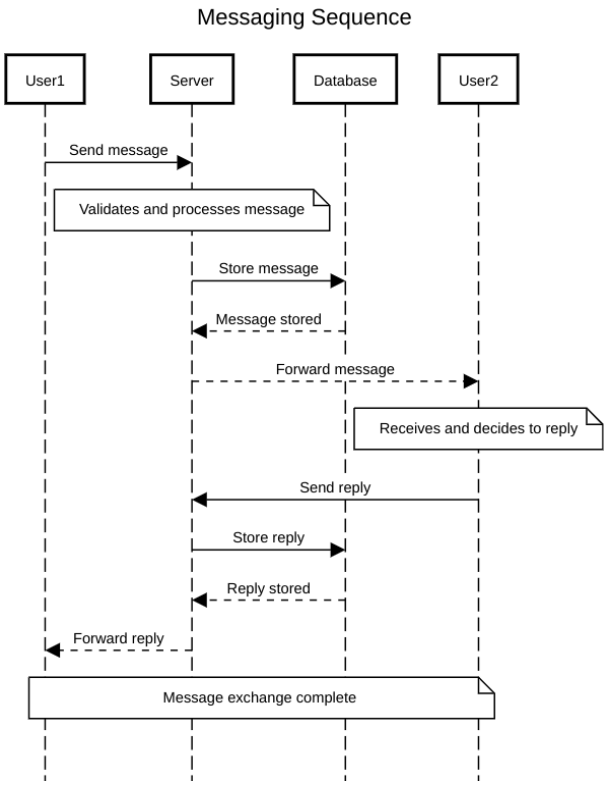
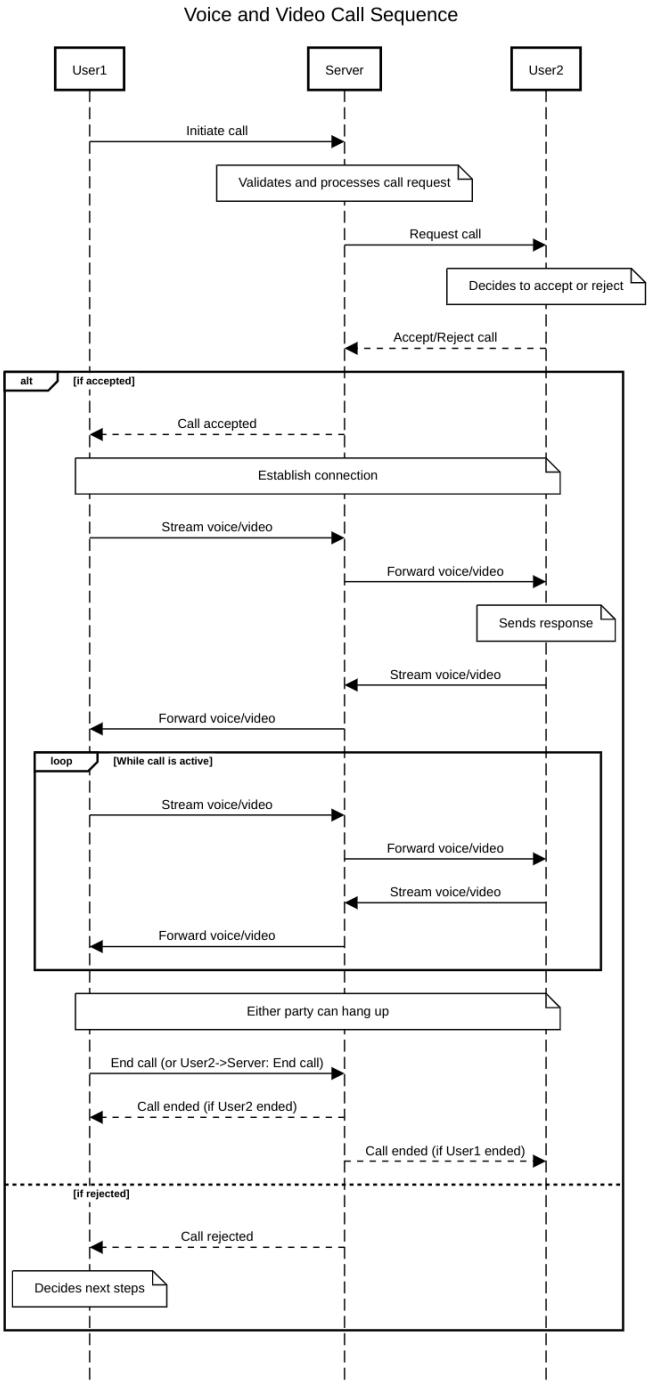
RISCORD MILESTONE 3

