

# VotaFun

Alphaeus  
Siddiq  
Ryan  
Lloyd  
Roy  
Charles

# Table of Contents

01

Project Introduction

02

Design & Maintainability

03

Frontend Design

04

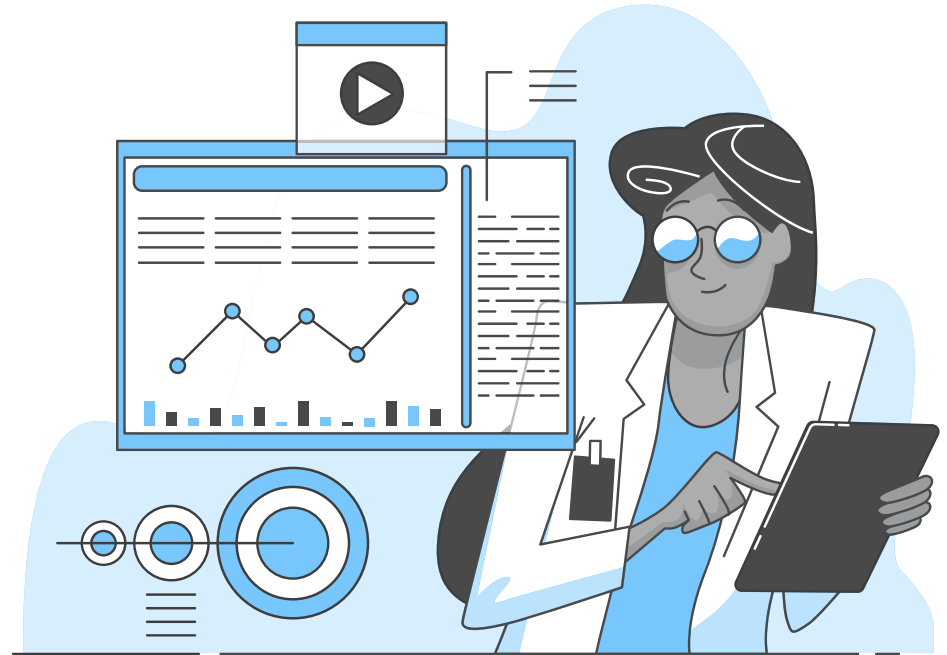
Backend Design

05

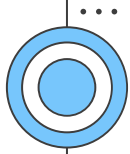
VotaFun Demonstration

06

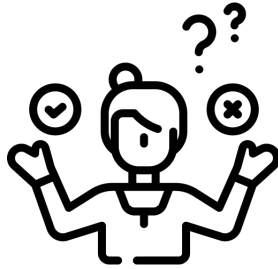
VotaFun Releases



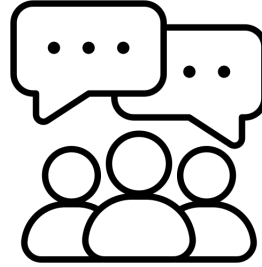
# Project Introduction



# Project Description

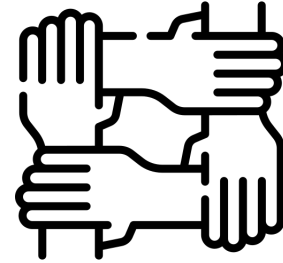


- Groups of friends may struggle to agree on a common activity
- Why? – Differing preferences and opinions



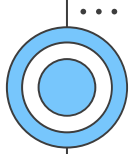
- Lengthy discussion
- Inefficient use of time

...

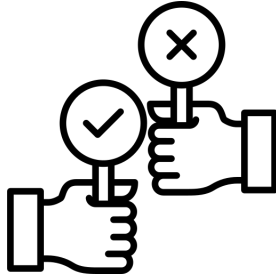


- A solution that is efficient for planning
- Maintain the spirit of unity





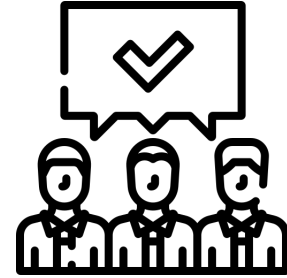
# Project Objectives



- Enhance decision making efficiency

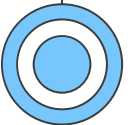


- Simplifying communication



- Collaboration in promoting consensus

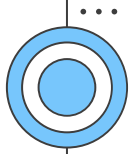
...



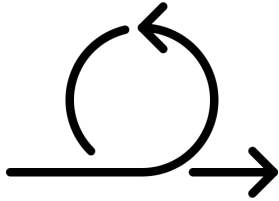
# Proposed solution

Vota Fun

...



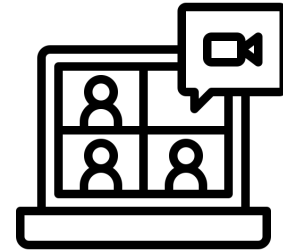
# Software methodology



- Agile software development methodology



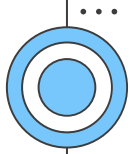
- Sprint cycles of 1 week



- Weekly standup meetings

...





