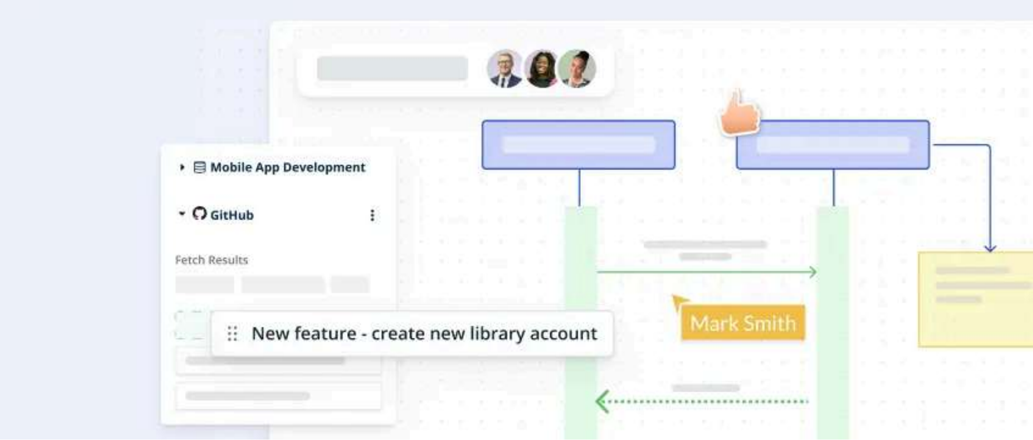
Sequence Diagram Tutorial – Complete Guide with Examples

Updated on: 12 December 2022 | 13 min read



This sequence diagram tutorial is to help you understand sequence diagrams better; to explain everything you need to know, frc the common mistakes you should avoid when drawing one.

There are 3 types of Interaction diagrams; Sequence diagrams, communication diagrams, and timing diagrams. These diagrams between parts within a system. Among the three, sequence diagrams are preferred by both developers and readers alike for the

In this sequence diagram tutorial you will learn about;

What is a Sequence Diagram?

Sequence diagrams, commonly used by developers, model the interactions between objects in a single use case. They illustrate interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is exec

In simpler words, a sequence diagram shows **how different parts of a system work in a 'sequence' to get something done**.

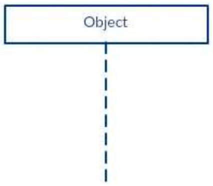
Sequence diagrams are commonly used in software development to illustrate the behavior of a system or to help developers de: They can be used to model both simple and complex interactions between objects, making them a useful tool for software archi

Sequence Diagram Notations

A sequence diagram is structured in such a way that it represents a timeline that begins at the top and descends gradually to m object has a column and the messages exchanged between them are represented by arrows.

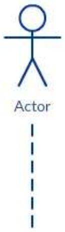
A Quick Overview of the Various Parts of a Sequence Diagram

Lifeline Notation

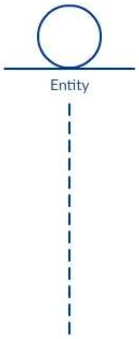


A sequence diagram is made up of several of these lifeline notations that should be arranged horizontally across the top of the c
overlap each other. They represent the different objects or parts that interact with each other in the system during the sequenc

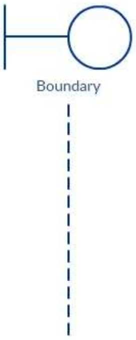
A lifeline notation with an actor element symbol is used when the particular sequence diagram is owned by a use case.



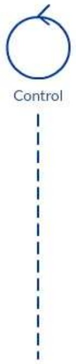
A lifeline with an entity element represents system data. For example, in a customer service application, the Customer entity wo
customer.



A lifeline with a boundary element indicates a system boundary/ software element in a system; for example, user interface scree
users interact with, are boundaries.



