

Module 1: Authentication

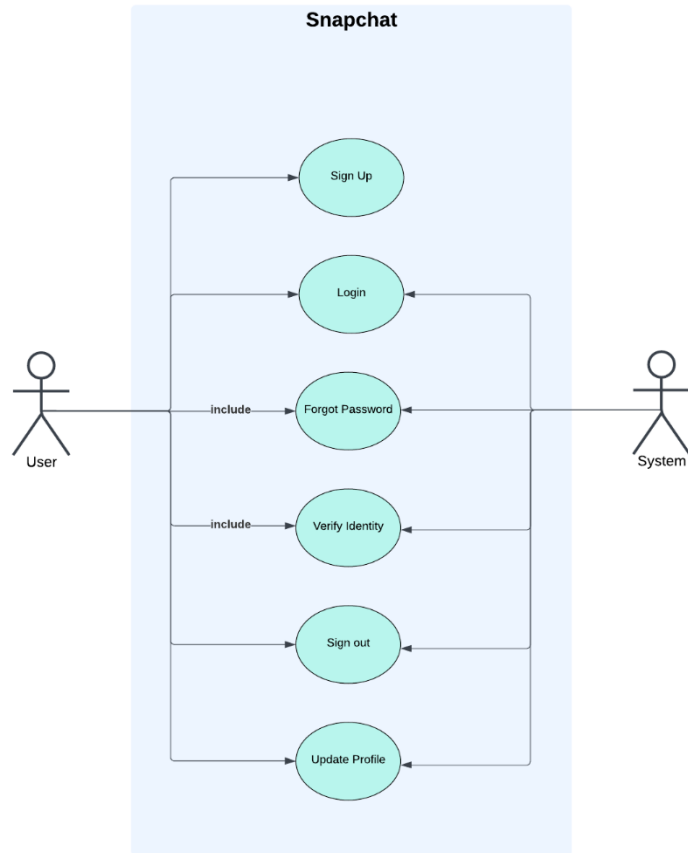
i. Use Case Diagram

Use Case Diagram of Snapchat: Authentication

This flowchart outlines the user journey for Snapchat, including options for signing up, logging in, verifying identity, updating profile information, and signing out. The system is also depicted as a separate entity connected to the user flow.

Use case shape key

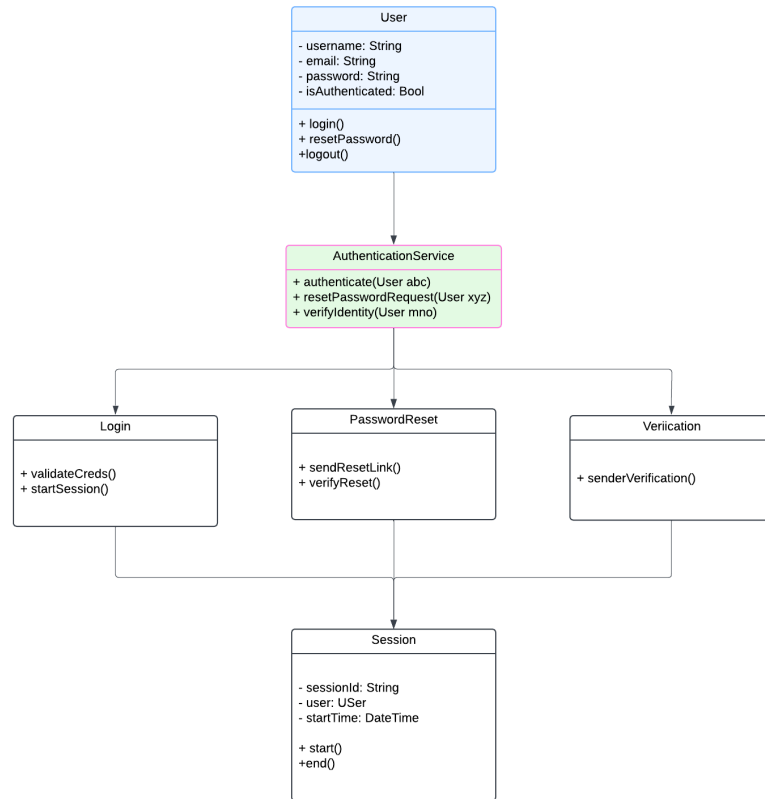
- Actor
- Use case
- Use case extension
- System



ii. Class Diagram

**Class Diagram of Snapchat:
Authentication**

The **User** class interacts with **AuthenticationService** for authentication, password reset, and verification. **AuthenticationService** coordinates with **Login**, **PasswordReset**, and **Verification** classes for specific tasks, while **Session** handles active user sessions post-login.