Jackson Wilson, Soumil Chhabra, Ronak Jain, Chinmay Arvind, Anitej Isaac Sharma

Sequence Diagrams For 2 Main Features





Testing Plan

As of right now our testing plan mainly consists of unit testing, and while it serves its purpose, there is room for improvement. It serves as a very rudimentary form of testing that allows us to test our code at a pretty basic level. While this works on a base level, an expansion of our testing plan should improve both the efficiency and quality of our produced code.

Our plan moving forward is to incorporate user testing into our testing plan, having participants unfamiliar with the code run through it in some regulated testing process in order to test its efficiency. These participants can be either group members that did not contribute to that specific section of code, or alternatively someone unfamiliar to the project as a whole. This is to remove bias from how the code is tested, as a contributor may either knowingly or unknowingly stay away from certain elements so as to not run into an issue. The users testing the code will do so by documenting their process. Ie: to what extent did they test the code, what edge cases did they test, errors or lack thereof encountered, etc. The unbiased testing combined with the documentation should provide whoever is currently working on that block of code with ample feedback to modify, change, and improve the code accordingly.

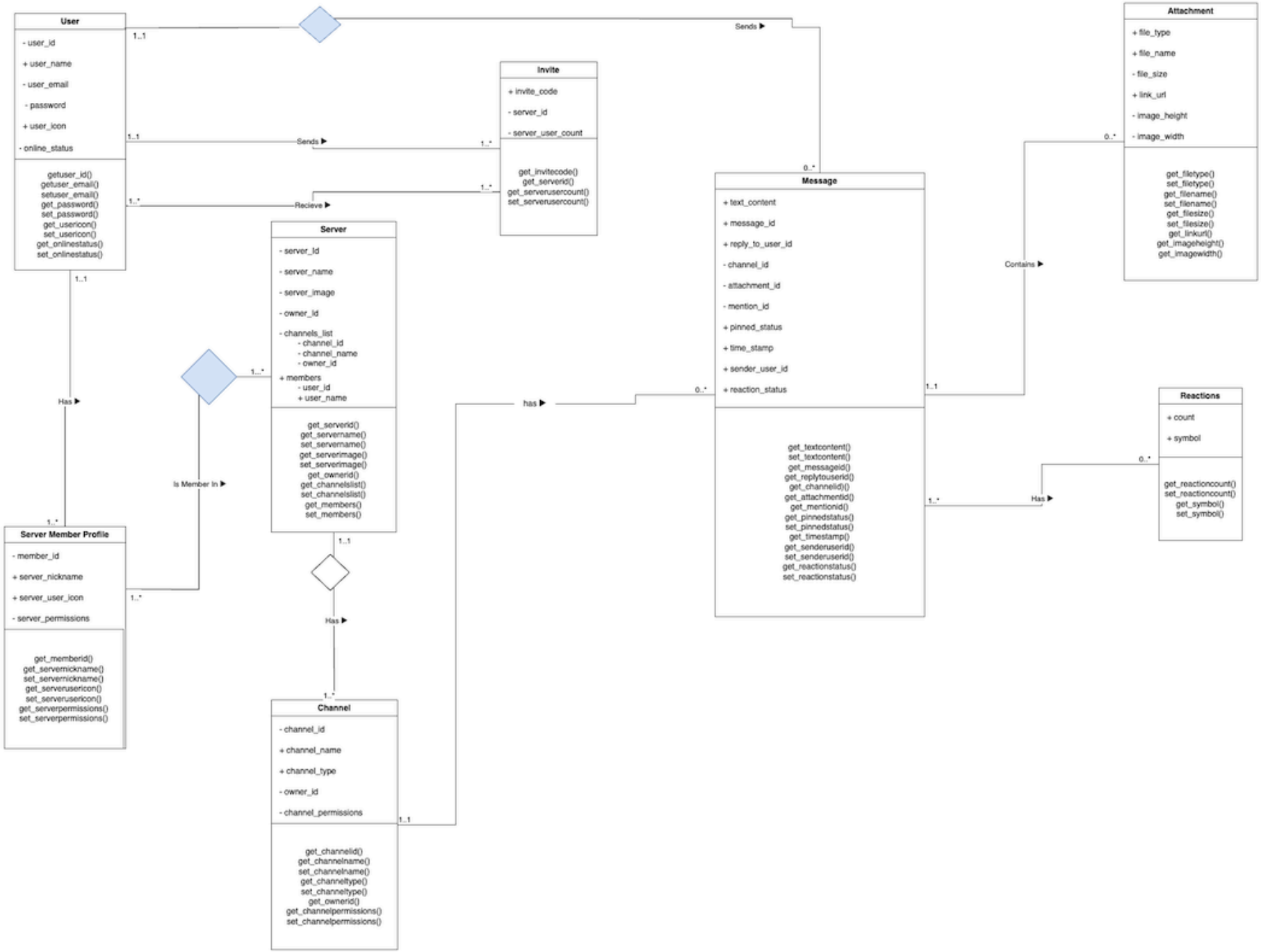
Design Principles

The first design principle we plan to use is Model-View Controller (MVC) as its general use case relates pretty well to our plans for the app. The idea behind MVC is to separate the application's data, UI, and control logic into separate components. Including this design principle in our development would allow us to develop, test, and maintain our software in a way that makes sense. MVC is already a commonly used design principle in desktop and web applications as well, so incorporating it into ours would make the most sense.

The second design principle is Protocol Stack. Being a user to user communication app, a design principle such as this makes sense as there will be multiple layers to the communication inherently. The primary layer will obviously be the text sent between users, and will be the only visible layer from the UI. Below that will be things like metadata, user information, sender and receiver addresses, etc.

