Planning

1. Requirement Gathering
    a. Functional
    b. Non-functional
2. Component Architecture
3. High level APIs and Protocol
4. Data Entities
5. Data Store / Frontend Store
6. Performance Optimization
7. Accesbillity

Function Requirements

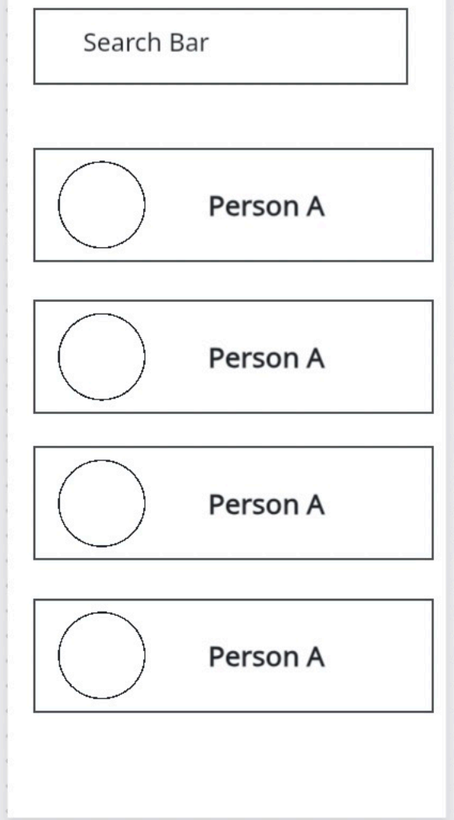
1. User Can send the Message(40-50ms)
2. User can receive the message
3. User can attache the attachments
    a. Audio
    b. video
    c. pictures
    d. location
4. User can see Contact list
5. User can invite to other

Non-Functional requirements

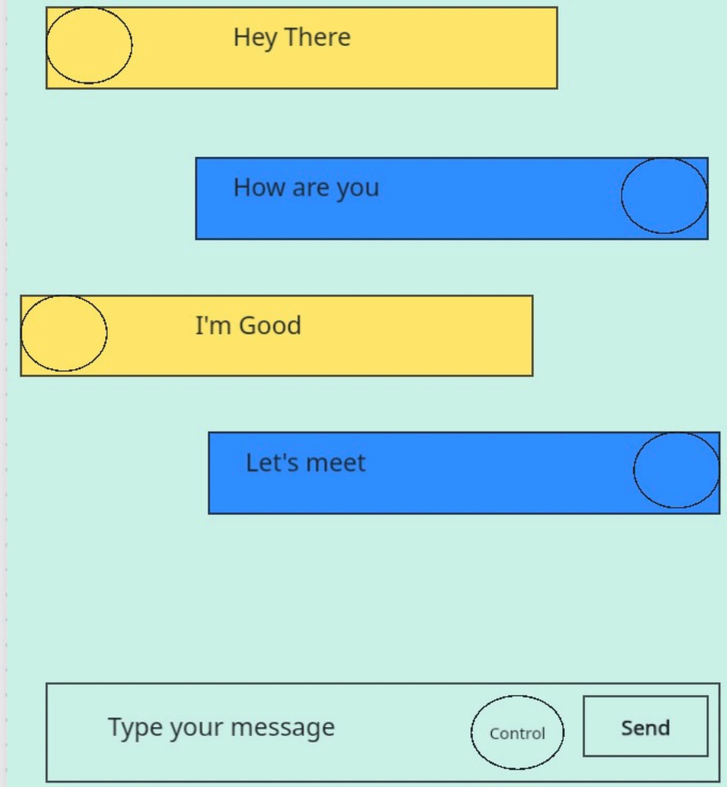
1. It should be support by variety of devices (Tab/Desktop/Phone)
2. It should be accssible at remote area as well
3. It should support Multi lagnguage
4. Sending / Receiving has low lattencey
5. It should be secure
6. Responsive
7. Offline Support
8. Web vitals
9. Logging and Monitoring

