Project Title

DISCORD

Submitted to: Sir Fawad

Submitted by

Uzair Hassan (48525)

Daniyal Wajid (48528)

School of Computing and Innovation

Riphah International University, Lahore Campus.

Date of Submission: 15-Jan-24

Introduction:

Discord is a voice, video, and text chat tool that tens of millions of users use to communicate and socialize with friends and groups. For those searching for an easy to use low-resource voice chat software that can do almost anything, this is the ideal option. Everyone may feel like they belong in Discord, in a manner. While offering a wide variety of information, most social media platforms have a certain focus. Instagram is all about fashion trends, Discord was all about gaming, and Twitter is all about politics and pop culture. Discord now offers a server for anything, including intellectual stuff as well as games, movies, art, and memes.

Objective:

Discord aims to provide smooth real-time conversation, support community development via modifiable servers, and facilitate efficient content exchange. With the help of the platform, users should be able to engage in lively communities that share media material, have discussions, and have similar interests in real-time.

Problem Description:

The primary goal of Discord's creation was to address a major problem: the challenge of facilitating online gaming conversations with friends everywhere. At the time, all the tools built for this job were slow, unreliable, and complex. Discord made it easy to genuinely communicate with friends, going beyond casual talking. Friends were staying in touch with their different communities. Discord made it simple to engage in discussion by allowing users to switch between text, audio, and video chat.

The initiative can improve communication and coordination across professional teams, educational institutions, and gaming communities, enabling real-time communication and information sharing.

By providing a centralized, user-friendly platform, the main objective is to improve community development, communication, and cooperation. Discord effortlessly combines text, voice, and video channels into a single platform, revolutionizing real-time communication.

Discord's real-time features facilitate rapid and easy communication, whether you're catching up with friends, working on a project together, or planning strategy during a gaming session.

Features:

- Integration with other apps
- User status and Presence
- Cross-platform Support
- Accessibility feature
- Nitro Subscription
- Server templates
- User roles and permissions
- Share screens
- Community engagements
- Customizable profiles
- Server Boosts

Users:

- General Users
- Content creators
- Administrators and Moderators
- Bot developers
- Business or Educational Users
- Bot Users
- Server Admins

Requirements:

1. Login:

a. Account registration
b. Recovery and email verification

2. Direct messages:

a. End-to-end encrypted
b. Send and receive messages
c. Message reactions

3. File sharing:

Users can send and share files to one another

4. Server creation:

a. Text and voice channels
b. User roles and permissions

5. User status:

a. User is Online
b. User is Offline

6. Notifications:

Users can receive alert messages

7. Share Screen:

a. Reliability
b. Selecting Application\Screen

8. Search Navigation:

Streamline search and navigation

9. Modify profiles

a. Users can customize their profiles
b. Users can apply themes to their profiles

10. Activity feed:

a. Joining and Leaving servers
b. Security

11. User blocking:

a. Mute Users
b. Block and kick users

12. Community Rules

Communities can conduct guidelines according to their requirements

13. Performance:

a. Response time of functions
b. Load handling

14. Scalability

a. Queue management and elasticity
b. Database scaling

15. Usability

a. Easy to use
b. Understandable GUI

16. Availability

Continuous access and uptime

17. Stakeholder constraints

Maintaining various user groups' demands and preferences

18. Security Constraints:

Adherence to industry encryption and secure communication standards

19. Resource constraints:

Resources for servers and hosting to meet demand and user growth.

20. Content moderation:

To provide a secure workplace, use both automatic and manual moderation techniques to identify and manage unwanted material.

21. Real time communication:

a. Facilitate real-time text and voice communication
b. Make sure that voice chat and text messaging have low latency

References:

1. https://discord.com/safety/360044149331-what-is-discord
2. https://blog.hootsuite.com/what-is-discord/

Functional Requirements Description:

Functional Req. ID #	Function Name	Function Requirement Description	Software Requirement
FR1	Login	Registration and authentication	1a
FR2	Direct messages	User can send and receive messages	2b
FR3	File sharing	Users can share and receive files	3
FR4	Server Creation	Users can create servers and customize them	4a
FR5	User Status	User can set their status to online status	5
FR6	Notifications	Users can receive instant alert messages	6
FR7	Share Screen	Users can share their screens with one another	7b
FR8	Search Navigation	Streamline search and navigation	8
FR9	Modify profiles	Users can customize their profiles	9
FR10	Activity feed	Check the users activity in servers	10a
FR11	User blocking	Admins can block and mute users	11b
FR12	Community Rules	Community conduct guidelines	12