Class Diagram

It is on the following page & here is the link to the diagram:https://app.diagrams.net/#G1x1uRZgq5yxdQTKXVvxRhHal_rw2wmKNN


User
- user_id
+ user_name
- user_email
- password
+ user_icon
- online_status
getuser_id()
getuser_email()
setuser_email()
get_password()
set_password()
get_usericon()
set_usericon()
get_onlinestatus()
set_onlinestatus()
Sends
Invite
+ invite_code
- server_id
- server_user_count
get_invitecode()
get_serverid()
get_serverusercount()
set_serverusercount()
Recieve
Server
- server_Id
- server_name
- server_image
- owner_id
- channels_list
- channel_id
- channel_name
- owner_id
+ members
- user_id
+ user_name
get_serverid()
get_servername()
set_servername()
get_serverimage()
set_serverimage()
get_ownerid()
get_channelslist()
set_channelslist()
get_members()
set_members()
Has
Is Member In
Server Member Profile
- member_id
+ server_nickname
+ server_user_icon
- server_permissions
get_memberid()
get_servernickname()
set_servernickname()
get_serverusericon()
set_serverusericon()
get_serverpermissions()
set_serverpermissions()
Channel
- channel_id
+ channel_name
+ channel_type
- owner_id
- channel_permissions
get_channelid()
get_channelname()
set_channelname()
get_channeltype()
set_channeltype()
get_ownerid()
get_channelpermissions()
set_channelpermissions()
has
Message
+ text_content
+ message_id
+ reply_to_user_id
- channel_id
- attachment_id
- mention_id
+ pinned_status
+ time_stamp
+ sender_user_id
+ reaction_status
get_textcontent()
set_textcontent()
get_messageid()
get_replytouserid()
get_channelid()()
get_attachmentid()
get_mentionid()
get_pinnedstatus()
set_pinnedstatus()
get_timestamp()
get_senderuserid()
set_senderuserid()
get_reactionstatus()
set_reactionstatus()
Contains
Attachment
+ file_type
+ file_name
- file_size
+ link_url
- image_height
- image_width
get_filetype()
set_filetype()
get_filename()
set_filename()
get_filesize()
set_filesize()
get_linkurl()
get_imageheight()
get_imagewidth()
Has
Reactions
+ count
+ symbol
get_reactioncount()
set_reactioncount()
get_symbol()
set_symbol()