iii. Activity Diagram

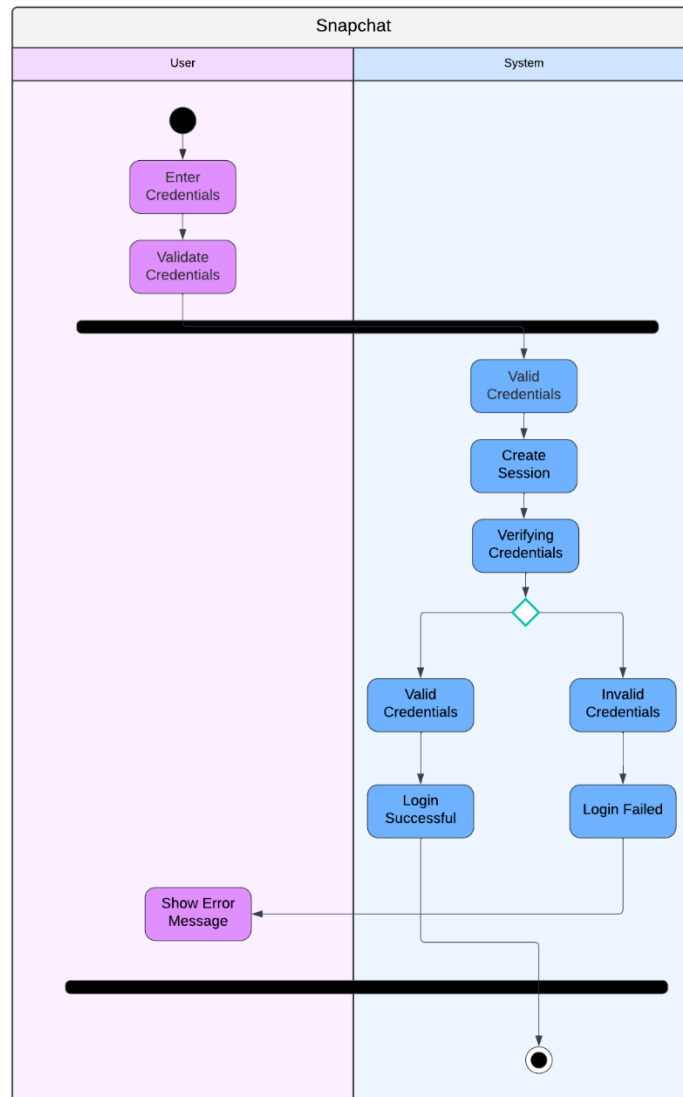


Snapchat Login Flowchart

The flowchart depicts the Snapchat login process. After entering credentials, the system validates them and either creates a session or shows an error message. Successful login leads to a login successful action while invalid credentials lead to a login failed action.

Activity diagram shape key

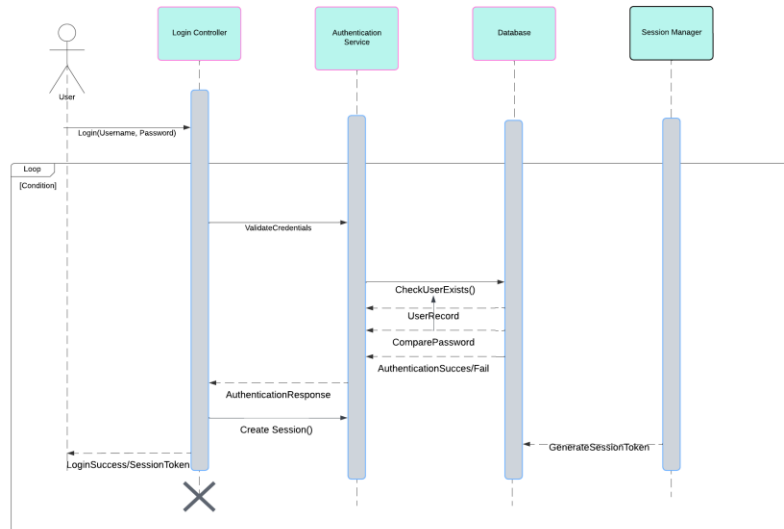
- Start
- Action
- Horizontal fork/Join
- ◇ Branch/Merge
- ★ End
- Team/department name
- Team/department name



iv. Sequence Diagram

Sequence Diagram of Snapchat: Authentication

This sequence diagram shows the **User Authentication** process in Snapchat. When a **User** submits login credentials, the **Login Controller** forwards them to the **Authentication Service** for validation. The **Authentication Service** queries the **Database** to check if the user exists and compares the provided password with the stored password hash. If the credentials are correct, **Session Manager** generates a session token to maintain the user's session, and a success message along with the token is sent back to the **User**. If authentication fails, the user is notified of the login failure.



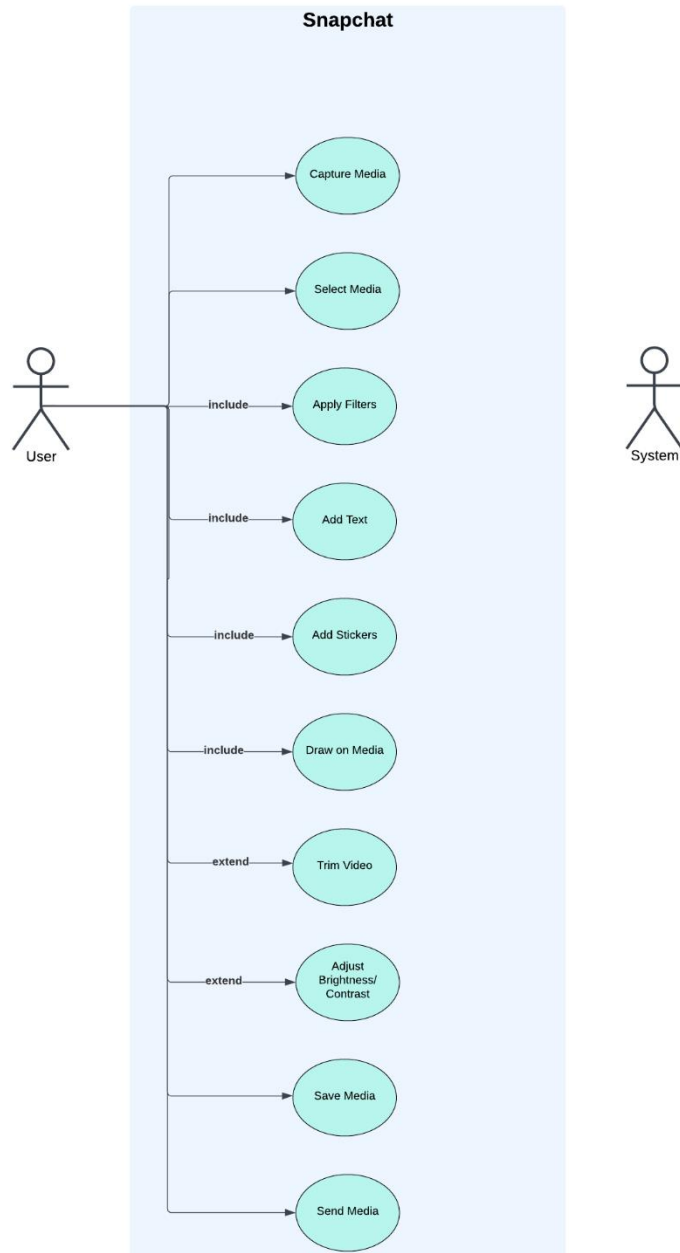
Module 2: Media Capturing and Editing

i. Use Case Diagram

Use Case Diagram of Snapchat: Media Capturing and Editing

This flowchart outlines the media editing process in Snapchat, including features such as applying filters, adding text and stickers, drawing on media, and adjusting brightness/contrast. The flowchart also includes steps for capturing, selecting, saving, and sending media, with some steps being optional extension points.