Non-functional Requirements Description

Non-Functional Req. ID #	Function Name	Non-Function Requirement Description	Software Requirement
NFR1	Performance	How the system handles the Load	13b
NFR2	Scalability	Efficiently scales with Load	14
NFR3	Security	Chats and data is end-to-end encrypted	2a
NFR4	Reliability	System performance is consistent on high load	7a
NFR5	Usability	GUI is easy to understand	15b
NFR6	Availability	Continuous access and uptime	16

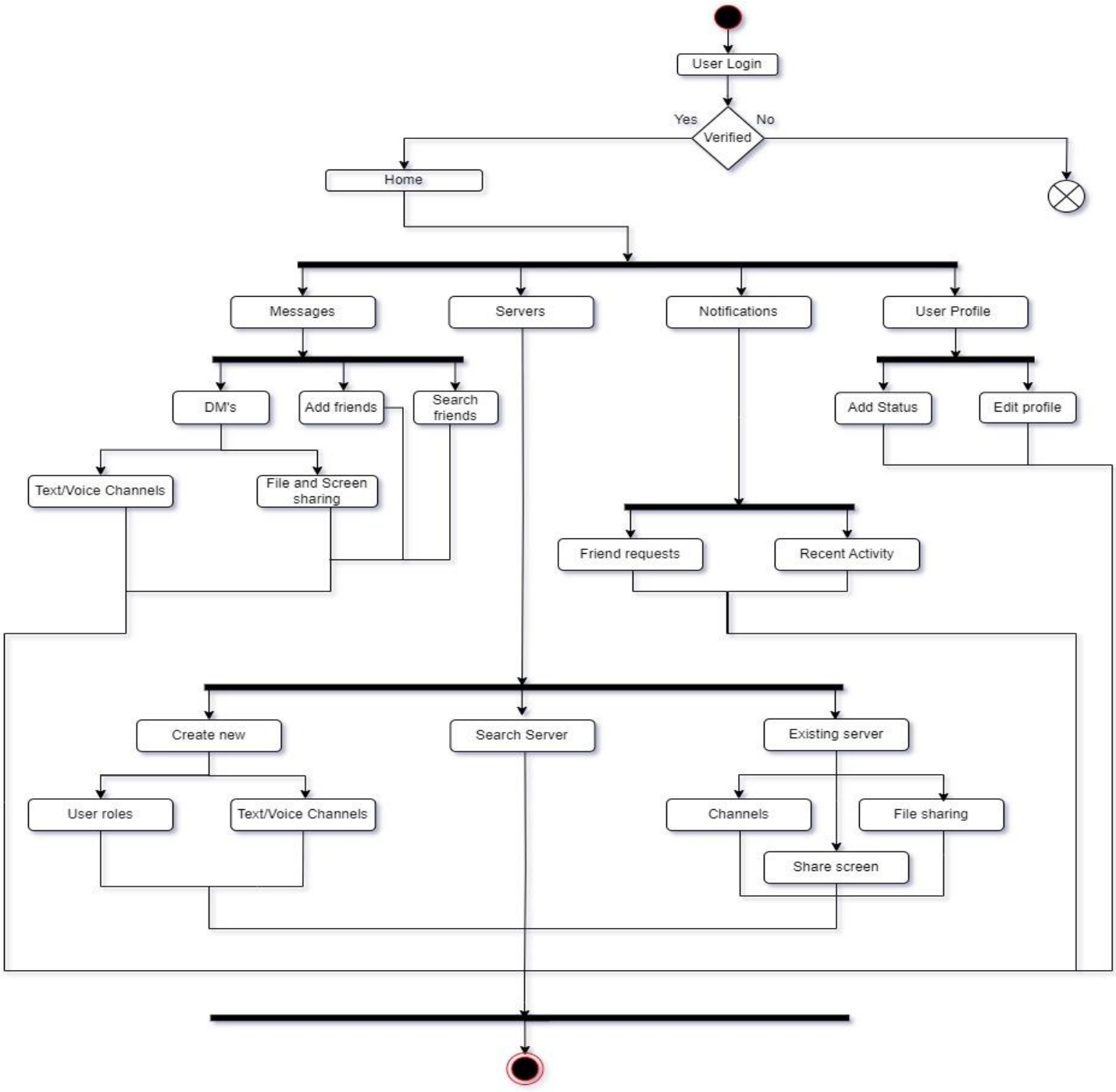
Constraint Requirements Description

Constraint Req. ID #	Function Name	Constraints Requirement Description	Software Requirement
CON1	Stakeholder constraints	Maintains the user demands	17
CON2	Security constraints	Encrypt data & secure communication	18
CON3	Resources constraints	Satisfying users for Servers & hosting	19