# Team Members

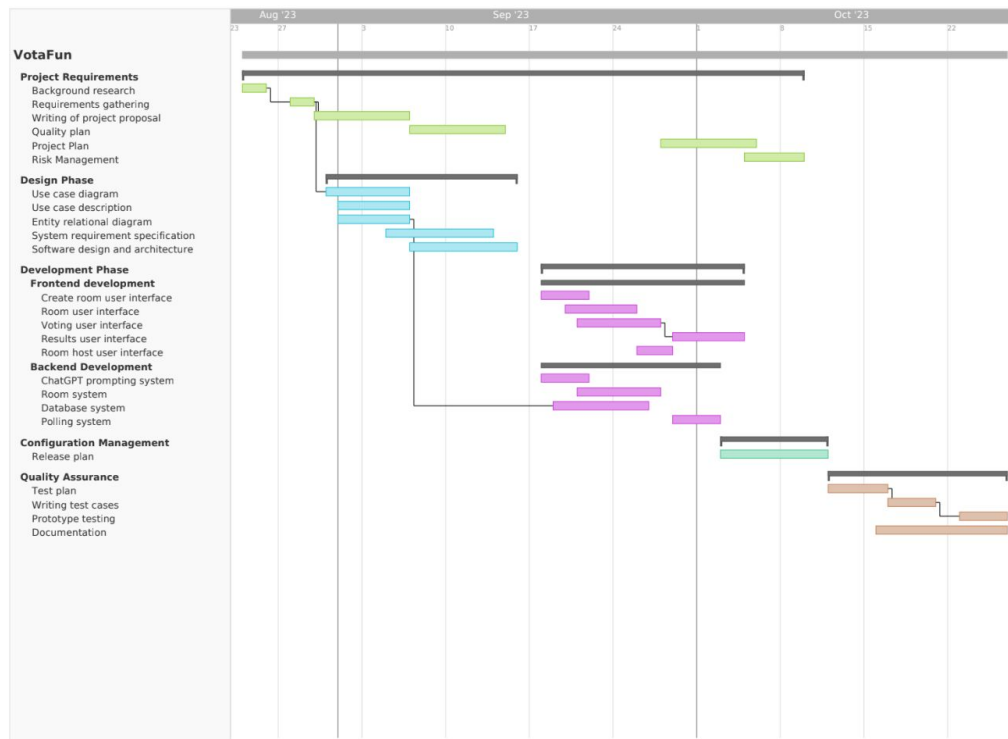
- **Project Manager:** Alphaeus
- **Lead Front-end Developer:** Siddiq
- **Front-end Developer:** Ryan, Lloyd
- **Lead Back-end Developer:** Roy
- **Back-end Developer:** Alphaeus, Charles
- **Quality Assurance Engineer / Manager:** Charles, Siddiq
- **Release Engineer / Manager:** Lloyd



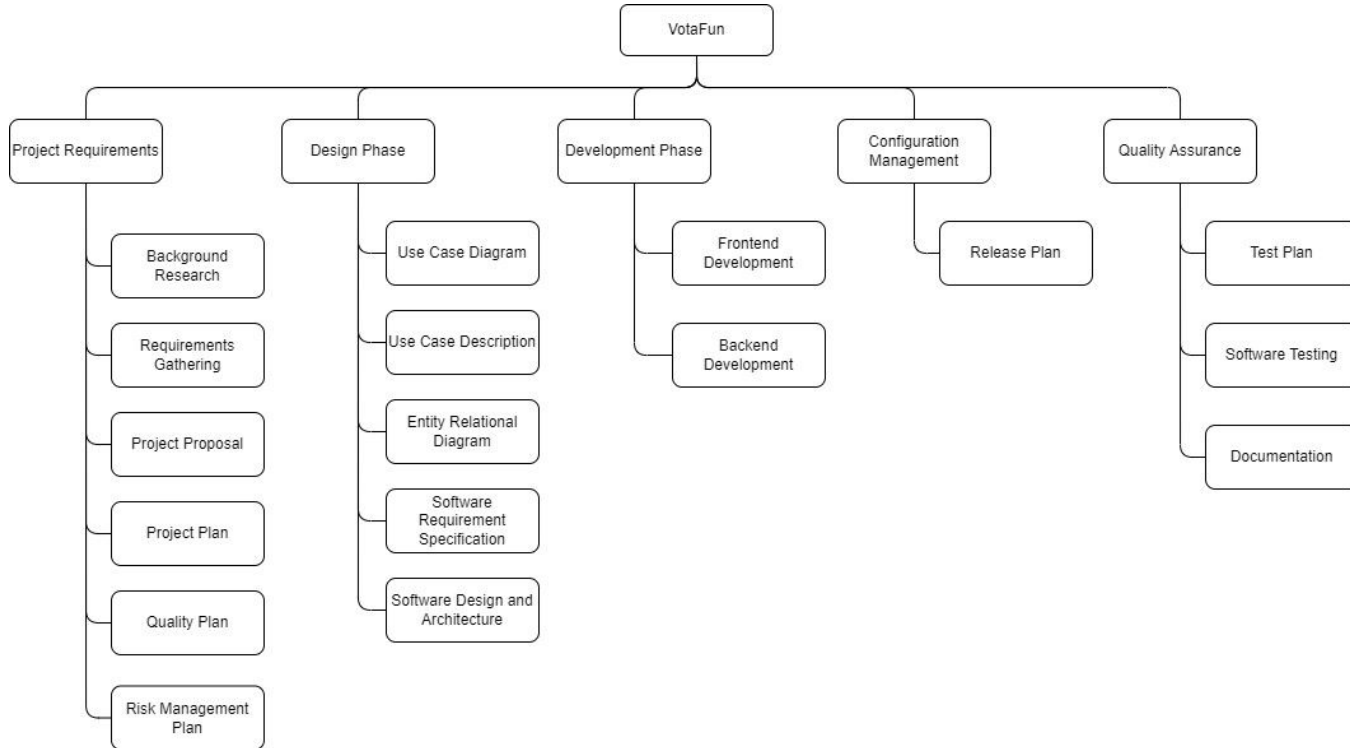


# Project Schedule

teamgantt  
Created with Free Edition



# Project Work Breakdown



# Project Efforts, Duration and Team Size Estimation

Element	Complexity	Detail
Inputs	Medium	Vote for an option
	Low	Kick a user
	Medium	Joining a room by entering the room code/ link
	Medium	Create a room, setting the room location and activity
Outputs	High	Generates question and options, or activities for users
	Medium	Display questions and options
	Low	Display the final activity
	Medium	Alert users if they are kicked
	Medium	Alert users if the room data changes
	Medium	Display room location and activity

# Project Efforts, Duration and Team Size Estimation

Element	Complexity	Detail
Inquiries	Medium	View room information (Number of people, Current host)
Logical Files	Medium	Add or kick a user in a room, and update the database
	Low	Modify room status and update the database
	Medium	Stores generated questions and options or activities
	Medium	Update the votes in the database when a user votes for an option
Interfaces	-	-

...