Component Architecture

Canvas 1

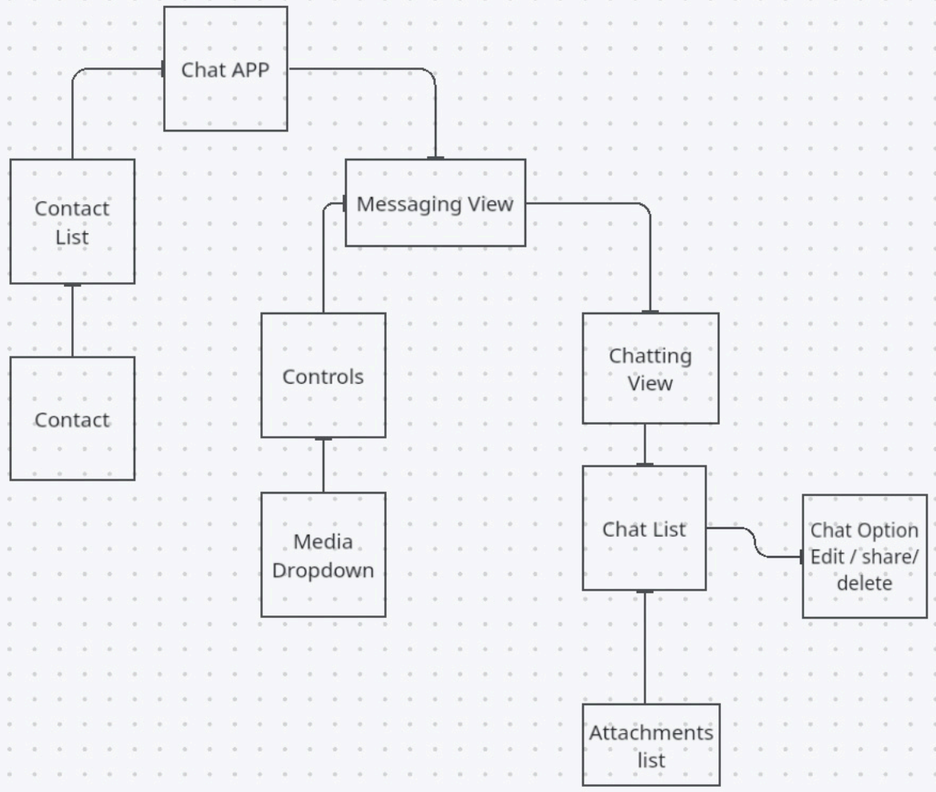


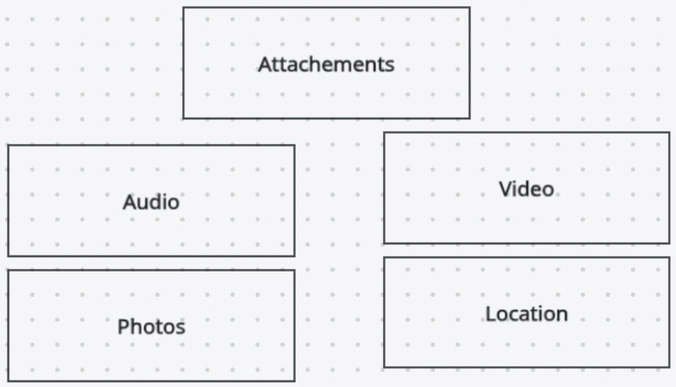
Canvas 2

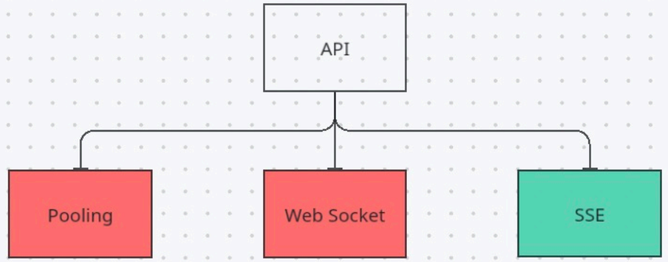


When the user will click on any message , he can see all four options

Component Hierechy







1. HTTP benifits
2. Easy to implement
3. Un-necessay making server call
4. Client -> server

CONS

1. Not scalable
2. Increase the network load, because every time we have to send all the details, header, token, etc.

1. ws:// or wss://
    a. ws-> websocket protocl
    b. wss -> websockt secure protocle
2. Websocks are fast
3. Bi-drectional communtical
4. Client <----> Server

CONS

Keeping connection open is expensive
Hard to load balance
Proxy and Firewalls block websocket connection

1. HTTP prototcols
2. Easy to scale
3. Client <---- Server
4. Easy to laod balance
5. Auto-matic reconnect if connections drop

CONS

Server only send data in text format
We can 't close the connection from client side
It is unidirectional

`List<Contact> getContactList(token)`

`sendingMessage(token, user_id, message)`

`addAttachments(token, user_id, binary_data)`

`deleteChat(token, chat_id)`

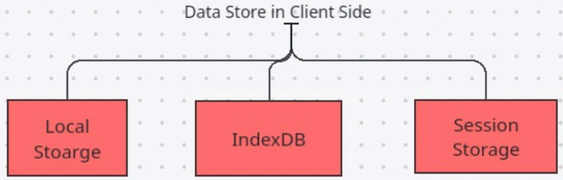
`editMessage(token, user_id, chat_id, message)`

`deleteMessage(token, user_id, chat_id)`

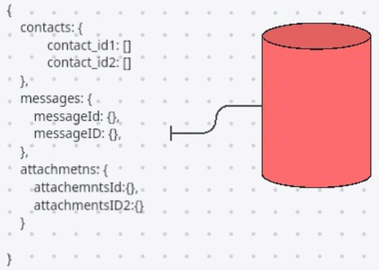
`List<Chats> getAllchats(token, user_id)`

```
Type Contact =  {
   id: String
   avatar_link: String
   name: String
}
```

```
Type Chats =  {
   id: String
   message: String
   date: DateTime
   receiverId: String
   attachments: [Attachments]
}
```



Stroring after normalization



```
{
  contacts: {
       contact_id1: []
       contact_id2: []
  },
  messages: {
     messageId: {},
     messageID: {},
  },
  attachmetns: {
     attachemntsId:{},
     attachmentsID2:{}
  }

}
```