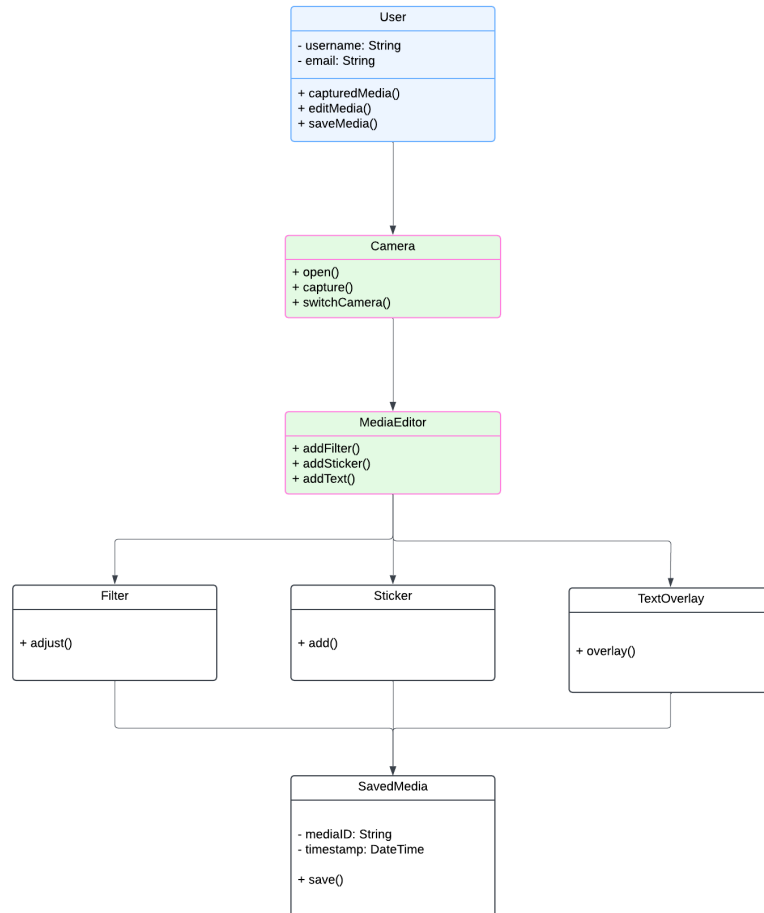
Use case shape key
Actor
Use case
Use case extension
System



ii. Class Diagram

Class Diagram of Snapchat: Media Capturing and Editing

In this Media Capturing and Editing class diagram, the **User** initiates media capture via the **Camera** and **MediaCapture** classes. **MediaEditor** enables adding effects like **Filters**, **Stickers**, and **TextOverlay**, while **SavedMedia** stores the edited media with unique identifiers. This setup allows users to capture, edit, and save media seamlessly within Snapchat.



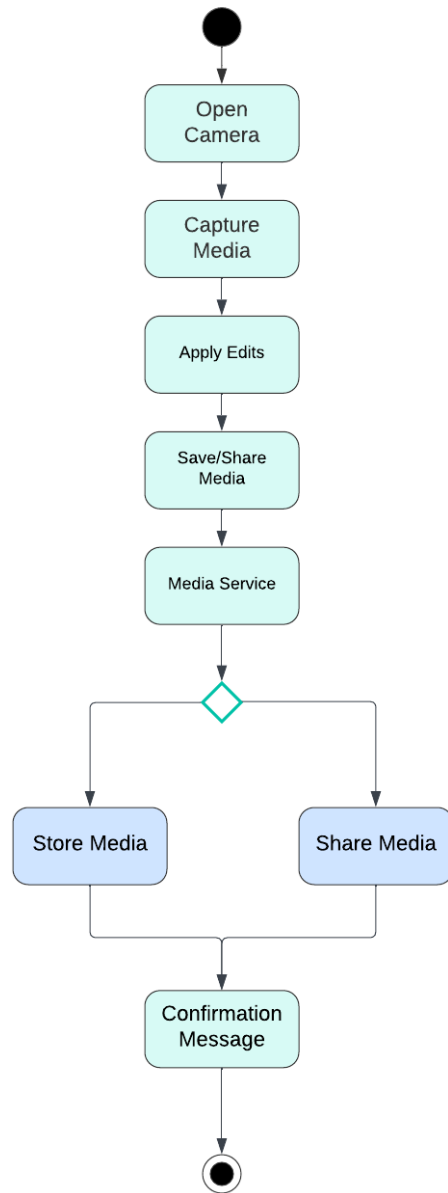
iii. Activity Diagram

Activity Diagram of Snapchat: Media Capturing and Editing

The flowchart depicts the Snapchat login process. After entering credentials, the system validates them and either creates a session or shows an error message. Successful login leads to a login successful action while invalid credentials lead to a login failed action.

Activity diagram shape key

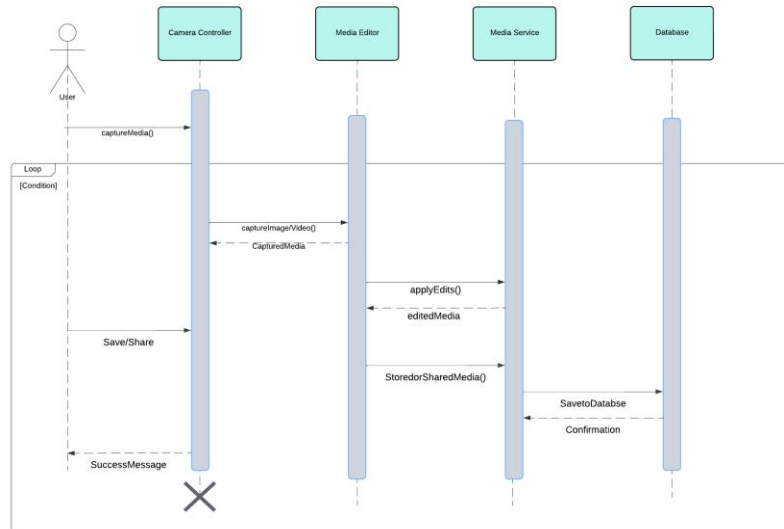
- Start
- Action
- Horizontal fork/Join
- ◇ Branch/Merge
- End
- Team/department name
- Team/department name