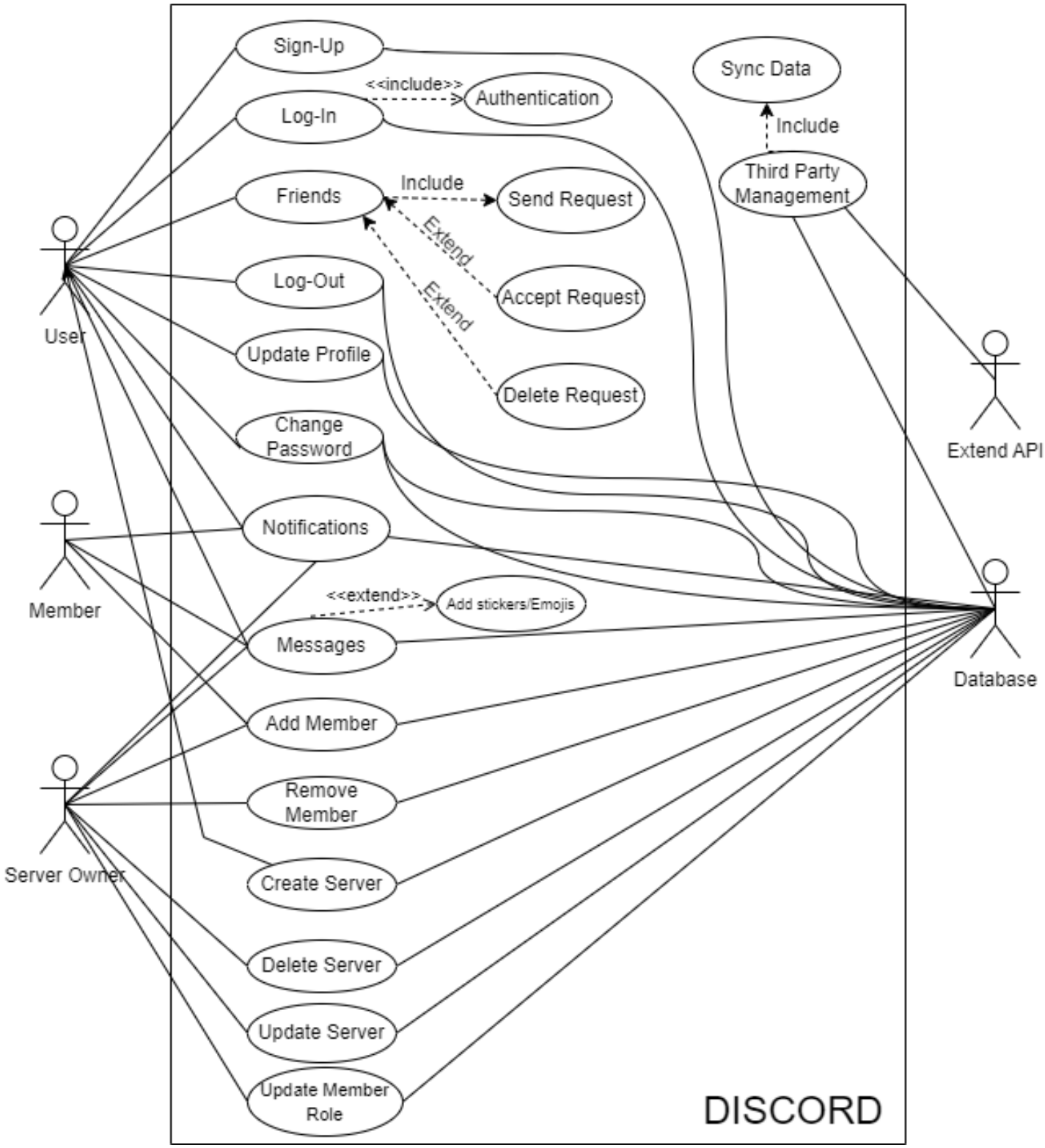
Domain Requirements Description

Domain Req. ID #	Function Name	Domain Requirement Description	Software Requirement
DOM1	Content moderation	Manages unwanted materials	20
DOM2	Real time chats	Ensure real time chats have low latency	21b