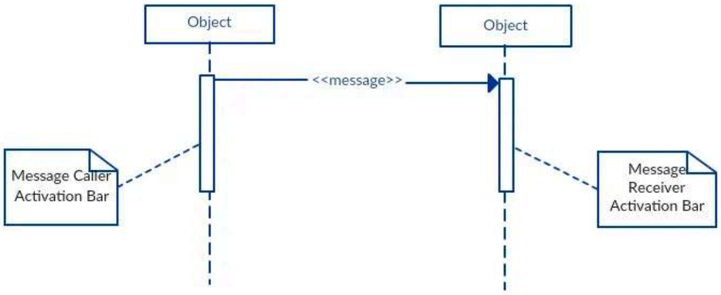
And a lifeline with a control element indicates a controlling entity or manager. It organizes and schedules the interactions betwe
serves as the mediator between them.



Activation Bars

The activation bar is the box placed on the lifeline. It is used to indicate that an object is active (or instantiated) during an intera of the rectangle indicates the duration of the objects staying active.

In a sequence diagram, an interaction between two objects occurs when one object sends a message to another. The use of the Message Caller (the object that sends the message) and the Message Receiver (the object that receives the message) indicate during the exchange of the message.


Object
Object
<<message>>
Message Caller Activation Bar
Message Receiver Activation Bar

Message Arrows

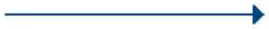
An arrow from the Message Caller to the Message Receiver specifies a message in a sequence diagram.   A message can flow in left, or back to the Message Caller itself. While you can describe the message being sent from one object to the other on the arr indicate the type of message being sent or received.

The message arrow comes with a description, which is known as a message signature, on it. The format for this message signat message_name are optional.

attribute = message_name (arguments): return_type

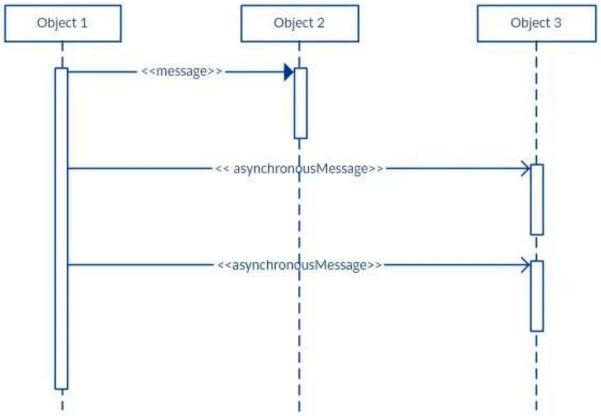
- ***Synchronous message***

As shown in the activation bars example, a synchronous message is used when the sender waits for the receiver to process the with another message. The arrowhead used to indicate this type of message is a solid one, like the one below.



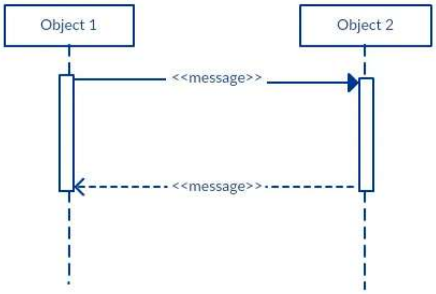
- ***Asynchronous message***

An asynchronous message is used when the message caller does not wait for the receiver to process the message and return b objects within the system. The arrowhead used to show this type of message is a line arrow as shown in the example below.



- ***Return message***

A return message is used to indicate that the message receiver is done processing the message and is returning control over to
are optional notation pieces, for an activation bar that is triggered by a synchronous message always implies a return message.