# Project Efforts, Duration and Team Size Estimation

Characteristic	Low	Medium	High
Inputs	$1 \times 3 = 3$	$3 \times 4 = 12$	$0 \times 6 = 0$
Outputs	$1 \times 4 = 4$	$4 \times 5 = 20$	$1 \times 7 = 7$
Inquiries	$0 \times 3 = 0$	$1 \times 4 = 4$	$0 \times 6 = 0$
Logical files	$1 \times 7 = 7$	$3 \times 10 = 30$	$0 \times 15 = 0$
Interfaces	$0 \times 5 = 0$	$0 \times 7 = 0$	$0 \times 10 = 0$
Unadjusted Function Points (FP)	14	66	7
Total = L + M + H	$14 + 66 + 7 = 87$		

...

# Project Efforts, Duration and Team Size Estimation

**Total Score of Influence Factors** = 43

**Influence Multiplier** = Total Score  $\times$  0.01 + 0.65 =  $43 \times 0.01 + 0.65 = 1.08$

**Adjusted FP** = Unadjusted FP  $\times$  Influence Multiplier =  $87 \times 1.08 = 93.96$

**LOC** =  $35.64 \text{ FP} \times 53 \text{ LOC/FP} + 58.32 \text{ FP} \times 24 \text{ LOC/FP} = 3288.6 \text{ LOC}$

**Effort** = Size / Production Rate =  $(3288.6 \text{ LOC}) / (39 \text{ LOC/PD}) = 85 \text{ PD}$  (rounded up to the nearest integer)

**Duration** =  $3 \times (\text{Effort})^{1/3} = 3 \times (85)^{1/3} = 14 \text{ Days}$  (rounded up to the nearest integer)

**Initial schedule** = 14 Days / 5 days a week = 3 Weeks (rounded up to the nearest integer)

**Team size** =  $85 \text{ PD} / 14 \text{ D} = 6.07 \text{ P} = 7 \text{ Persons}$  (rounded up to the nearest integer)

**Total person-hours (PH)** =  $85 \text{ PD} \times 8 \text{ hours} = 680 \text{ PH}$  (~ 3 weeks for implementation)

...

# Project Budget


Item	Supplier	Quantity	Unit Price (SGD)	Cost per month (SGD)	Total (3 months)
Project Manager	-	1	-	\$7,000.00	\$21,000.00
Software Developers	-	3	-	\$5,500.00	\$49,500.00
QA/Release Engineers	-	3	-	\$5,000.00	\$45,000.00
Laptops	Dell	6	\$1,090.00	-	\$5,540.00
Printer	Fuji Xerox	1	\$450.00	-	\$450.00
Technology Licence	Visual Paradigm	1	\$99.00	-	\$99.00
Cloud Services	Heroku	1	-	\$300.00	\$900.00
Office rental	NTU	1	-	\$5,000.00	\$15,000.00
			...	<b>TOTAL</b>	<b>\$137,489.00</b>

Medium


# Risk Management

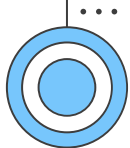
Identified Risk	Risk description	Response	Overall risk level	Probability of occurrence	Impact level
RISK 01	More changes to requirements than anticipated	Mitigate ▼		Low ▼	High ▼
RISK 02	Launch of new large language models (LLM)	Mitigate ▼		Low ▼	Medium ▼
RISK 03	Underestimation of project size	Mitigate ▼		Low ▼	High ▼
RISK 04	Project running behind schedule	Mitigate ▼		Low ▼	High ▼
RISK 05	Disagreements within the project team	Mitigate ▼		Medium ▼	Medium ▼
RISK 06	Key personnel leaving the team	Mitigate ▼		Low ▼	Medium ▼
RISK 07	Underestimation of budget needed for the project	Mitigate ▼		Low ▼	High ▼
RISK 08	Dependency on 3rd party APIs	Mitigate ▼		Low ▼	High ▼
RISK 09	Market changes resulting in less uptake	Mitigate ▼		Low ▼	High ▼





# Design & Maintainability





# Maintainability Practices

## 1. Modular Design

- a. Multiple small components (Sockets, Database, User interface)

## 2. Clean Code and Coding Standards

- a. Pre-commit hooks (Lintstaged, husky, black, flake8)
- b. Testing with Pytest (Heroku CICD)

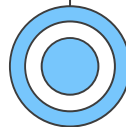
## 3. Documentation

- a. Design plans drafted before implementation
- b. Constantly updated to reflect changes

...