1. ACTIVITY DIAGRAM



USE CASE Model
(2.1) Diagram



(2.2) USE-CASE DESCRIPTIONS

Use Case	Signup
Description	This use case starts when user wants to create a new account on the discord app
Primary Actor	User
Goals	The user is able to register a new account on the discord app.
Pre-condition	The user must have an email id to create account. The user must have a unique user name.
Post-condition	A new account has been registered on the database of discord. Users Email address has been verified.
Trigger	The user clicks on the signup button to register a new account
Scenario	The users opens the discord app and clicks on the signup button and then register an account by entering the details and then verify his email address.

Use Case	Login
Description	This use case starts when user wants to login to his/her account on the discord app
Primary Actor	User
Goals	The user is able to login to his/her account on the discord app. User enters username and password to login.
Pre-condition	The user enter his email id and password to login.
Post-condition	The user interacts with the discord GUI after signing in to his/her account
Trigger	The user clicks on the Login button to register a new account.
Scenario	The users opens the discord app and clicks on the Login button and then enter his username and password to login to his/her account.

Use Case	Friends
Description	This use case starts when user wants to add friends to his/her account on the discord app
Primary Actor	User
Goals	The user is able to add friends to his/her account on the discord app by finding their friends from their username.
Pre-condition	The user enters his friend’s username to search his/her account. User sends request to his friend.
Post-condition	The request is sent to his/her friend.
Trigger	The user clicks on the add friend button to add new friend.
Scenario	The user clicks on the add friend button and then enter his/her friends username to send request.

Use Case	Messages
Description	This use case starts when user wants to send a message.
Primary Actor	User
Goals	The user clicks on the messages button and then send a message to his/her friend and also check messages from there.
Pre-condition	The user must have a friend to message.
Post-condition	User can check the timestamp to ensure that the message was sent successfully.
Trigger	The user clicks on the messages button to message a friend.
Scenario	The user clicks on the messages button and can send a message from their also check the incoming messages

Use Case	Notifications
Description	User can check the new notifications from notification button.
Primary Actor	User
Goals	The user clicks on the notifications button and then check for new notifications like requests etc.
Pre-condition	The discord is connected to the server to access the designated notification
Post-condition	User can view notifications from the notification tab button
Trigger	The user clicks on the notification button to see new notifications.
Scenario	The user clicks on the notifications button and can see for new notification like friend requests server join requests or message requests etc.

Use Case	Create Server
Description	User can create server by setting a server name and adding channels in it.
Primary Actor	User
Goals	The goal is to create a server on the discord app for meetings and announcements.
Pre-condition	User must have a valid discord account
Post-condition	Notification arrives that a server created successfully.
Trigger	The user clicks on the create server button to create a new server.
Scenario	The user clicks on the create server button and then set a server name and other necessary things to create a server.