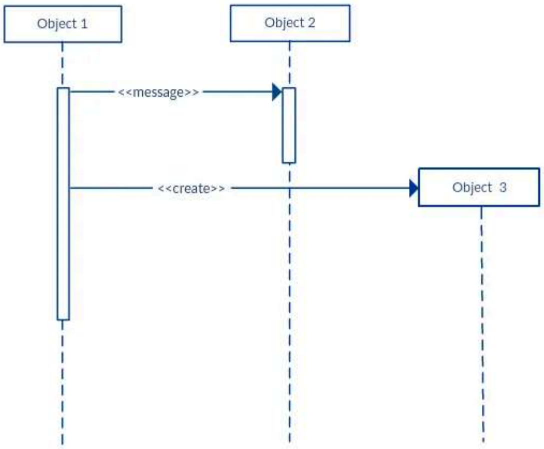


Tip: You can avoid cluttering up your diagrams by minimizing the use of return messages since the return value can be specified

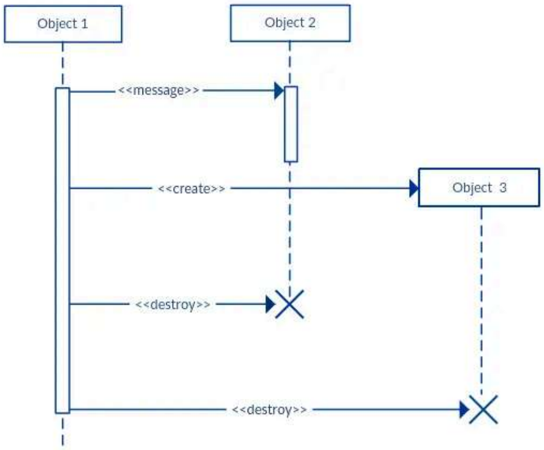
- ***Participant creation message***

Objects do not necessarily live for the entire duration of the sequence of events. Objects or participants can be created accordi

The dropped participant box notation can be used when you need to show that the particular participant did not exist until the c
participant does something immediately after its creation, you should add an activation box right below the participant box.



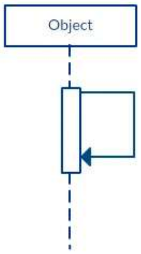
- ***Participant destruction message***



Likewise, participants when no longer needed can also be deleted from a sequence diagram. This is done by adding an 'X' at the participant.

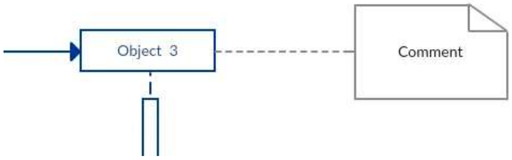
- ***Reflexive message***

When an object sends a message to itself, it is called a reflexive message. It is indicated with a message arrow that starts and er example below.



Comment

UML diagrams generally permit the annotation of comments in all UML diagram types. The comment object is a rectangle with a comment can be linked to the related object with a dashed line.



Note: View Sequence Diagram Best Practices to learn about sequence fragments.

Sequence Diagram Best Practices

- **Manage complex interactions with sequence fragments**

A sequence fragment is represented as a box that frames a section of interactions between objects (as shown in the examples b