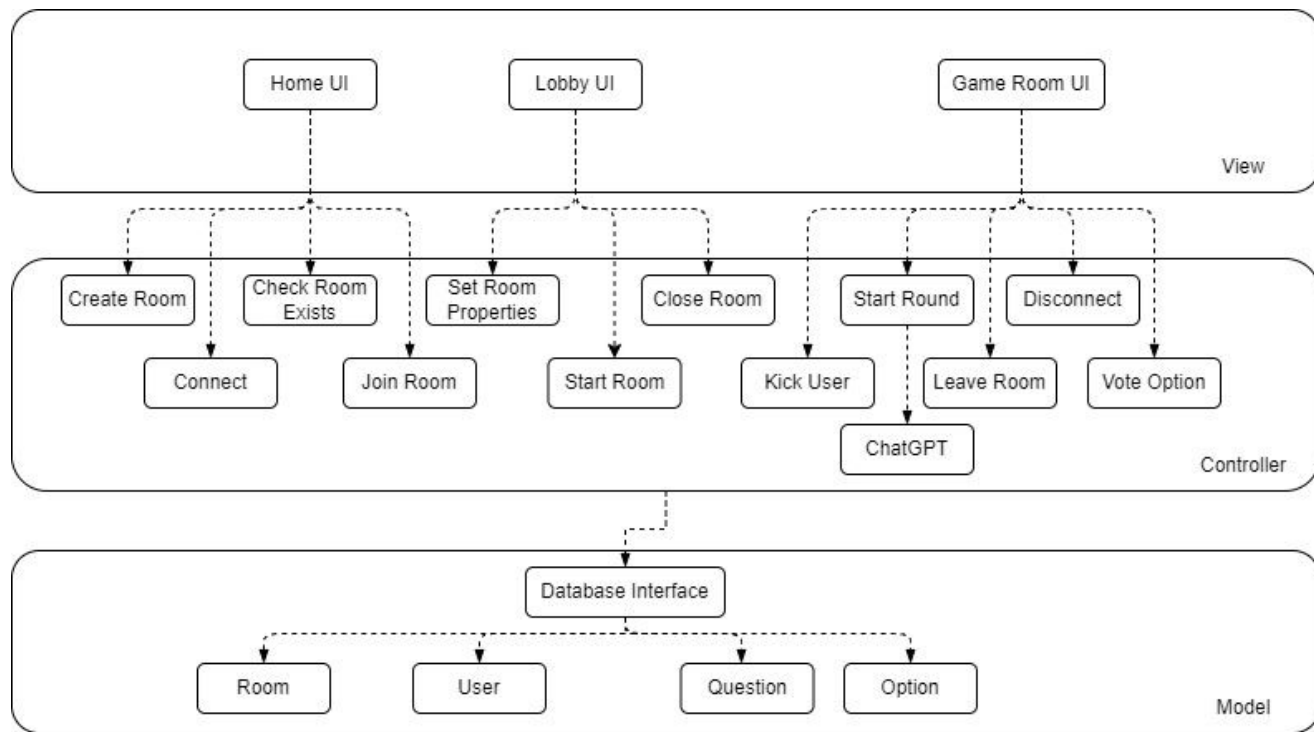


...



...

# System Architecture



# Configuration management tools



**MediaWiki**



**GitHub**



Google Drive

...