iv. Sequence Diagram

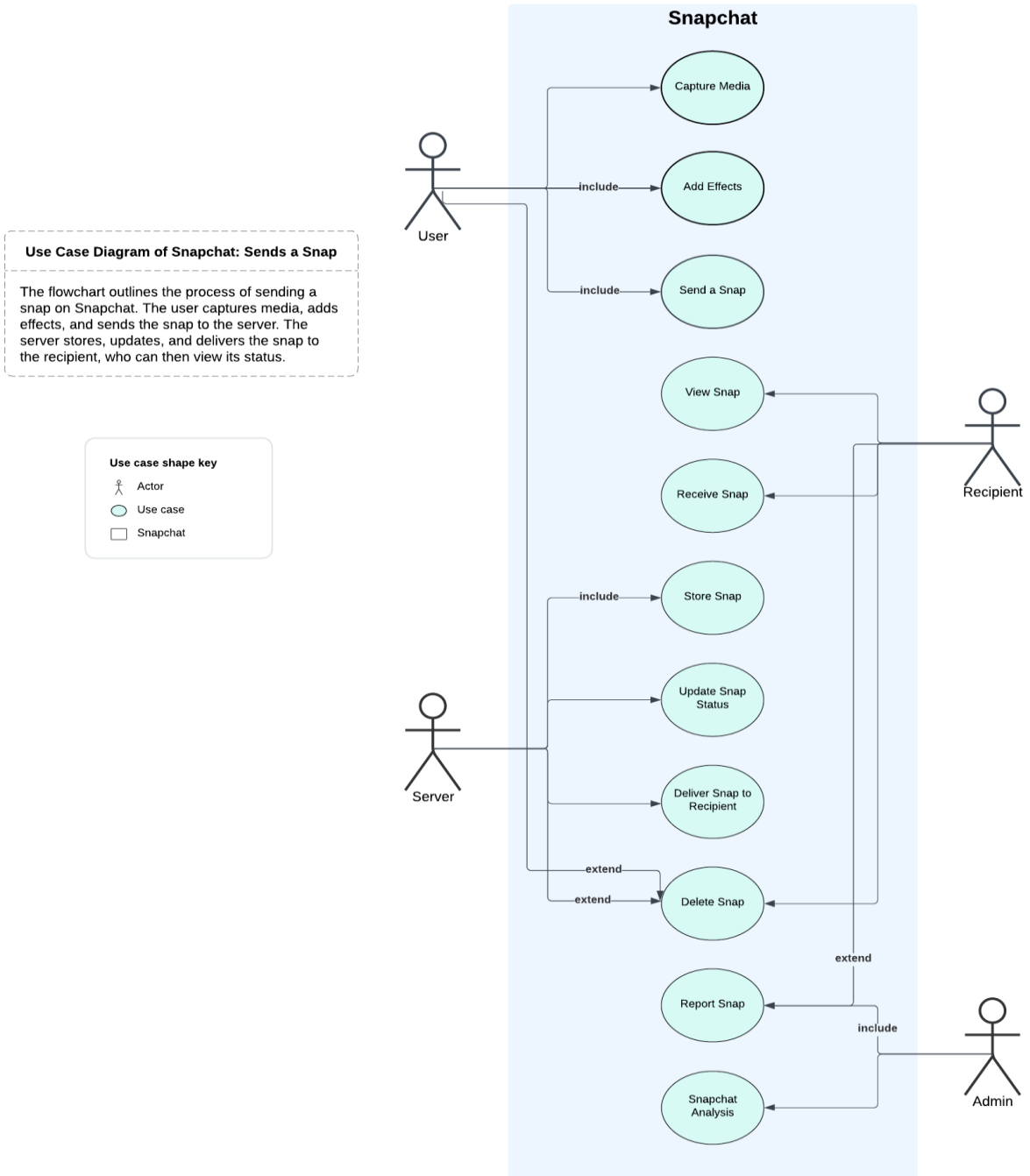
Sequence Diagram of Snapchat: Media
Capturing and Editing

This sequence diagram illustrates the **Media Capturing and Editing** process in Snapchat. The **User** initiates media capture via the **Camera Controller**, which captures a photo or video. The media is then passed to the **Media Editor**, where the user can apply edits, such as filters or text. After editing, the user can choose to save or share the media. If saved, the **Media Service** stores it with metadata in the **Database**. If shared, the **Media Service** sends it to the selected recipients and logs it. Finally, a success message confirms the media's save or share status.



Module 3: Send and Receive a Snap

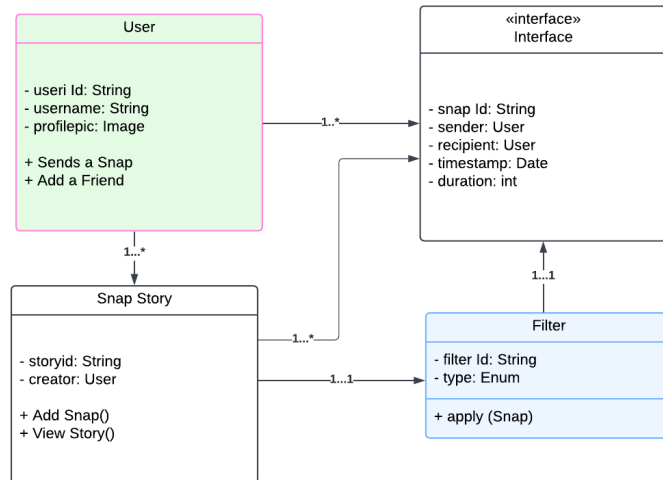
i. Use Case Diagram



ii. Class Diagram

Class Diagram of Snapchat: Sends a Snap

The Snapchat system consists of a User interface with attributes such as user ID and profile picture, an Interface with attributes such as snap ID and sender/recipient information, a Filter with an apply function for snaps, and a Snap Story interface with attributes such as story ID and creator information. Users can send snaps and add friends, while Snap Stories can be viewed and added to.