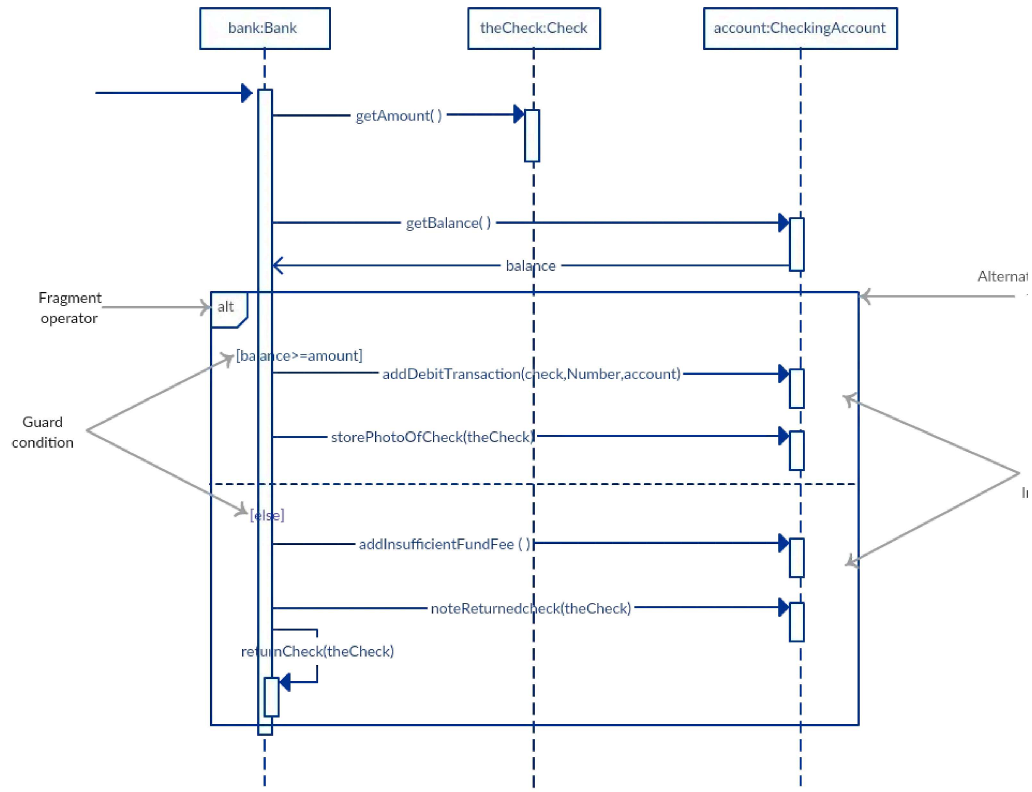
It is used to show complex interactions such as alternative flows and loops in a more structured way. On the top left corner of th fragment operator – specifies what sort of a fragment it is.

Alternatives

The alternative combination fragment is used when a choice needs to be made between two or more message sequences. It mo

The alternative fragment is represented by a large rectangle or a frame; it is specified by mentioning 'alt' inside the frame's name

To show two or more alternatives, the larger rectangle is then divided into what is called interaction operands using a dashed lin example above. Each operand has a guard to test against and it is placed at the top left corner of the operand.



Options

The option combination fragment is used to indicate a sequence that will only occur under a certain condition, otherwise, the se
then” statement.

Similar to the alternative fragment, the option fragment is also represented with a rectangular frame where ‘opt’ is placed inside

Unlike the alternative fragment, an option fragment is not divided into two or more operands. Option’s guard is placed at the top

(Find an example sequence diagram with an option fragment in the Sequence Diagram Templates and Examples section).

Loops

Loop fragment is used to represent a repetitive sequence. Place the words ‘loop’ in the name box and the guard condition near t

In addition to the Boolean test, the guard in a loop fragment can have two other special conditions tested against. These are mir
number] and maximum iterations (written as maxint = [the number]).

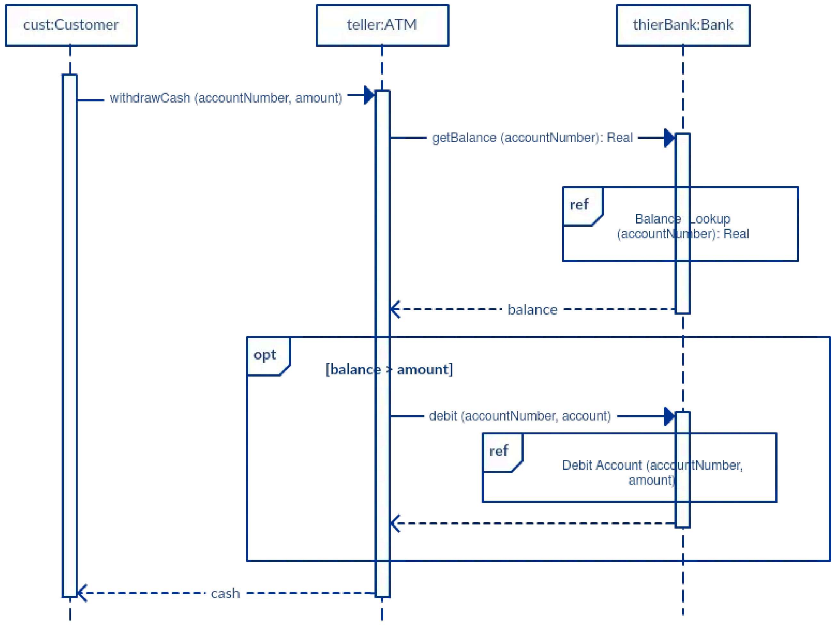
If it is a minimum iterations guard, the loop must execute not less than the number mentioned, and if it is a maximum iterations g
than the number indicated.

