# Getting Ready: The Facebook System

Understand the Facebook problem and learn the questions to further simplify this problem.

**We'll cover the following**

* [Problem definition](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Problem-definition)
* [Expectations from the interviewee](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Expectations-from-the-interviewee)
  + [Discoverability](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Discoverability)
  + [Friends and following](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Friends-and-following)
  + [Groups and pages](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Groups-and-pages)
  + [Privacy](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Privacy)
  + [Alerts](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Alerts)
* [Design approach](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Design-approach)
* [Design pattern](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Design-pattern)

## Problem definition

**Facebook** is a social media platform for users to connect with people and engage with different types of media and content. Users can connect with other users by sending friend requests or using the direct messaging feature. Each user has a profile where they can create posts to share with their friends. Facebook also allows its users to create pages regarding a topic of interest and groups to form a community of similar people. Facebook generates a personalized feed for its users based on their friends, liked pages, groups, and the content they engage with to ensure the best experience.

The Facebook home page of a user

## Expectations from the interviewee

Facebook provides multiple functionalities to its users. It is important to narrow down what components you will include in your Facebook design. The section below provides an overview of some of the main expectations that the interviewer will want to hear you discuss in more detail during the interview.

### Discoverability

It is important to develop a better understanding of how Facebook’s discoverability works. The interviewer would expect you to ask the questions listed below:

* How are users able to search other users’ profiles?
* Can users search other users using their phone numbers and email addresses?

### Friends and following

The “Add friend” feature of Facebook is its most renowned feature. Make sure to ask the interviewer the questions provided below:

* How are users able to add other users as friends?
* How are users able to follow or unfollow other users without becoming a direct connection?

### Groups and pages

Groups and pages on Facebook create a space for like-minded individuals. Make sure to define your requirements. You may ask the interviewer the questions listed below:

* How are users able to create both groups and pages in addition to their user profiles?
* What are the criteria for joining a group? Can a user join a group if they are not friends with the creator of the group?

### Privacy

You can ask the following questions to learn about the privacy feature in your design:

* How will the system handle privacy lists with certain content to be displayed only to specified connections?
* How do users set the privacy of groups? How does this change the way members and non-members interact with the group?

### Alerts

Notifications and alerts allow users to stay updated with the activity in their circle. Therefore, you may want to understand how alerts will work in your system. You may ask the questions listed below:

* How will users be notified of pre-selected events?
* How much control do users have in choosing what notifications they can receive?

## Design approach

We’ll design Facebook using the bottom-up design approach. For this purpose, we will follow the steps below:

* Identify and design the smallest components first, like, a post and comment.
* Use these small components to design bigger components, for example, pages, groups, and profiles.
* Repeat the steps above until we design the complete Facebook platform.

## Design pattern

It is always a good practice to discuss the design patterns that Facebook falls under, during the interview. Stating the design patterns will give the interviewer a positive impression and shows that the interviewee is well-versed in the advanced concepts of object-oriented design.

The following design pattern can be used to design Facebook:

* Observer design pattern

Let’s explore the requirements of Facebook in the next lesson.

# Requirements for Facebook

Learn about all requirements of Facebook.

**We'll cover the following**

* [Requirements collection](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Requirements-collection)

In this lesson, we’ll list the requirements of Facebook. This is a very crucial step as requirements define the scope of a problem, so getting them right from the interviewer and understanding them well will make the design of the rest of the system smooth and easy.

We’ll use the notational convention to identify each requirement with a unique label "Rn", where "R" is short for Requirement and "n" is a natural number.

## Requirements collection

The requirements for Facebook are defined below:

**R1:** Users should be able to set the privacy of their profile page. They should also be able to create their profile page and add information such as work experience, education, and place of living.

**R2:** Users of our system should be able to search for groups, pages, and other users.

**R3:** Users should be able to write a new post and set its privacy.

**R4:** Users should be able to send as well as respond to the friend requests of other users. Users should also be able to unfriend or block other users.

**R5:** Users should be able to follow other users without adding them to their friend list.

**R6:** Users should be able to like, share, and comment on a post. They should also be able to like or comment on an existing comment.

**R7:** The system should send the user a notification whenever there has been an interaction with them, such as receiving a message, a friend request, or a comment on their post.

**R8:** A user should be able to send messages and receive messages from other users.

**R9:** Users should be able to follow existing pages and join existing groups. They should also be able to unfollow or leave joined groups or followed pages.

**R10:** Users should be able to create their own groups and pages. Users can later set privacy or delete the groups or pages they own.

We've identified our requirements for the problem, and in the next lesson, we will define different use cases of Facebook.

