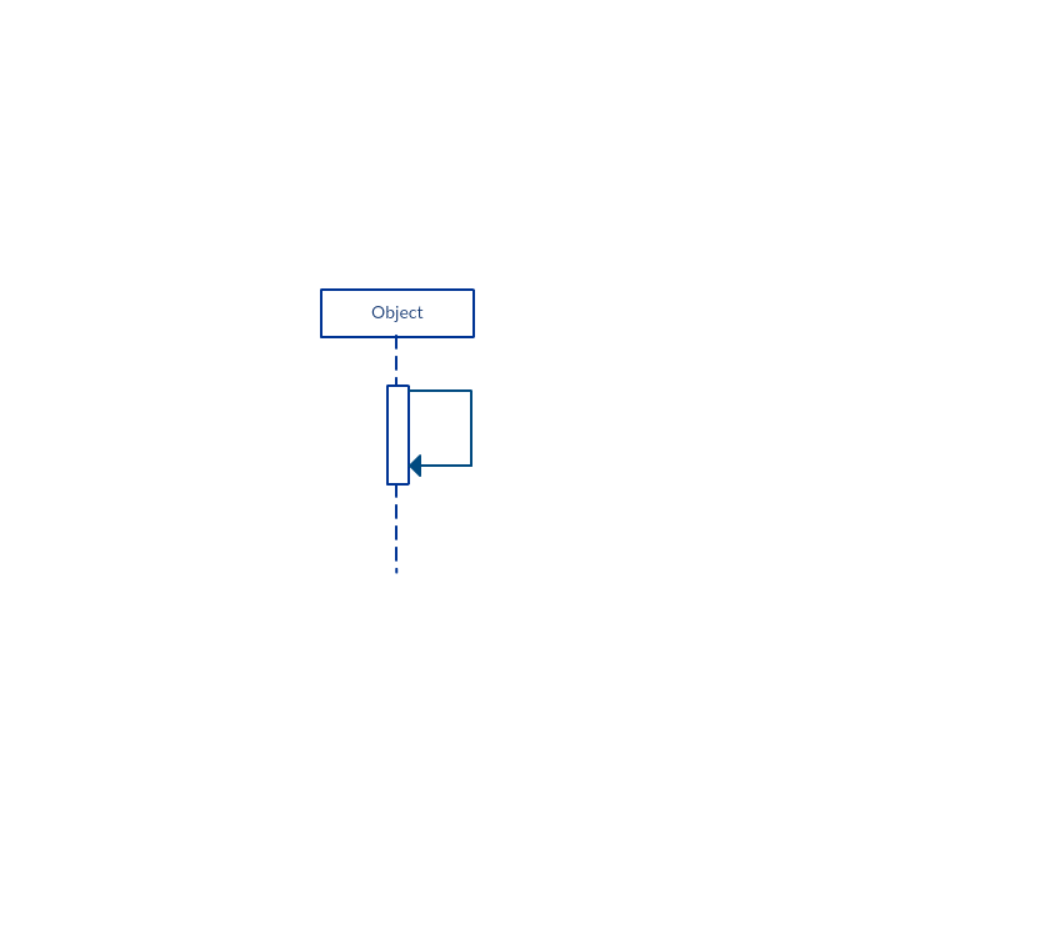
* Return message

Return message outlines a particular statement between lifeline of an interaction. It exemplifies the pass of information return to the caller of a corresponded former message.