(Find an example of a loop fragment below in the sequence diagram templates and example section)

Reference Fragment

You can use the ref fragment to manage the size of large sequence diagrams. It allows you to reuse part of one sequence diagra
reference part of a diagram in another diagram using the ref fragment.

To specify the reference fragment, you have to mention ‘ref’ in the name box of the frame and the name of the sequence diagrar
frame.


cust:Customer
teller:ATM
thierBank:Bank
withdrawCash (accountNumber, amount)
getBalance (accountNumber): Real
ref
Balance Lookup (accountNumber): Real
balance
opt
[balance > amount]
debit (accountNumber, account)
ref
Debit Account (accountNumber, amount)
cash

For more sequence fragments refer to Beyond the Basics of Sequence Diagrams: Part 1,Part 2 and Part 3.

- **Draw smaller sequence diagrams that capture the essence of the use case**

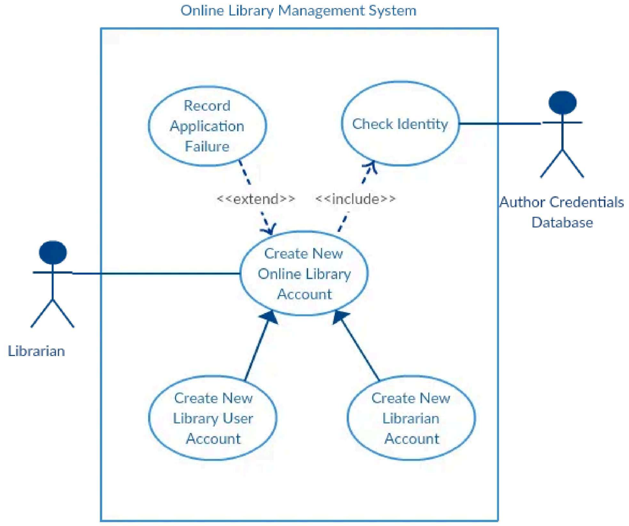
Instead of cluttering your sequence diagram with several objects and groups of messages that will confuse the reader, draw a fe
explain what your system does. Make sure that the diagram fits on a single page and leaves space for explanatory notes too.

Also instead of drawing dozens of sequence diagrams, find out what is common among the scenarios and focus on that. And if t
its own, there's no need to draw a sequence diagram in the first place.

How to Draw a Sequence Diagram

A sequence diagram represents the scenario or flow of events in one single use case. The message flow of the sequence diagra
particular use case.

Then, before you start drawing the sequence diagram or decide what interactions should be included in it, you need to draw the
comprehensive description of what the particular use case does.


Online Library Management System
Record Application Failure
Check Identity
<<extend>>
<<include>>
Author Credentials Database
Create New Online Library Account
Librarian
Create New Library User Account
Create New Librarian Account

From the above use case diagram example of 'Create New Online Library Account', we will focus on the use case named 'Create
sequence diagram example.