# Use Case Diagram for Facebook

Learn how to define use cases and create the corresponding use case diagram for the Facebook problem.

**We'll cover the following**

* [System](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#System)
* [Actors](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Actors)
  + [Primary actors](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Primary-actors)
  + [Secondary actors](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Secondary-actors)
* [Use cases](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Use-cases)
  + [User](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#User)
  + [Page/Group Admin](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Page/Group-Admin)
  + [System](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#System)
* [Relationships](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Relationships)
  + [Generalization](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Generalization)
  + [Associations](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Associations)
  + [Include](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Include)
* [Use case diagram](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Use-case-diagram)

Let's begin building the use case diagram of Facebook and understand the relationship between its different components.

First, we are going to define the different elements of our Facebook system, followed by the complete use case diagram of the system.

## System

Our system is "Facebook."

## Actors

Now, we’ll define the main actors of Facebook.

### Primary actors

* **User:**This actor can create a profile of themselves containing their personal information. They can create posts, pages, and groups, and add various sorts of multimedia like images or videos. They can also interact with other users by sending them friend requests and messages, commenting on their posts, etc.

### Secondary actors

* **Page/Group Admin:** The admin is in charge of performing numerous operations, including blocking or unblocking users from groups or pages, deleting an existing group, changing the group’s privacy, etc.
* **System:** This is responsible for sending out notifications for new friend requests, messages, comments, etc.

## Use cases

In this section, we’ll define the use cases for Facebook. We have listed the use cases according to their respective interactions with a particular actor.

**Note:** You’ll see some use cases occurring multiple times because they are shared among different actors in the system.

### User

* **Add/update profile:**To add information like work, education, and places of visit details or to update an existing profile
* **Follow/unfollow user:**To follow or unfollow other users
* **Send message:**To send a message to other users
* **Send friend request:**To send a friend request to other users
* **Create/like/follow/share page:** To either create a new page or perform actions including liking, following, or sharing an existing page.
* **Create/join/leave group:**To either create a new group or perform actions like joining or leaving an existing group
* **Invite users to group:** To invite other users to an existing group.
* **Add/update/delete post:** To add a new post, update the content of the post, or delete an existing post
* **Like/comment/share post**: To like a post, comment on a post, or share a post
* **Add/update/delete/like comment:**To add a new comment, update the content of the comment, like a particular comment, or delete a comment
* **Accept/reject friend request:** To accept or reject a friend request from another user
* **Update privacy:** To update the privacy settings of the profile
* **Search users/groups/pages/posts:** To search for other users, any existing groups or pages, or any posts made by users
* **Accept/Reject friend request:**To accept or reject a friend request from another Facebook user
* **Accept/Reject group join invitation:**To accept or reject a group joining invitation from another Facebook user
* **Like/Comment/Share Post:**To like, comment, or share any post visible to users on Facebook

### Page/Group Admin

* **Block/unblock user:**To block or unblock a user from a group or page
* **Enable/disable page:**To enable or disable a page
* **Delete group:**To delete an existing group
* **Change a group's privacy:**To change the privacy settings of a group (from public to private and vice versa)

### System

* **Send new friend request notification:**To send a notification of any friend request sent by another user
* **Send message notification:**To send a notification of any new messages
* **Send new post notification:** To send a notification of any new posts
* **Send comment notification:** To send a notification of any comments on your or others' posts

## Relationships

We describe the relationships between and among actors and their use cases in this section.

### Generalization

* The “Page/Group Admin” has a generalization relationship with the “User” as the admin can perform all those tasks that a normal user can perform.
* We can search for other users, groups, pages, or even posts. This shows that the “Search” use case has a generalization relationship with the “users,” “groups,” “pages,” and “posts” use cases.

### Associations

The table below shows the association relationship between actors and their use cases.

|  |  |  |
| --- | --- | --- |
| **User** | **Page/Group Admin** | **System** |
| Add/update profile | Add/update profile | Send new friend request notification |
| Follow/unfollow user | Follow/unfollow user | Send message notification |
| Send message | Send message | Send new post notification |
| Send friend request | Send friend request | Send comment notification |
| Create/like/follow/share page | Create/like/follow/share page |  |
| Create/join/leave group | Create/join/leave group |
| Invite user to group | Invite user to group |
| Add/update/delete post | Add/update/delete post |
| Add/update/delete/like comment | Add/update/delete/like comment |
| Accept/reject friend request | Accept/reject friend request |
| Update privacy | Update privacy |
| Search users/groups/pages/posts | Search users/groups/pages/posts |
| Accept/Reject friend request | Block/unblock user |
| Accept/Reject group join invitation | Enable/disable page |
| Like/Comment/Share Post | Delete group |
|  | Change a group's privacy |