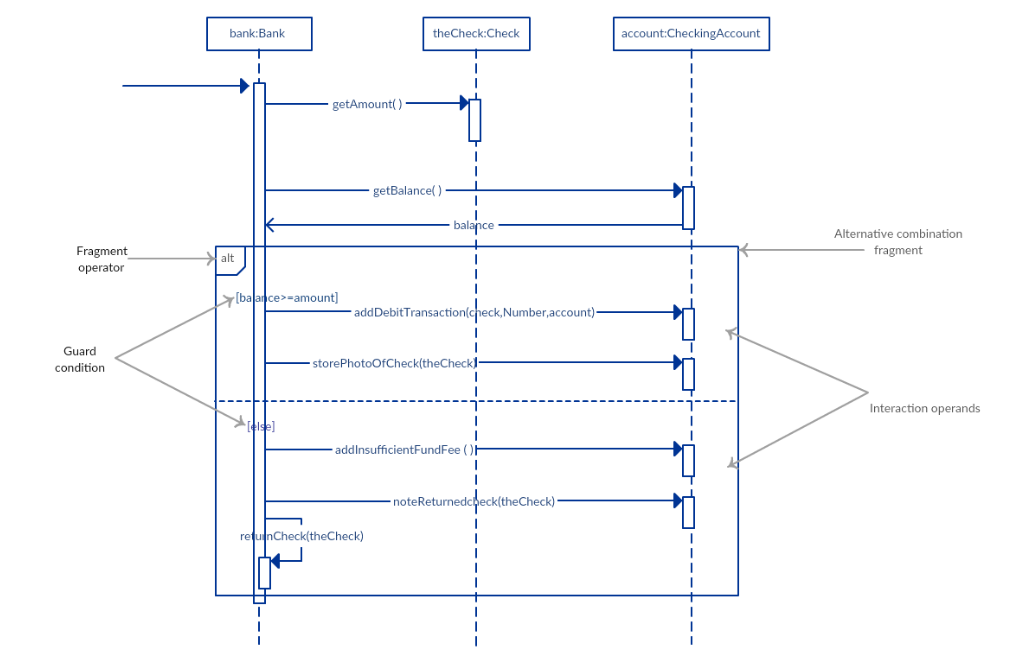
* Reflexive messages

A message outlines a particular communication between lifelines of an interaction. It is type of message that shows the supplication of message of the same lifeline. Its goal points to an initiation on the top of the initiation where the message was invoked from.



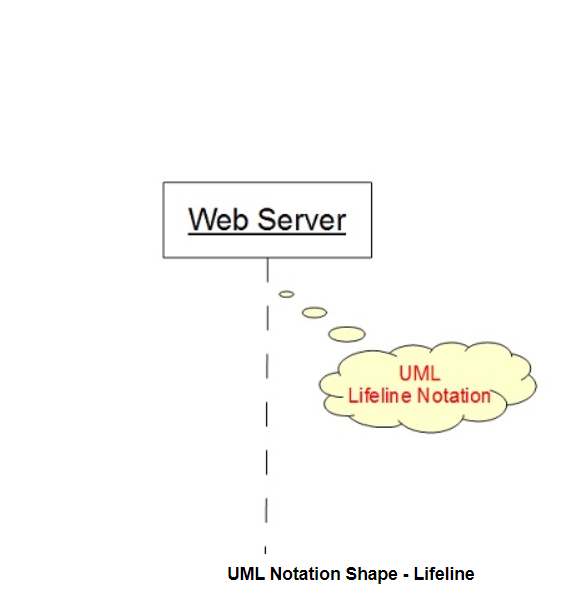
1. Guards

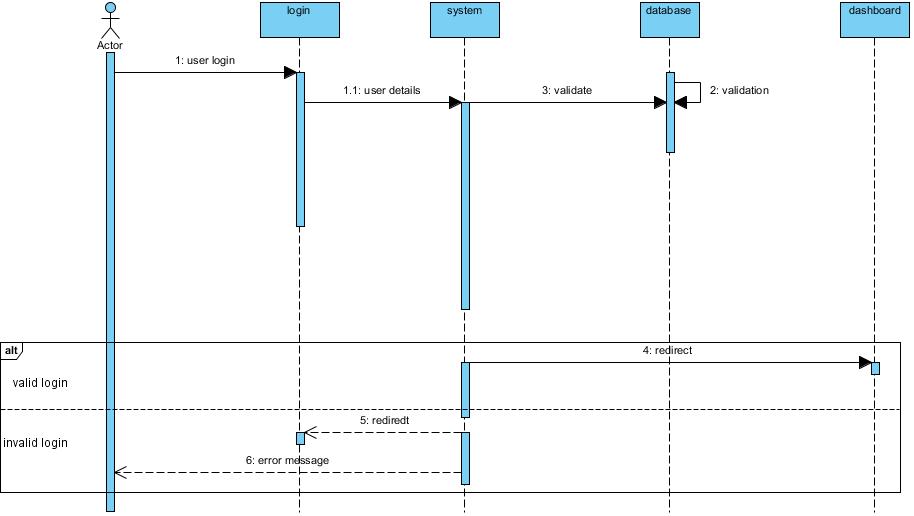
Guards are used when we prerequisite to restrict the flow of messages on the excuse of a condition being met. It plays a significant role in letting software developers know the constraint attached to a system or a certain process. For example in order to able to withdraw cash having a balance greater than zero.



1. Lifeline

A lifeline is an element that portrays an individual participant in a sequence diagram. Fundamentally each instance in a sequence diagram is graphed by a lifeline. Lifeline elements are situated at the top left in a sequence diagram. It is an individual participant in the interaction.