# Frontend Design

# Frontend frameworks

# NEXT.js



**tailwindcss**

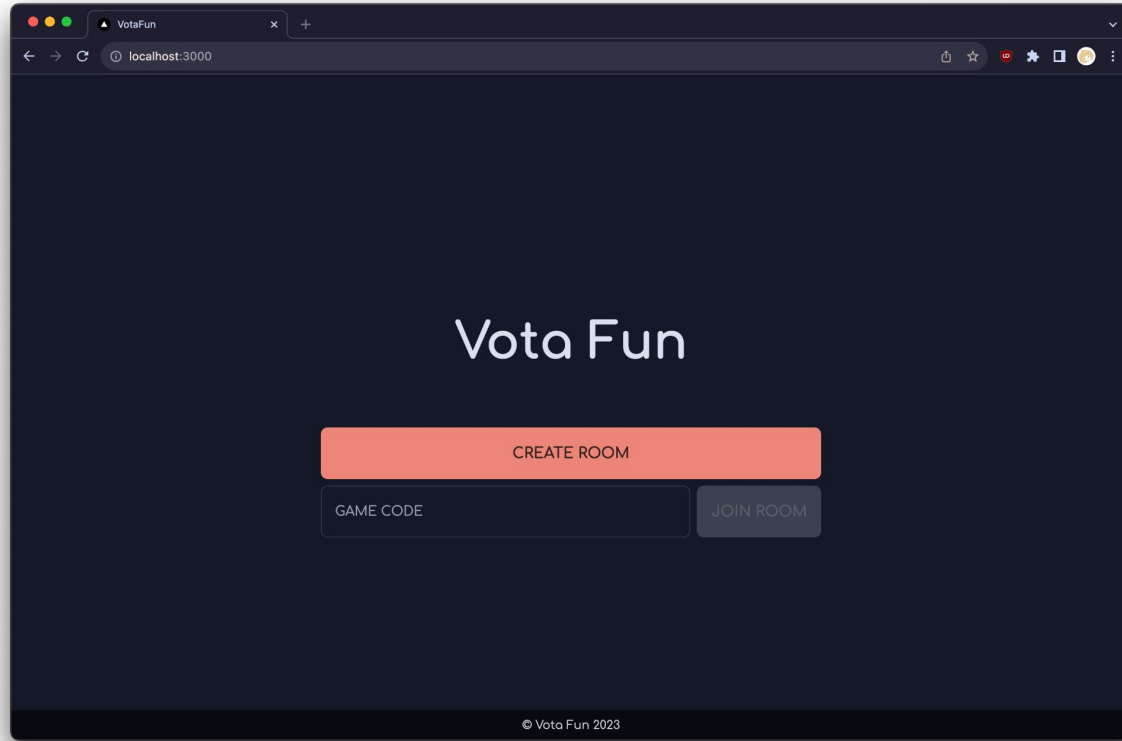


**Motion**

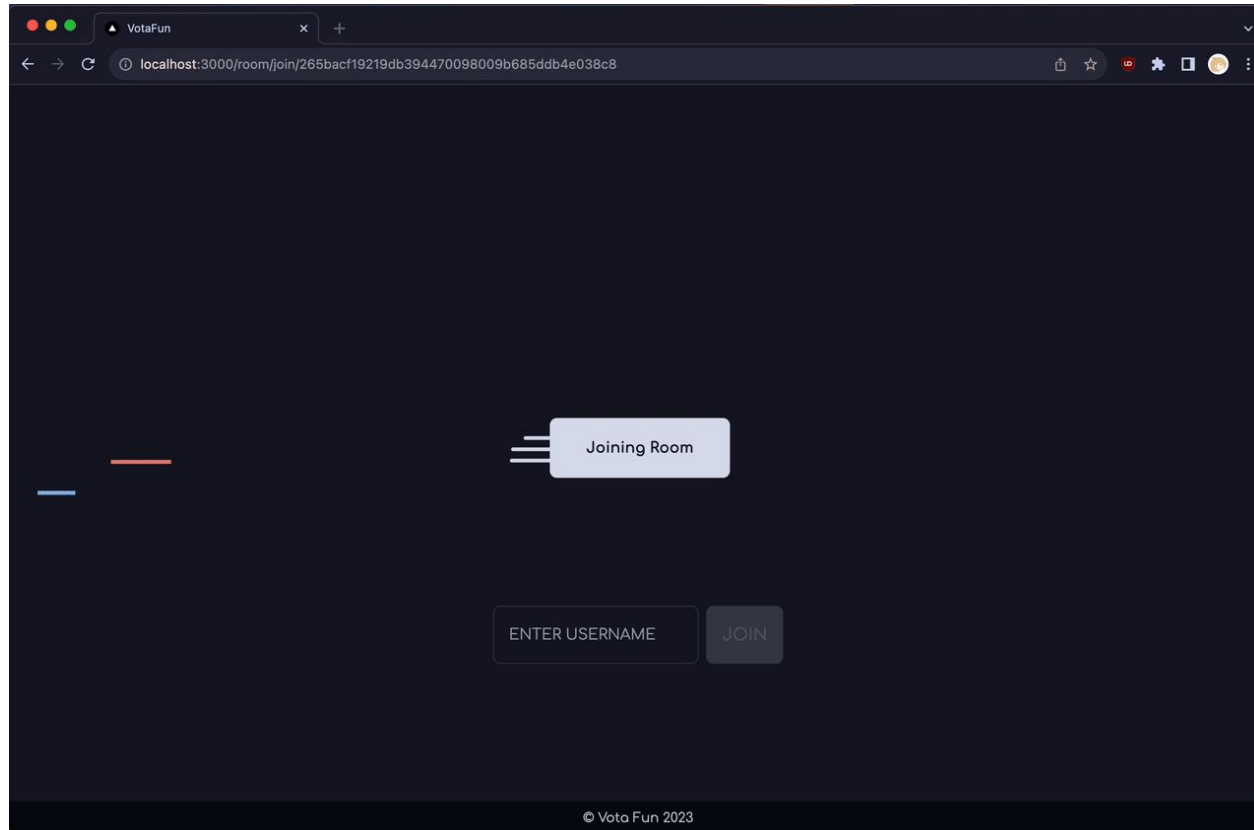


**daisyUI**

# User Interface

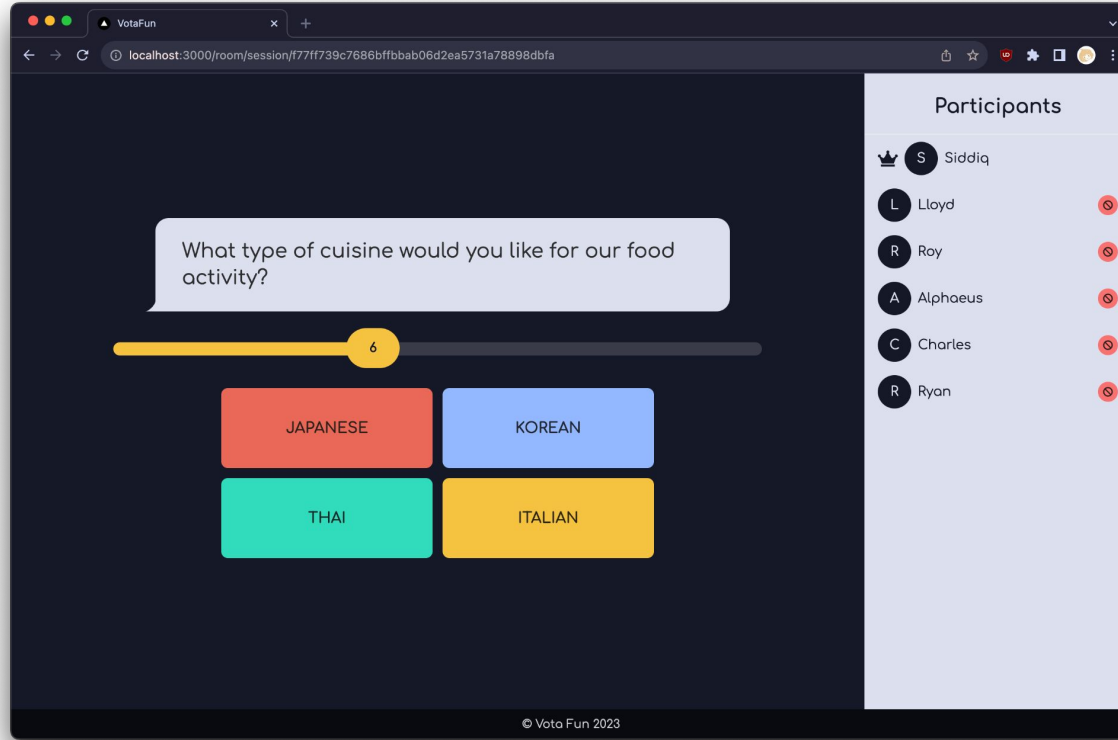


# User Interface

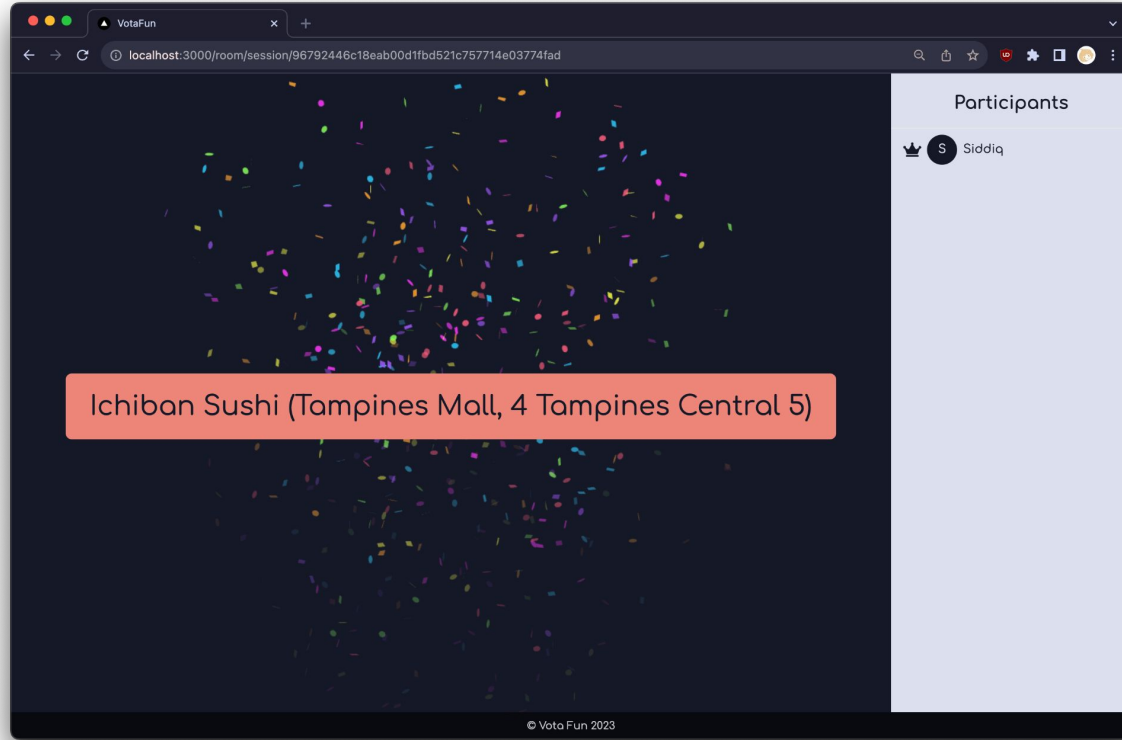




# User Interface



# User Interface

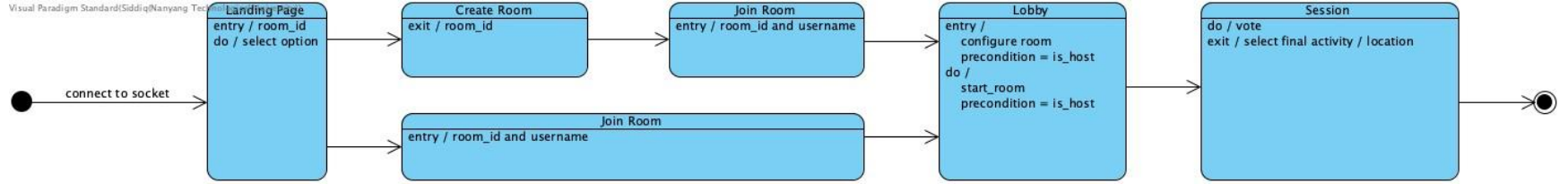




# Dialog Map

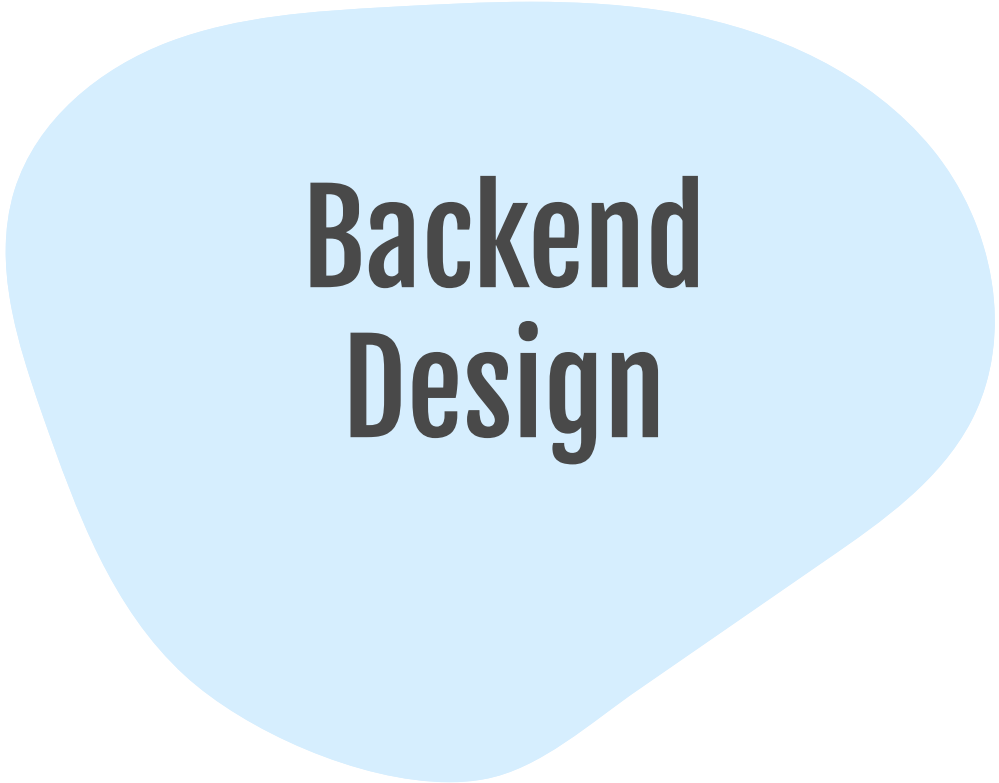



Visual Paradigm Standard (Siddiq Nanyang Technology)