Before drawing the sequence diagram, it's necessary to identify the objects or actors that would be involved in creating a new u

- Librarian
- Online Library Management system
- User credentials database
- Email system

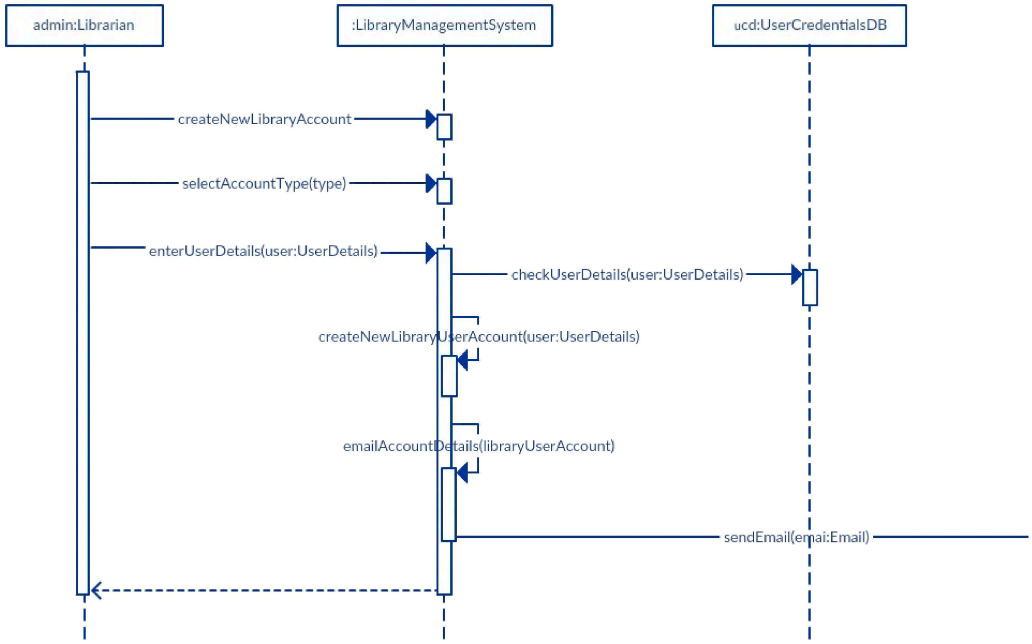
Once you identify the objects, it is then important to write a detailed description of what the use case does. From this descriptic interactions (that should go in the sequence diagram) that would occur between the objects above, once the use case is execut

Here are the steps that occur in the use case named 'Create New Library User Account'.

- The librarian request the system to create a new online library account
- The librarian then selects the library user account type
- The librarian enters the user's details
- The user's details are checked using the user Credentials Database
- The new library user account is created
- A summary of the new account's details is then emailed to the user

From each of these steps, you can easily specify what messages should be exchanged between the objects in the sequence dia and start drawing the sequence diagram.

The sequence diagram below shows how the objects in the online library management system interact with each other to perfoı Account'.


admin:Librarian
:LibraryManagementSystem
ucd:UserCredentialsDB
createNewLibraryAccount
selectAccountType(type)
enterUserDetails(user:UserDetails)
checkUserDetails(user:UserDetails)
createNewLibraryUserAccount(user:UserDetails)
emailAccountDetails(libraryUserAccount)
sendEmail(emai:Email)

Sequence Diagram Common Mistakes

When drawing sequence diagrams, designers tend to make these common mistakes. By avoiding these mistakes you can ensure

- Adding too much detail. This clutters up the diagram and makes it difficult to read.
- Obsolete and out-of-date sequence diagrams that are irrelevant when compared to the interfaces, actual architectures, etc them or modify them.
- Leaving no blank space between the use case text and the message arrow; this makes it difficult for anyone to read the diag
- Not considering the origins of message arrows carefully.

See these common mistakes explained in detail in Sequence Diagram Guide: Common Mistakes to Avoid When Drawing Sequen

Sequence Diagram Templates and Examples

Following are a few sequence diagram examples and templates that are drawn using Creately. Create sequence diagrams online template to open it in the editor.