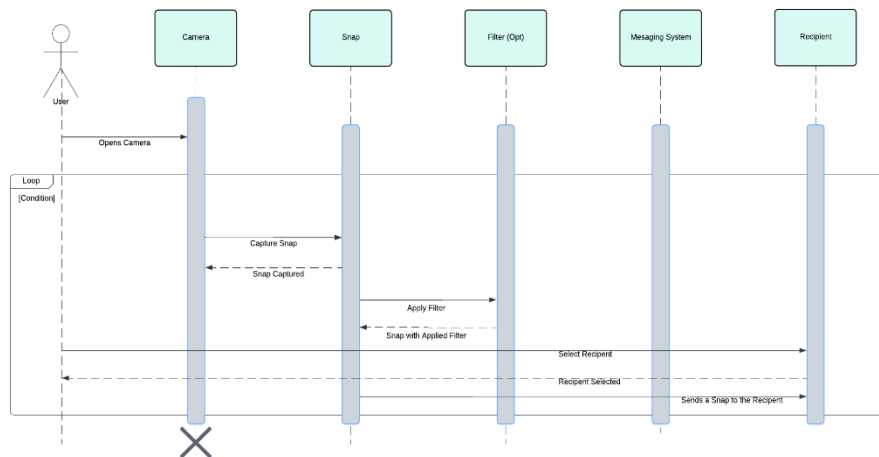
iii. Sequence Diagram

Sequence diagram shapes

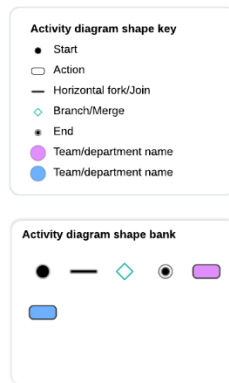


Sequence Diagram of Snapchat: Sends a Snap

The sequence for sending a snap begins with the user opening the Snapchat app and selecting a snap. The user then captures or selects a snap, edits it if desired, sets options like time limits, and sends it. The app confirms whether the snap was successfully sent or if there was an issue.

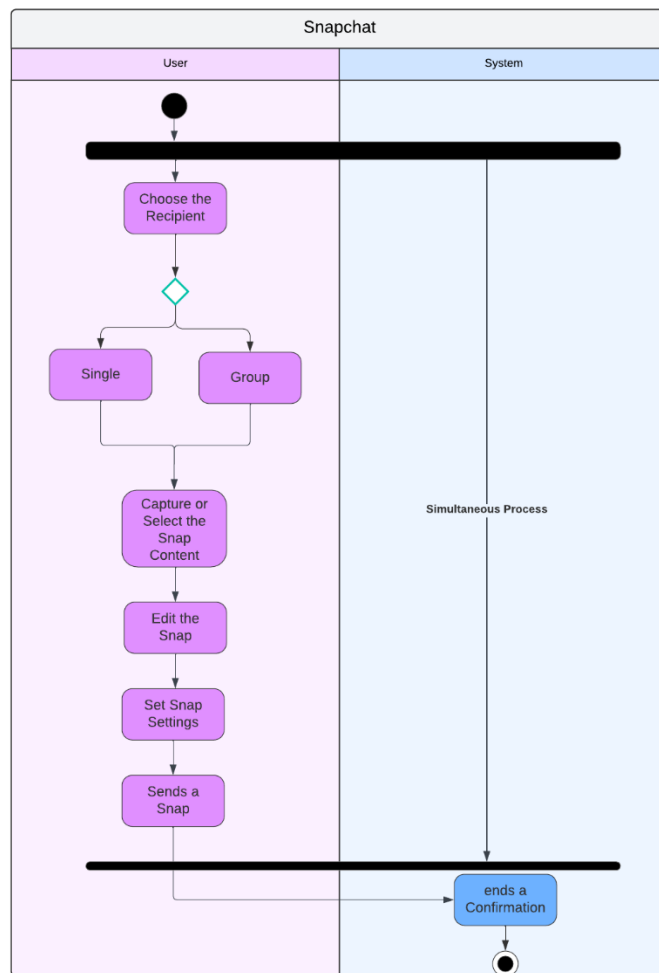


iv. Activity Diagram



Activity Diagram of Snapchat: Sends A Snap

The Snapchat sending process involves capturing or selecting snap content, editing it, setting snap settings, choosing a recipient, and sending the snap. The system then sends a confirmation to the user and the recipient receives the snap simultaneously.