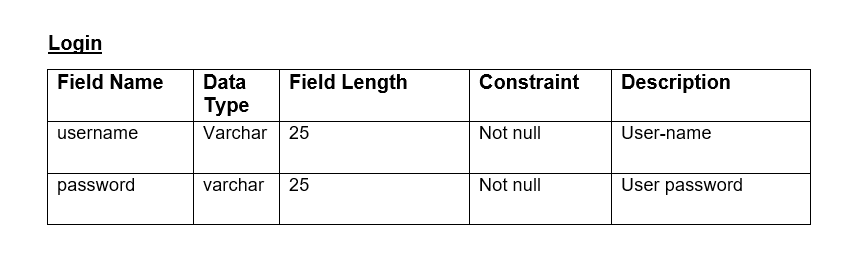
## Database modelling

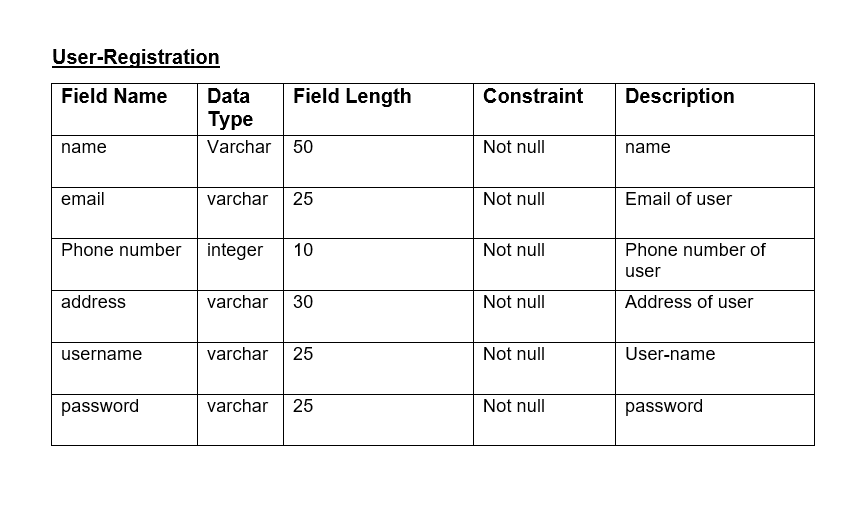
Database modelling is procedure of creating a data model for the data to be stored in a database. It is theoretical representation of data objects, associations between different data object, or of the rules. Database modelling assistances in the visual illustration of data and implements business guidelines, symmetry, submissions, and government policies on data. It guarantees reliability in naming conventions, default values, semantics, security while conforming quality of the data.

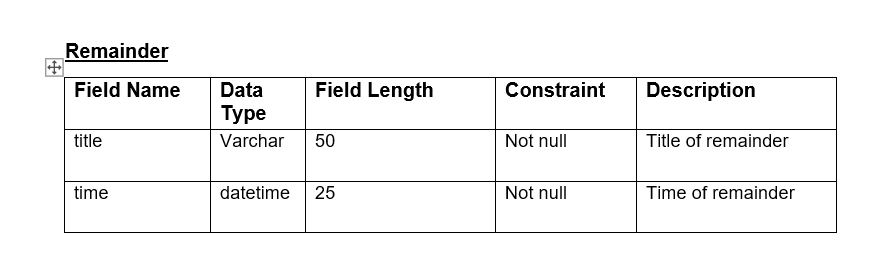
# Data dictionary

Data dictionary stays as metadata. It is storehouse of data about data. It is an assortment of narration of data objects or items in data model. It is also called data definition matrix, delivers detailed information about the business data or standard definition of data element, their meanings, and allowable values.

In actual fact, a data dictionary offers a tool that permits you to interconnect business stakeholder requirements in such way of designing a relational database or data structure to achieve requirements.

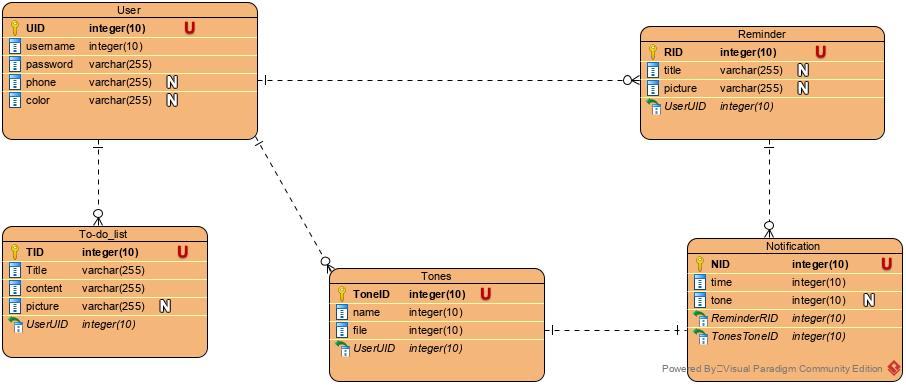






# ER-diagram

The entity relationship diagram is a high-level conceptual data model diagram. ER-diagram is established on the idea of real-world entities and the relationship between them. It facilitates us to analyze data obligation systematically to produce a well-designed database. It is GUI representation logical structure of database. It embraces many specialized symbols, and its denotations make the model exceptional.



## UI Modelling

User interface is a front-end application interpretation to which user interrelates so as to use the software. The software improves to be widespread uncertainty its user interface is attractive, modest to use, responsive, consistent, and clear.

The goal of the interface design is to outline the set of interface object and action i.e. control mechanisms that enable user to achieve desired tasks. In the other hand, interface construction and implementation begin with prototype that assists practice scenario to be gauged.

# Prototyping

Prototyping is a flow of a product that permits us to discover our ideas and demonstrate the intention behind a feature or general concept to user before capitalizing money and time into development. It is an indispensable step in the product development process.

The concept clarifies the merits and demerits of different methodologies for prototyping and defines the desperate factors and procedures needed to implement prototyping successfully.