Sequence Diagram of an Online Exam System

Online Examination - Sequence Diagram Template

Online Examination - Sequence Diagram Template (Click on the template to edit it online)

Sequence Diagram Example of a School Management System

School Management System - Sequence Diagram Template

School Management System - Sequence Diagram Template (Click on the template to edit it online)

Example of an Option Combination Fragment

Example of an option fragment

Example of an option fragment (Click on the template to edit it online)

Example of a Loop Sequence

Loops - Sequence Diagram Example

Loops - Sequence Diagram Example (Click on the template to edit it online)

Sequence Diagram Example of a Card Game

Card Game - Sequence Diagram Template

Card Game - Sequence Diagram Template (Click on the template to edit it online)

Sequence Diagram Example of a Balance Lookup

Balance Lookup - Sequence Diagram Example

Balance Lookup - Sequence Diagram Example (Click on the template to edit it online)

Sequence Diagram Example of an Online Movie Ticket Booking System

Online Movie Ticket Booking - Sequence Diagram Template

Online Movie Ticket Booking - Sequence Diagram Template (Click on the template to edit it online)

Here are some more sequence diagram templates and examples that you can edit right away.

Feedback on the Sequence Diagram Guide

This sequence diagram tutorial covers everything you need to know on sequence diagrams and drawing them. If you have any s sequence diagram tutorial, feel free to leave a comment.

More Diagram Tutorials

- Business Process Modeling Tutorial (BPM Guide Explaining Features)
- Ultimate Flowchart Guide (Complete Flowchart Tutorial with Examples)
- Use Case Diagram Tutorial (Guide with Examples)

FAQs on Sequence Diagrams

What are the benefits of sequence diagrams?

What are UML diagrams?

What distinguish sequence diagrams from communication diagrams?

sequence diagram | sequence diagram tutorial | sequence diagram templates

Author

Amanda Athuraliya

Amanda Athuraliya

Communications Specialist

Amanda Athuraliya is the communication specialist/content writer at Creately, online diagramming and collaboration tool. She is an avid reader, a budding writer and a passionate researcher who loves to write about all kinds of topics.

linl icc

View all posts by Amanda Athuraliya →

Leave a Comment

Type your message here

Name

Email

Related Tools and Resources

Step-by-Step Visual Guide to Mobile App Planning

Step-by-Step Visual Guide to Mobile App Planning

How to Make a User Flow Diagram

How to Make a User Flow Diagram

What Is Product Management? The Ultimate Beginner's Guide

What Is Product Management? The Ultimate Beginner's Guide

15 Types of Product Managers: Skills, Roles, and Career Paths

15 Types of Product Managers: Skills, Roles, and Career Paths

Join over thousands of organizations that use Creately to brainstorm, plan, analyze, and execute their projects successfully.

Get Started Here

Join Creately

Creately

Enterprise Grade Security

Creately is a leader in Diagramming, Visual Collaboration and Collaborative Whiteboard on G2

Creately reviews on Capterra
Read Creately reviews on G2

Company

About Us
Press Kit
Careers
Partner Program
Affiliate Program

Contact

Contact Sales
Contact Support

Resources

Blog
Help Center
Templates
Product Updates
Events

Solutions

HR Planning Software
IT and Operations
Marketing and Sales
Project Management
Software Teams
Production and Manufacturing
Creately for Education
AI Diagram Generator
On-Prem Services

Integrations

Confluence Cloud
Microsoft Teams
Slack

Use Cases

Visual Collaboration
Strategy & Planning
Business Process Management
Creately for Enterprise

Home

Resources

Plans

Features

Templates

Blog

Affiliate Program

Enterprise

Contact Sales

Facebbok X LinkedIn Pinterest YouTube Instagram

Terms of Service | Privacy Policy | Security Policy