### nclude

* The “Send friend request” use case also has an include relationship with the “Send new friend request notification,” because whenever a user receives a friend request, the user is notified.
* The “Send message” use case has an include relationship with the “Send message notification” use case, because whenever a user receives a message, the user is notified.
* The “Add post” use case has an include relationship with the “Send new post notification” use case. Whenever a new post is created by a user being followed by the default user, the default user is notified.
* The “Add comment” use case has an include relationship with the “Send new comment notification” use case. Whenever a new comment is made by another user on a post either created by the default user or followed by the default user, the default user is notified.

## Use case diagram

Here’s t

A diagram of a group of people

Description automatically generated

# Class Diagram for Facebook

Learn to create a class diagram for Facebook using the bottom-up approach.

**We'll cover the following**

* [Components of Facebook](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Components-of-Facebook)
  + [Account](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Account)
  + [Admin](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Admin)
  + [Person](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Person)
    - [User](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#User)
  + [Profile](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Profile)
  + [Work, education, and places](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Work,-education,-and-places)
  + [Page](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Page)
  + [Group](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Group)
  + [Post](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Post)
  + [Comment](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Comment)
  + [Friend request](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Friend-request)
  + [Message](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Message)
  + [Profile privacy](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Profile-privacy)
  + [Notification](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Notification)
  + [Interfaces implemented by the user](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Interfaces-implemented-by-the-user)
  + [Search catalog](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Search-catalog)
  + [Enumerations](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Enumerations)
  + [Custom data type](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Custom-data-type)
* [Relationship between the classes](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Relationship-between-the-classes)
  + [Association](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Association)
  + [Composition](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Composition)
  + [Aggregation](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Aggregation)
  + [Generalization](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Generalization)
  + [Inheritance](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Inheritance)
* [Class diagram of Facebook](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Class-diagram-of-Facebook)
* [Design pattern](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Design-pattern)
* [Additional requirements](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Additional-requirements)

In this lesson, we’ll identify and design the classes, abstract classes, and interfaces based on the requirements that we have previously gathered from the interviewer in our Facebook social network system.

## Components of Facebook

As mentioned earlier, we will design the Facebook social network using a bottom-up approach.

### Account

The Account class identifies a Facebook user using their username and ID. Users with an account will have the option to create groups, pages, posts, comments, and like other comments and posts.

A representation of the class definition is provided below:

A screenshot of a computer

Description automatically generated

### Person

The Person class contains details like the name, address, phone number, and email. It is derived into the Userclass.

#### User

The User class is the main class of Facebook and is responsible for making various kinds of operations such as sending messages or friend requests to other users, creating or joining groups or pages, and numerous other operations.

The class diagram of these two classes are provided below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a web page

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Class diagram of Facebook

Here’s the complete class diagram for Facebook:

A diagram of a computer

Description automatically generated

## Design pattern

We know that Facebook has a group feature that allows multiple users to join a group. By default, when a user joins a group, they get notified of any new posts in the group. To effectively map this behavior in our design, we can use the Observer design pattern.

On Facebook, all members of the group are set as subscribers by default. Whenever there is a new activity in the group, the subscribers are notified. The members also have the option to opt out of these notifications, which will remove them from the subscriber list.

## Additional requirements

The interviewer can introduce some additional requirements in the Facebook social network system, or they can ask some follow-up questions. Let's see some examples of additional requirements:

* Recommendations: Users can send recommendations of pages and groups to other users. The class diagram provided below shows the relationship of Recommendation with the other classes:

A screenshot of a computer

Description automatically generated

# equence Diagram for Facebook

Visualize the sequence diagram for sending a friend request on Facebook.

**We'll cover the following**

* [Send a friend request](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Send-a-friend-request)

Sequence diagrams are a great way to understand the interactions between different entities and objects in the system. There can be different sequence diagrams that we can create for Facebook. In this lesson, we will create sequence diagrams for the following interaction:

* **Send a friend request:** A user sends a friend request to another user.

## Send a friend request

The sequence diagram for sending a friend request should have the following actors and objects that will interact with each other:

* **Actors:** User A and User B
* **Objects:** Catalog and FriendRequest

Here’re the steps for the “send a friend request” interaction:

1. User A searches for user B in the catalog.
2. If User B exists:
   1. The catalog returns user B
   2. User A adds user B as a friend.
   3. The friend request is sent to user B.
   4. If the request is accepted:
      1. User B is added to user A's friend list.
      2. User A gets a notification that the friend request has been accepted.
   5. Else, the request is rejected.
3. Else, if user B does not exist:
   1. User A receives a "user not found" error.