Module 4: Snap Storage and Management

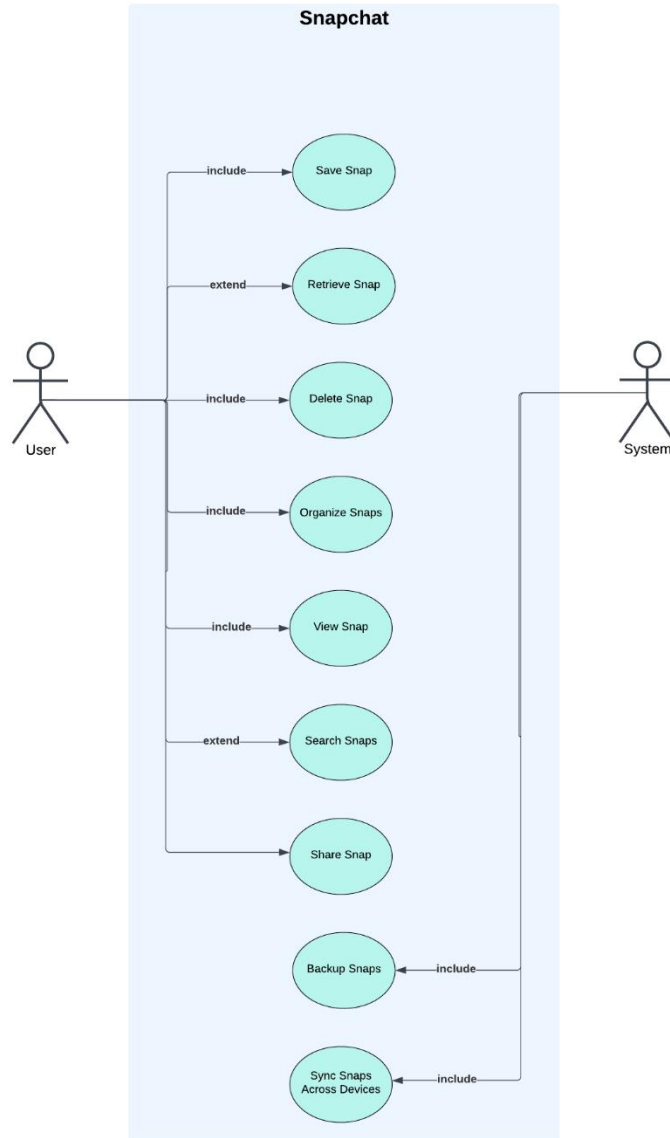
i. Use Case Diagram

Use Case Diagram of Sapchat: Snap Storage and Management

This flowchart outlines the Snapchat Snap Management System, which includes organizing, saving, and sharing snaps, as well as syncing them across devices and searching for them. The system also allows for deleting and retrieving snaps, with backup as an extension point.

Use case shape key

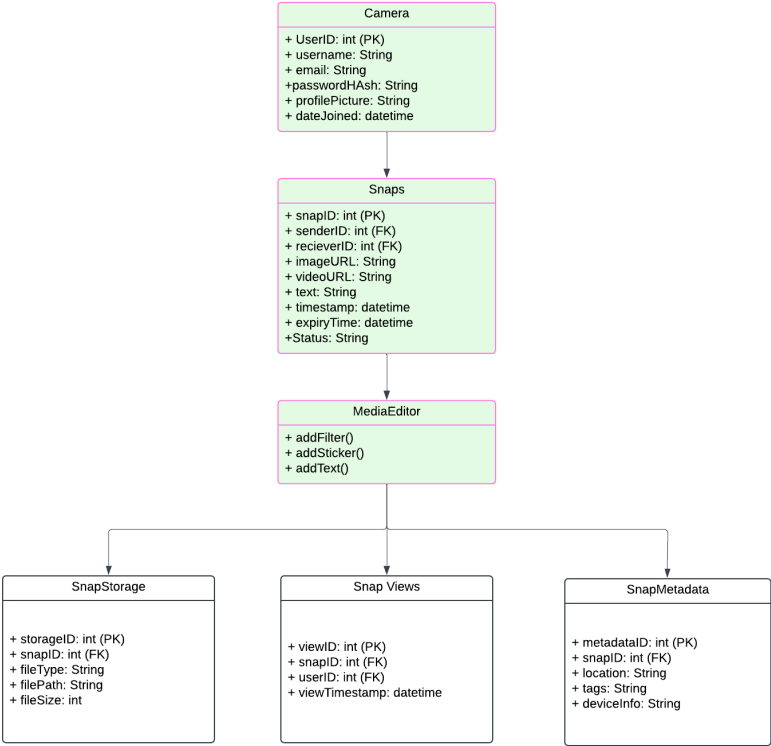
- Actor
- Use case
- Use case extension
- System



ii. Class Diagram

Class Diagram of Snapchat: Snap Storing and Mangement

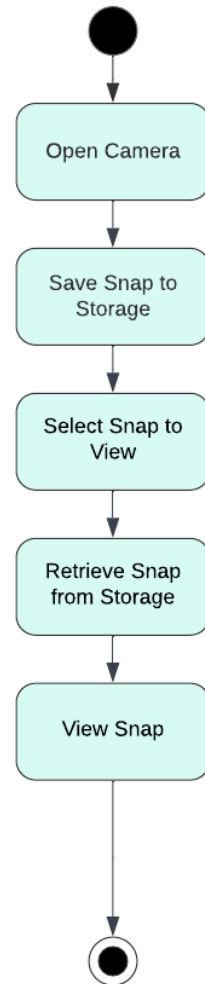
IThis class diagram represents the Snap Storage and Management system, where the **Users** table manages user information, and the **Snaps** table tracks individual snaps sent or received. The **SnapStorage** table stores snap files, **SnapViews** records user interactions with snaps, and **SnapMetadata** holds additional details like location and tags. Relationships are defined through foreign keys linking users, snaps, and metadata.



iii. Activity Diagram

Activity Diagram of Snapchat: Snap Storage and Management

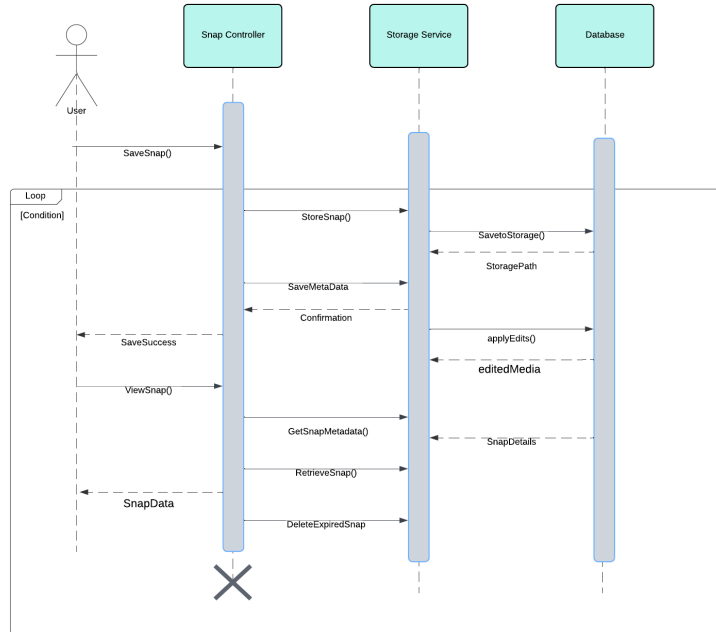
This **Activity Diagram** illustrates the process of **Snap Storage and Management** in Snapchat. The **User** captures or receives a snap, saves it to storage, and later selects it to view. The **System** stores the snap and retrieves it when the user requests to view it.



iv. Sequence Diagram

Sequence Diagram of Snapchat: Snap Storage and Management

This sequence diagram outlines the **Snap Storage and Management** process in Snapchat. When a **User** saves a snap, the **Snap Controller** stores it via the **Storage Service** and logs metadata in the **Database**. Later, if the user wants to view the snap, the **Snap Controller** retrieves metadata, accesses the snap through the **Storage Service**, and displays it. Expired snaps are deleted automatically, ensuring Snapchat's ephemeral experience.