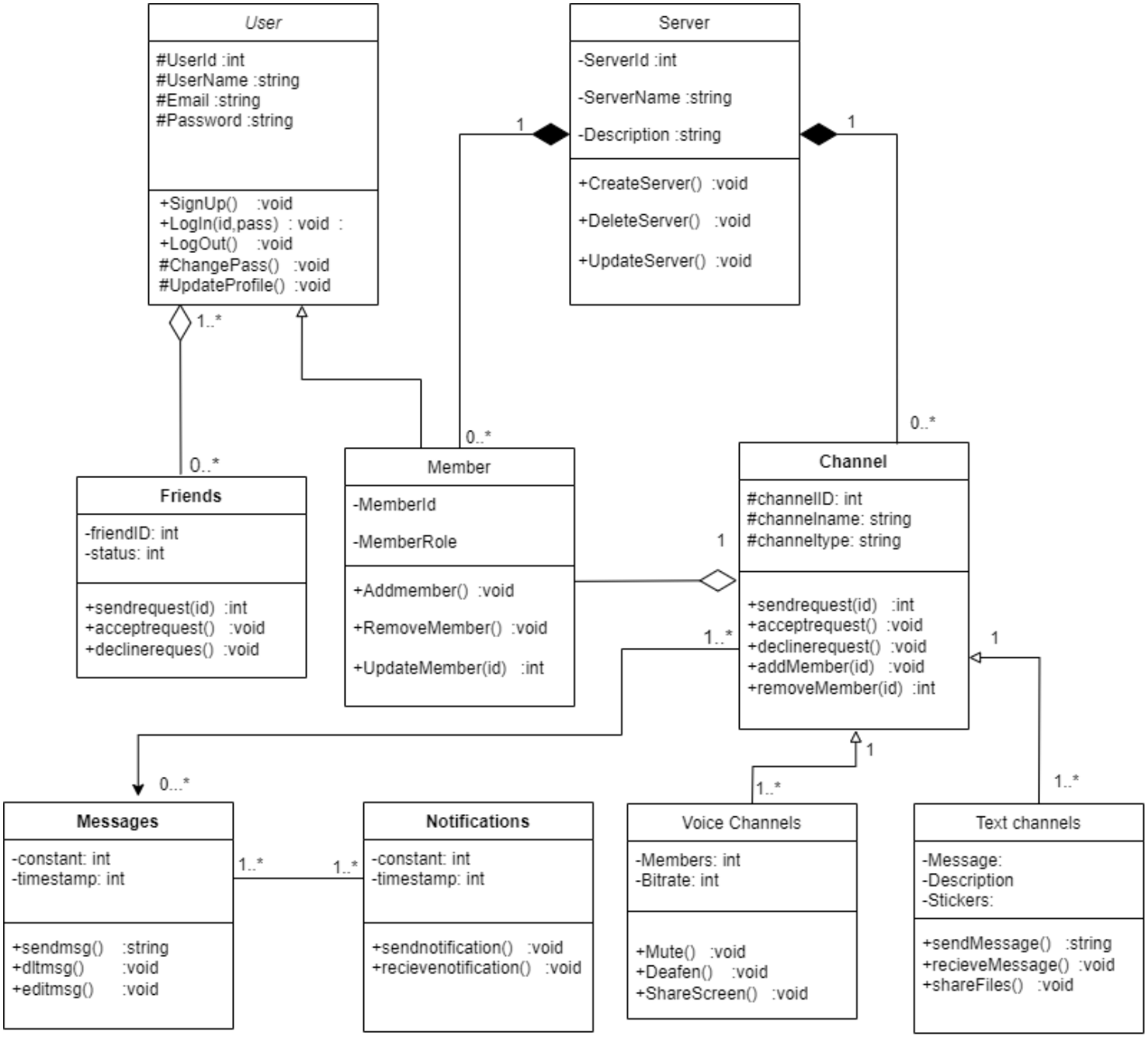
Use Case	Delete Server
Description	Owner can delete server by selecting the server which he wants to delete.
Primary Actor	Server owner
Goals	The goal is to create a server on the discord app for meetings and announcements.
Pre-condition	User must have a valid discord account
Post-condition	Notification arrives that a server created successfully.
Trigger	The user clicks on the create server button to create a new server.
Scenario	The user clicks on the create server button and then set a server name and other necessary things to create a server.

Use Case	Update Server
Description	Owner can update server by selecting the server which he wants to update.
Primary Actor	Server owner
Goals	The goal is to update server on the discord app for customizations.
Pre-condition	Owner must login to his/her account to update the server.
Post-condition	Notification arrives that a server updated successfully.
Trigger	The user clicks on the update server button to update server.
Scenario	The server owner clicks on the update server button and then customize the server according to the needs.