Based on the order above, the sequence diagram for sending a friend request on Facebook is given below:

A screenshot of a computer screen

Description automatically generated

# Activity Diagram for Facebook

Create some activity diagrams for the Facebook problem.

**We'll cover the following**

* [Creating a new post](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Creating-a-new-post)
  + [States](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#States)
  + [Actions](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Actions)
* [Activity challenge: A user joins a group](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Activity-challenge:-A-user-joins-a-group)

Activity diagrams are a great way to visualize the flow of messages from one activity to the other in the system. There can be different activity diagrams that we can create for Facebook. In this lesson, we will create activity diagrams for the following two activities:

* Creating a new post
* **Activity challenge:**A user joins a group.

## Creating a new post

The states and actions that will be involved in this activity diagram are provided below.

### States

**Initial state:**The user selects the new post option.

**Final state:** The post is published.

### Actions

The user selects the new post option, selects the privacy option, adds any attachments, and publishes the post.

A diagram of a website

Description automatically generated

## Activity challenge: A user joins a group

You’ll help us create an activity diagram of a user joining a group.A skeleton of the activity diagram is given below:

A diagram of a group

Description automatically generated

The slots below represent the activities, and the arrows represent the flow from one activity to the other.

Can you rearrange the slots below in the correct order they should appear in the activity diagram given above?

**Note:**If you get stuck, just click the “Show Solution” button to check the correct answer.

Alternatively, you can click the "Show complete diagram" button below to see the complete activity diagram.

A screenshot of a diagram

Description automatically generated

# Code for Facebook

Write object-oriented code to implement the design of the Facebook problem.

**We'll cover the following**

* [Facebook classes](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Facebook-classes)
  + [Constants](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Constants)
  + [Interfaces implemented by user](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Interfaces-implemented-by-user)
  + [Account](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Account)
  + [Person, user, and admin](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Person,-user,-and-admin)
  + [Profile, education, places, and work](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Profile,-education,-places,-and-work)
  + [Page, post, and comment](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Page,-post,-and-comment)
  + [Profile privacy](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Profile-privacy)
  + [Group and the group functions interface](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Group-and-the-group-functions-interface)
  + [Message and friend request](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Message-and-friend-request)
  + [Notification](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Notification)
  + [Search catalog and interface](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Search-catalog-and-interface)
* [Wrapping up](https://www.educative.io/courses/grokking-the-low-level-design-interview-using-ood-principles/getting-ready-the-amazon-online-shopping-system#Wrapping-up)

We’ve reviewed different aspects of Facebook and observed the attributes attached to the problem using various UML diagrams. Let’s now explore the more practical side of things, where we will work on implementing the Facebook network using multiple languages. This is usually the last step in an object-oriented design interview process.

We have chosen the following languages to write the skeleton code of the different classes present in Facebook:

* Java
* C#
* Python
* C++
* JavaScript

## Facebook classes

In this section, we will provide the skeleton code of the classes designed in the class diagram lesson.

**Note:**For simplicity, we are not defining getter and setter functions. The reader can assume that all class attributes are private and accessed through their respective public getter methods and modified only through their public method functions.

### Constants

The following code provides the definition of the various enums and custom data types being used in the Facebook design:

**Note:**JavaScript does not support enumerations so we will be using the Object.freeze() method as an alternative that freezes an object and prevents further modifications.

Java

public class Address {

private int zipCode;

private String houseNo;

private String city;

private String state;

private String country;

}

enum AccountStatus {

ACTIVE,

BLOCKED,

DISABLED,

DELETED

}

enum FriendInviteStatus {

PENDING,

ACCEPTED,

REJECTED,

CANCELED

}

enum PostPrivacySettings {

PUBLIC,

FRIENDS\_OF\_FRIENDS,

ONLY\_FRIENDS,

CUSTOM

}

### Interfaces implemented by user

Facebook will have several interfaces that will be implemented by users and are described below:

* PageFunctionsByUser: This will define the functions that a user will perform while interacting with pages.
* GroupFunctionsByUser: This will define the functions that a user will perform while interacting with groups.
* PostFunctionsByUser: This will define the functions that a user will perform while interacting with posts.
* CommentFunctionsByUser: This will define the functions that a user will perform while interacting with comments.