Module 5: Chat and Messaging

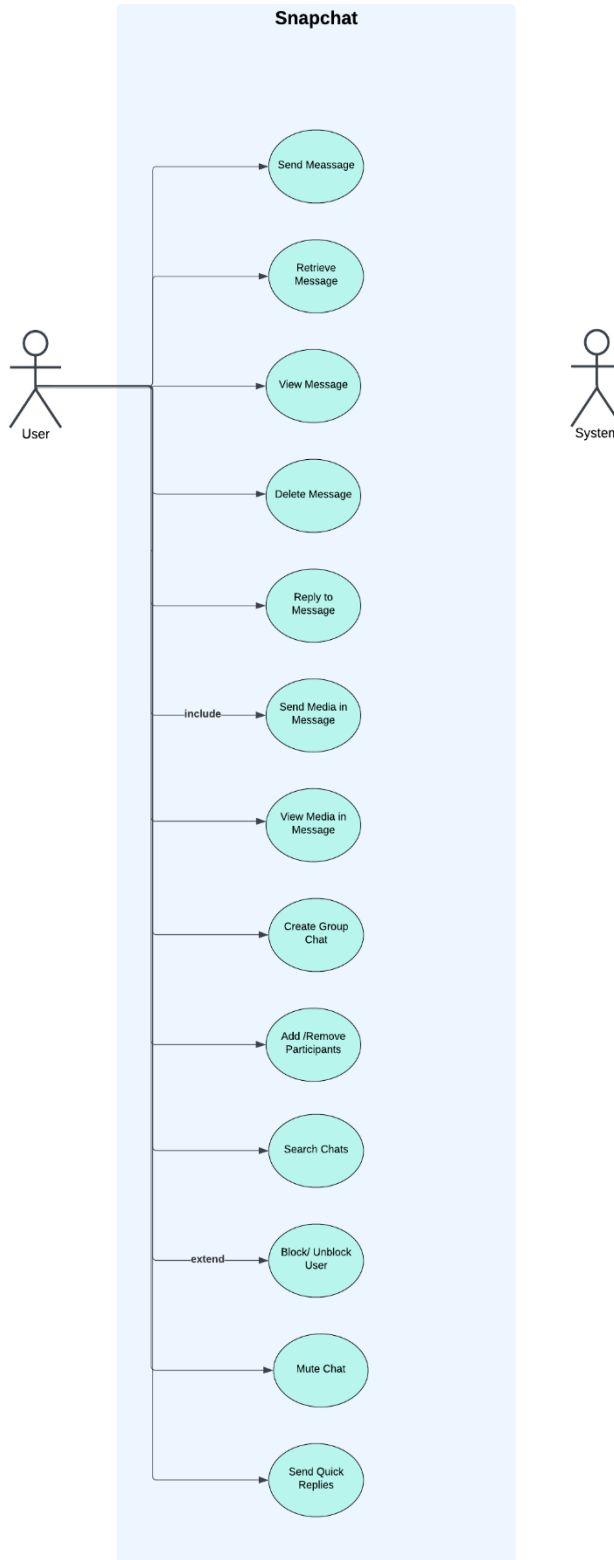
i. Use Case Diagram

Use Case Diagram of Snapchat: Chat and Messaging

This flowchart outlines the various functionalities available in Snapchat's chat feature, including creating group chats, blocking/unblocking users, sending messages and media, muting chats, searching chats, and retrieving and viewing messages. Each functionality has extension points for additional features, such as sending quick replies or adding/removing participants.

Use case shape key

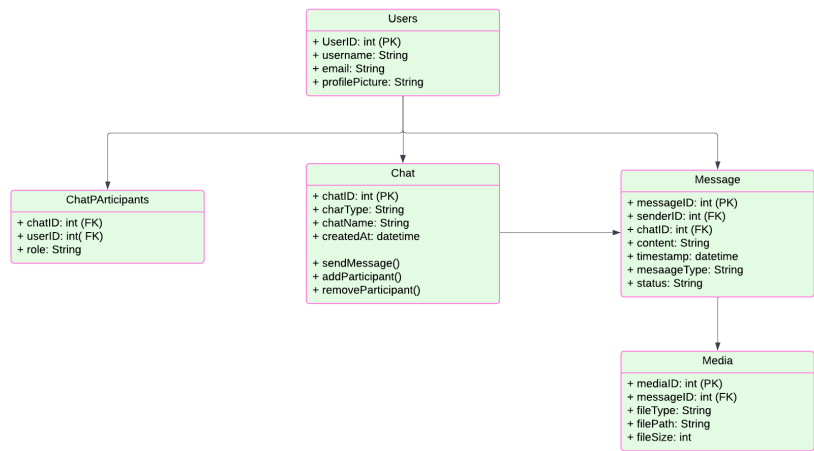
- Actor
- Use case
- Use case extension
- System



ii. Class Diagram

Class Diagram of Snapchat: Chat and Messaging

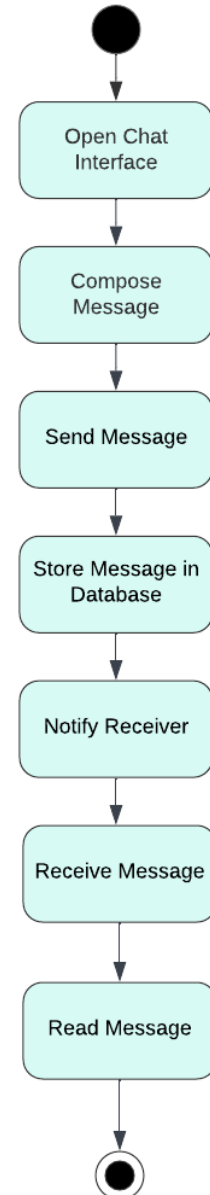
This class diagram models Snapchat's **Chat and Messaging** feature, where **Users** participate in **Chats** that contain multiple **Messages**. Each message can have text or media files (like images or videos) and includes details like timestamp and status. **ChatParticipants** handles user roles in group chats, supporting many-to-many relationships between users and chats, enabling smooth message exchange and multimedia sharing.