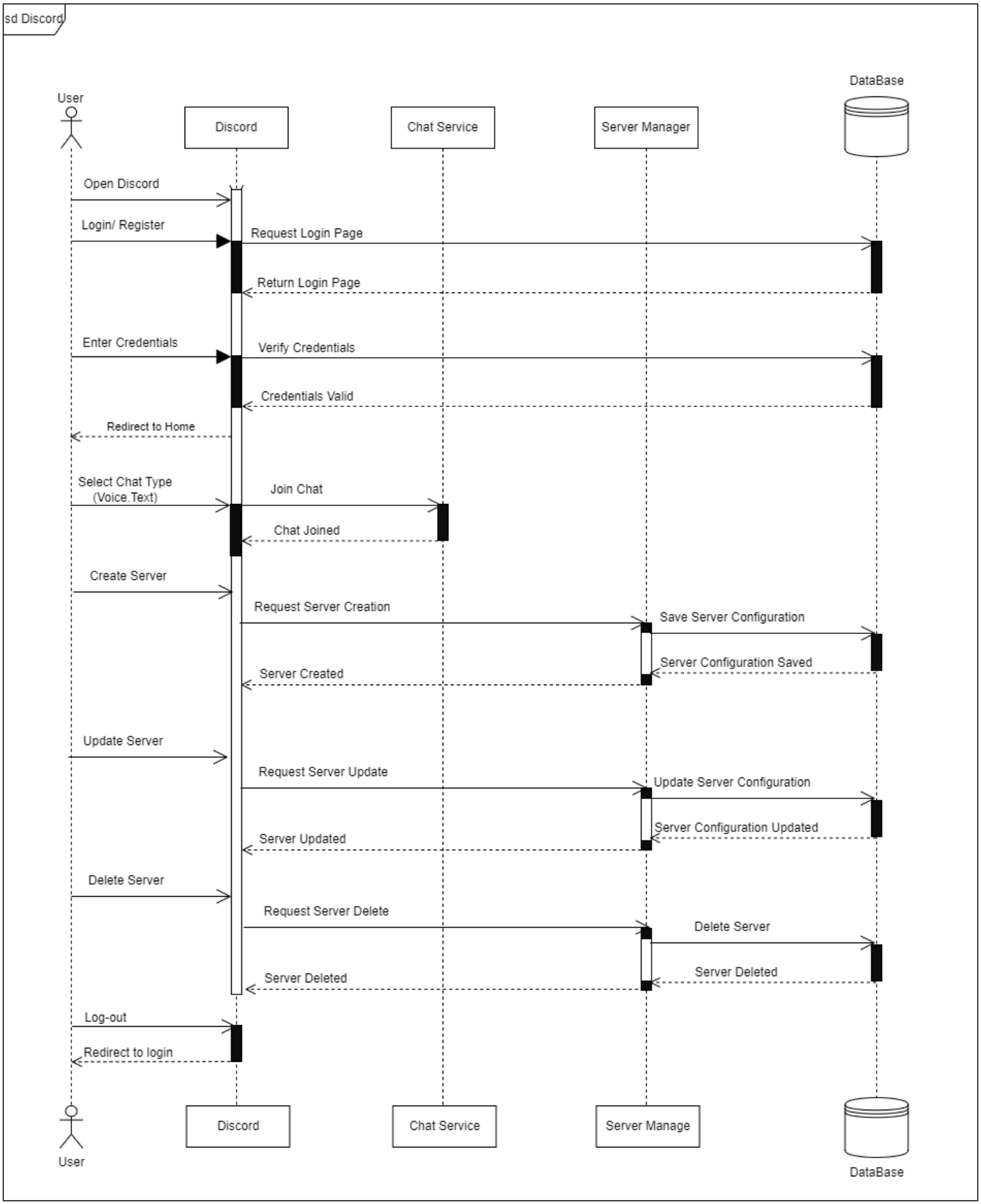
Use Case	Add member
Description	Admin can add new members to his/her server
Primary Actor	Server owner
Goals	Server admin can add his friends to his server and also he/she can share a link with his friends so that they can join his/her server
Pre-condition	Admin can add his/her friends by their username.
Post-condition	The notification arrives that the new member is added successfully.
Trigger	The server owner clicks on the add member button to add new members to his server.
Scenario	The server owner send server request to friends so that they can join the server.

Use Case	Remove member
Description	Admin can remove members from his/her server
Primary Actor	Server owner
Goals	Server admin can remove users from his server by selecting the user and remove him from the server.
Pre-condition	Admin can remove user from the server by selecting the user and remove him/her from server.
Post-condition	The notification arrives that the member is removed successfully.
Trigger	The server owner clicks on the remove member button to remove members from the server.
Scenario	The server owner selects the user which he wants to delete from the server and then remove the member. After removing notifications arrives to ensure that the member is removed successfully.

CLASS DIAGRAM



SEQUENCE DIAGRAM



ARCHITECTURE DIAGRAM