The definition of all these interfaces is given below:

public interface PageFunctionsByUser {

public Page createPage(String name);

public Page sharePage(Page page);

public void likePage(Page page);

public void followPage(Page page);

public void unLikePage(Page page);

public void unFollowPage(Page page);

}

public interface GroupFunctionsByUser {

public Group createGroup(String name);

public void joinGroup(Group group);

public void leaveGroup(Group group);

public void sendGroupInvite(Group group);

}

public interface PostFunctionsByUser {

public Post createPost(String content);

public Post sharePost(Post post);

public void commentOnPost(Post post);

public void likePost(Post post);

}

public interface CommentFunctionsByUser {

public Comment createComment(Post post, String content);

public void likeComment(Comment comment);

}

A screenshot of a computer

Description automatically generated

public class Admin {

public void blockUser(User user);

public void unblockUser(User user);

public void enablePage(Page page);

public void disablePage(Page page);

public void deleteGroup(Group group);

public void changeGroupPrivacy(Group group);

}

// Person will be an abstract class

public abstract class Person {

private String name;

private Address address;

private String email;

private String phone;

private Account account;

}

// Will be using only one interface example

public class User extends Person implements PageFunctionsByUser{

private int userId;

private String name;

private Date dateOfJoining;

// The following lists contain the pages and groups that a user is admin of

private List<Pages> pagesAdmin;

private List<Groups> groupsAdmin;

private Profile profile;

public boolean sendMessage(Message message);

public boolean sendRecommendation(Page page, Group group, User user);

public boolean sendFriendRequest(User user);

public boolean unfriendUser(User user);

public boolean blockUser(User user);

public boolean followUser(User user);

// The functions of the different interfaces will also be present here

public Page createPage(String name) {

// functionality

}

public void likePage(Page page){

// functionality

}

public void followPage(Page page){

// functionality

}

public void unLikePage(Page page){

// functionality

}

public void unFollowPage(Page page){

// functionality

}

public Page sharePage(Page page){

// functionality

}

}

### Profile, education, places, and work

The Work, Education, and Places classes will provide a user's personal information and will make up the Profile class. The definition of these classes is given below:

public class Profile {

private String gender;

private byte[] profilePicture;

private byte[] coverPhoto;

private List<User> friends;

private List<int> usersFollowed;

private List<int> pagesFollowed;

private List<int> groupsJoined;

private List<Work> workExperience;

private List<Education> educationInfo;

private List<Place> places;

public boolean addWorkExperience(Work work);

public boolean addEducation(Education education);

public boolean addPlace(Place place);

public boolean addProfilePicture(byte[] image);

public boolean addCoverPhoto(byte[] image);

public boolean addGender(String gender);

}

public class Work {

private String title;

private String company;

private String location;

private String description;

private Date startDate;

private Date endDate;

}

public class Places {

private String name;

}

public class Education {

private String school;

private String degree;

private String description;

private Date startDate;

private Date endDate;

}

### Page, post, and comment

Facebook users can create and like pages, posts, and comments. The definition of these classes is given below:

public class Page {

private int pageId;

private String name;

private String description;

private int likeCount;

}

public class Post {

private int postId;

private String content;

private byte[] image;

private int likeCount;

private int shareCount;

private User postOwner;

private PostPrivacySettings settings;

public changePostVisibility(Post post);

}

public class Comment {

private int commentId;

private String content;

private int likeCount;

private User commentOwner;

}

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

public class Message {

private int messageId;

private User sender;

private String content;

private List<User> recipent;

private List<byte[]> multimedia;

public boolean addRecipent(List<User> users);

}

public class FriendRequest {

private User recipent;

private User sender;

private FriendRequestStatus status;

private Date requestSent;

private Date requestStatusModified;

public boolean acceptRequest(User user);

public boolean rejectRequest(User user);

}

A screenshot of a computer

Description automatically generated

public interface Search {

public List<User> searchUsers(String name);

public List<Group> searchGroups(String name);

public List<Page> searchPages(String name);

public List<Post> searchPosts(String keywords);

}

public class SearchCatalog implements Search {

HashMap<String, List<User>> userNames;

HashMap<String, List<Group>> groupNames;

HashMap<String, List<Page>> pageTitles;

HashMap<String, List<Post>> posts;

public boolean addNewUser(User user) {}

public boolean addNewGroup(Group group) {}

public boolean addNewPage(Page page) {}

public boolean addNewPost(Post post) {}

public List<User> searchUsers(String name) {

// functionality

}

public List<Group> searchGroups(String name) {

// functionality

}

public List<Page> searchPages(String name) {

// functionality

}

public List<Post> searchPosts(String keywords) {

// functionality

}

}

## Wrapping up

We've explored the complete design of Facebook in this chapter. We've looked at how Facebook can be visualized using various UML diagrams and designed using object-oriented principles and design patterns.