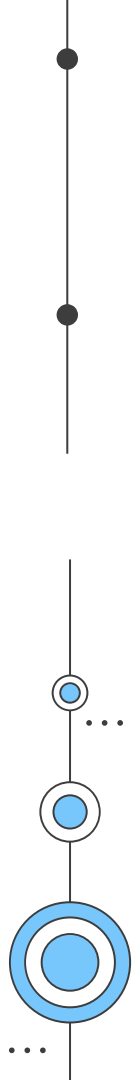


...





# Backend Design



# Backend Tools



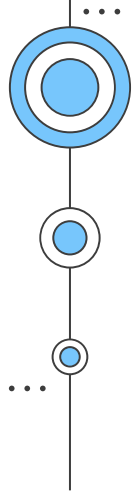
Flask

*flask* - SocketIO

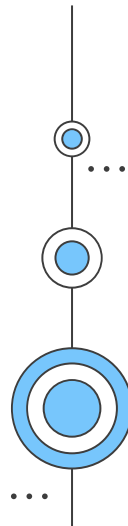


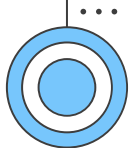
OpenAI

...



# Sockets





# Why do we need sockets?

1. **Real-time communication**
  - Low-latency message transfer
2. **Easy room management**
  - Push notifications
  - Easier to manage endpoints

...



...



...