iii. Activity Diagram

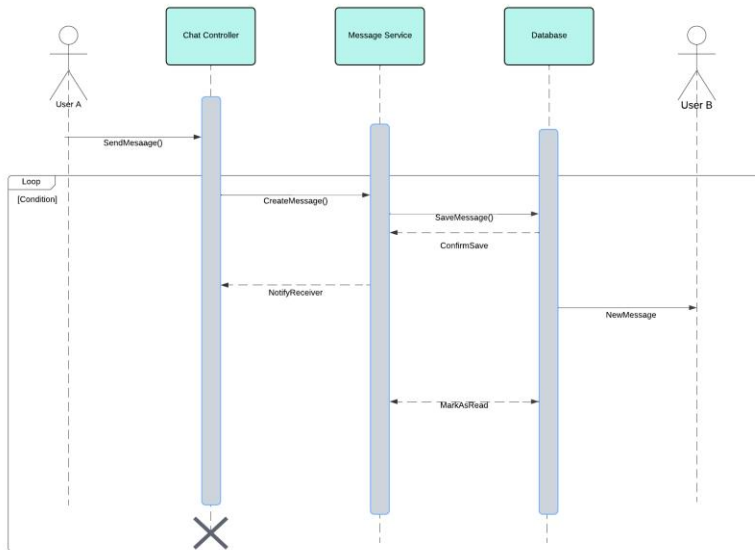
Activity Diagram of Snapchat: Chat and Messaging

This **Activity Diagram** illustrates the process of **Chat and Messaging** in Snapchat. The **User** opens the chat interface, composes and sends a message, and later receives and reads messages. The **System** stores the message in the database, notifies the receiver, and ensures the message is delivered successfully. This diagram shows the flow of actions involved in communication between users within the app.

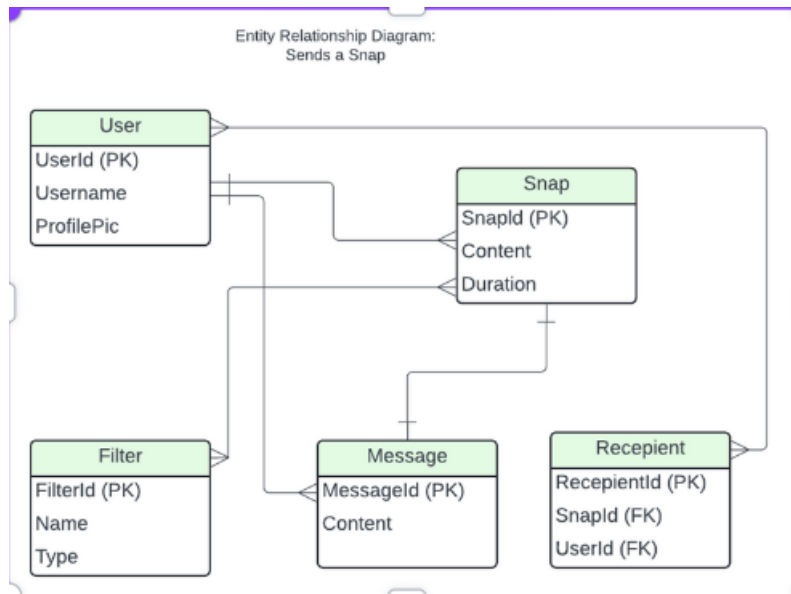


iv. Sequence Diagram

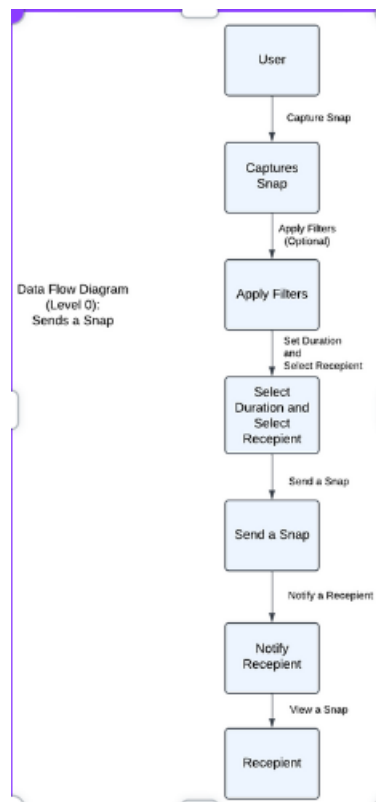
Sequence Diagram of Snapchat: Chat and Messaging
This sequence diagram illustrates the **Chat and Messaging** process in Snapchat. When **User A** sends a message, the **Chat Controller** forwards it to the **Message Service**, which stores the message in the **Database**. **User B** receives a notification, reads the message, and the message status is updated to "Read" in the database.



ER Diagram

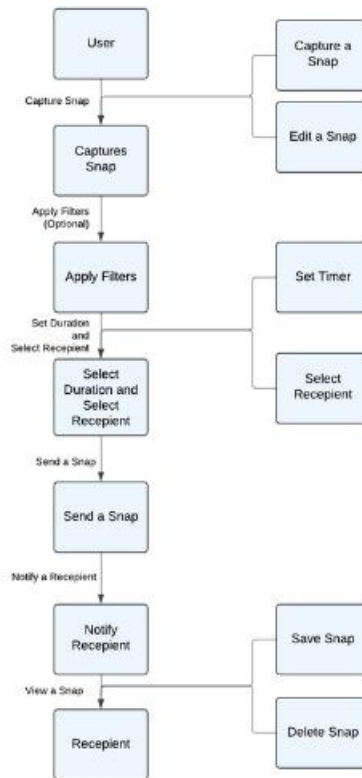


DFD: Level 0



DFD: Level 1

Data Flow Diagram
(Level 1):
Sends a Snap



DFD: Level 2