# Similarities between HTTP and websockets

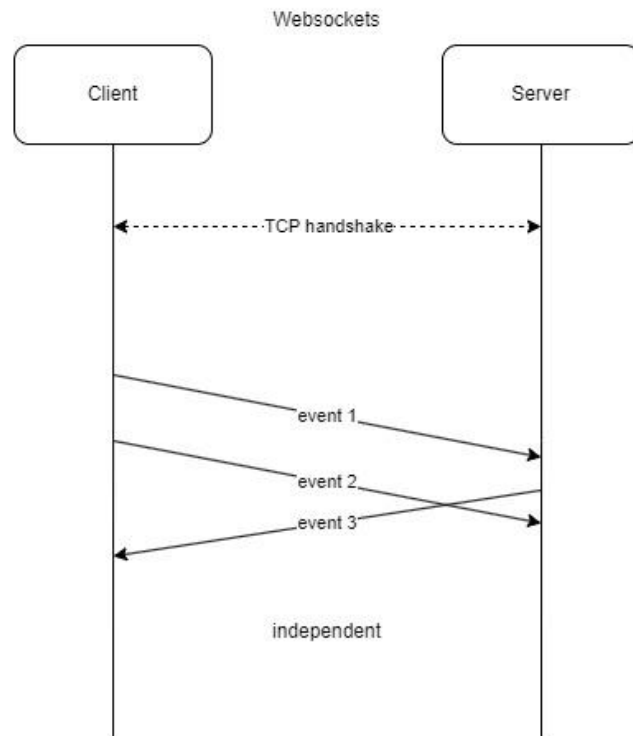
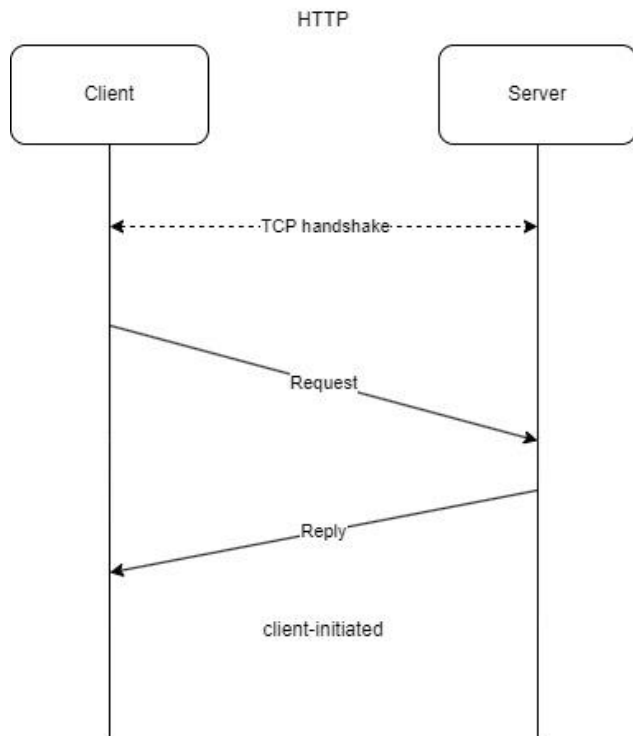
1. HTTP and websockets are both **network protocols**, operate through client-server communication
2. Both use **TCP/IP** as **transport protocol** (lower level)
  - Reliable message transfer
  - Retains message ordering



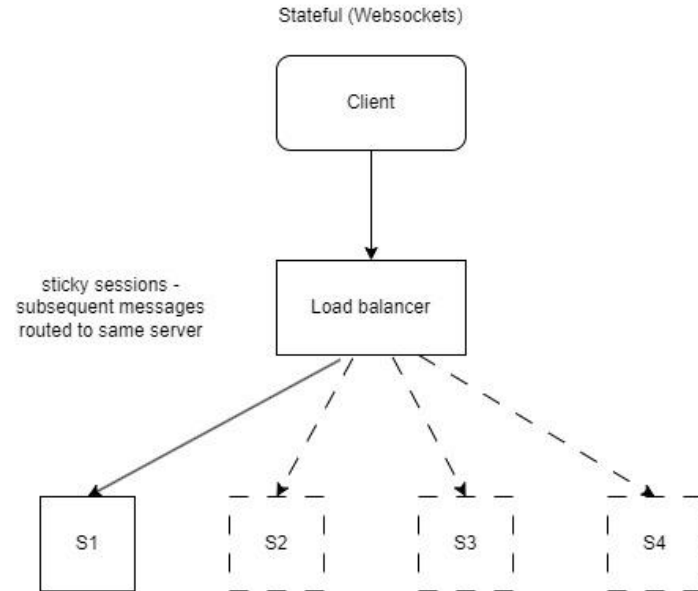
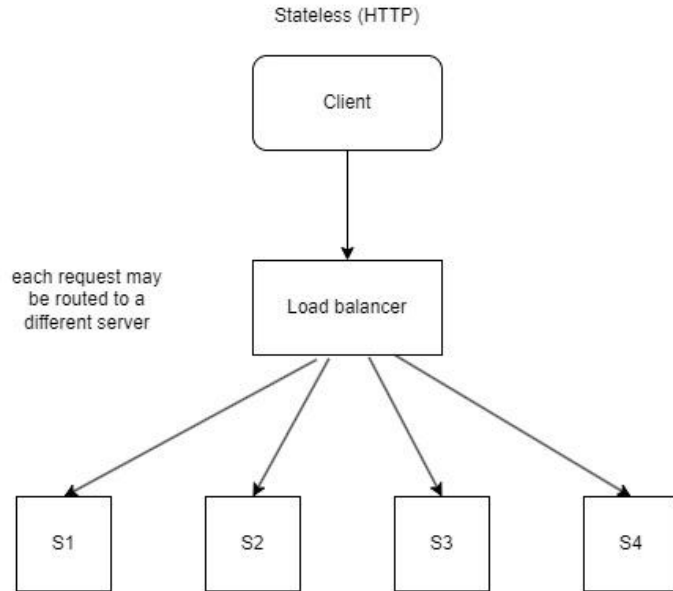
# Differences between HTTP and websockets

<b>HTTP</b>	<b>Websockets</b>
URLs typically start with http/https	URLs typically start with ws/wss
Request-response model	Bidirectional messages
Stateless	Stateful
More overhead	Less overhead
No room management	Supports room management

# Request-reply vs bidirectional



# Stateless vs stateful



# Room management

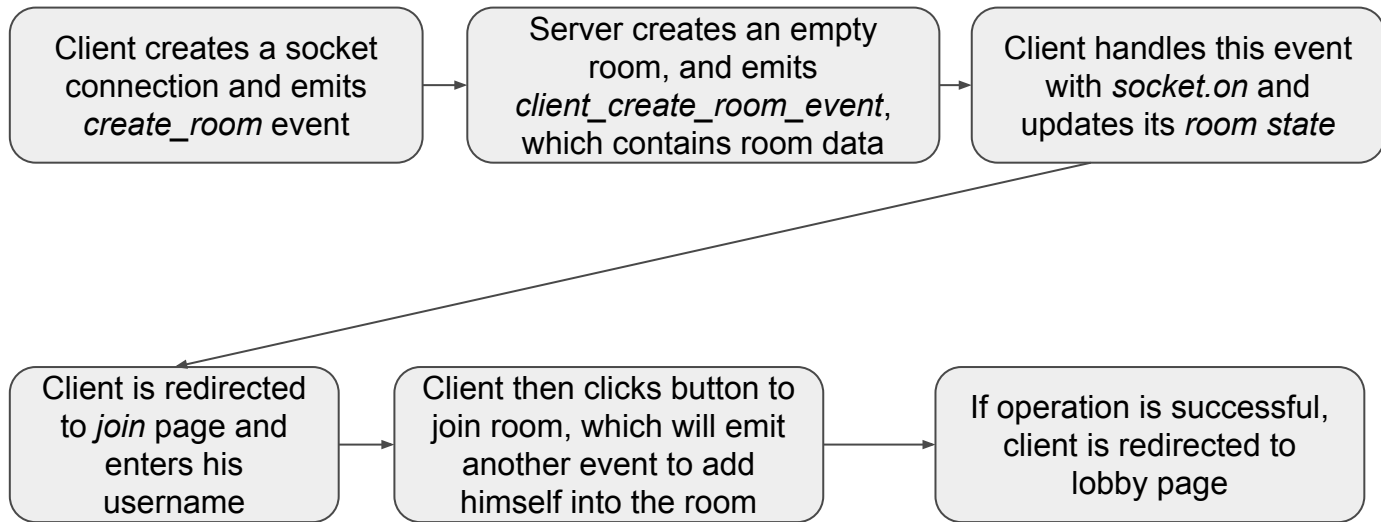
Flask-SocketIO provides easy way to manage rooms using **room id**. It uses an in-memory data structure to store rooms and their connected clients.

It also provides some useful functions.

Eg:

- join\_room
- leave\_room
- close\_room
- disconnect

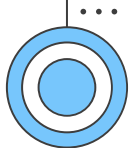
# How did we implement it? (An example)



The server can then send events to a room id to notify everyone in the same room

# Redis database





# Redis Database

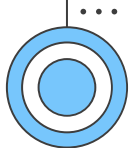
## What is it?

Redis is an open source, in-memory data structure store used as a database.

It is often referred to as a "data structure server." It's known for its speed and efficiency, making it a popular choice for various use cases, such as caching, real-time analytics, session management, and message queuing.



...



# Redis Database (Features)

## Why did we choose it?

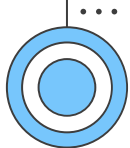
1. In-Memory Data Storage
2. Key-Value Store
3. Lack of data persistence

**It is quick, responsive, and suited for our project's needs.**

...







# Redis Database (Interaction)

1. Upon the creation of a room, the room's attributes are saved and stored in the Redis database, with the 'room\_id' as the primary key.
2. As users enter/leave the room, the information is updated real-time into the database.
3. When the game starts, each question and vote counts are updated in real-time as the game progresses.
4. After the game is over, the room data is cleared, preventing memory accumulation.

...



...



...

# Redis Database (Ensuring ACI)

```
7 def redis_pipeline(func):
8     @wraps(func)
9     def wrapper(self, *args, **kwargs):
10         room_id = kwargs.get("room_id", "")
11         with self.r.pipeline() as pipe:
12             for _ in range(Config.WATCH_ERROR_RETRIES):
13                 try:
14                     pipe.watch(room_id)
15                     pipe.multi()
16                     return func(self, pipeline=pipe, *args, **kwargs)
17                 except redis.exceptions.WatchError:
18                     continue
19             raise Exception(
20                 f"Watch operation failed {Config.WATCH_ERROR_RETRIES} times consecutively."
21             )
22
23     return wrapper
```

# ChatGPT prompt engineering

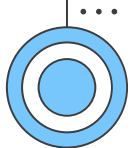


OpenAI

# Prompt engineering introduction

1. What is it? – **Developing and optimizing prompts** to use language models (LMs) for different applications. → To help users plan activities
2. Key principles:
  - a. **Clear and Specific Instructions**
  - b. **Context Setup**
  - c. **Desired Format**
  - d. **Negative Instructions**
  - e. **Iterative Refinement and Experimenta**





# ChatGPT (Initial prompt)

- a. Clear and Specific Instructions
- b. Context Setup
- c. Desired Format
- d. Negative Instructions

We are a group of friends planning for {room\_activity} activity in {room\_location} Singapore and we need your help to figure out the details of our activity. Can you give us a series of 5 questions, namely what, where, when to help us pinpoint the at a specific location? Each question should help narrow down our choices to a single location in {room\_location} Singapore, where we can carry out our {room\_activity} activity. Please avoid asking how far should an activity be, or when it should take place. Each question should provide 4-6 diverse options for us to vote for.

<additional prompt based on room\_activity>

Please avoid providing repetitive or similar questions.

Format the questions in this manner:

Question <x>: <question>

<number y>) <option y>

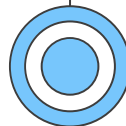
We will tell you the result of our votes in this format:

<option y>) <number of votes for y> ...

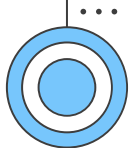
Most importantly, do not repeat any questions and options. Be concise when generating options, preferably within 10 words. Do ask one question at a time.



...



...



# ChatGPT (Re prompts)

- a. Clear and Specific Instructions
- b. Context Setup
- c. Desired Format
- d. Negative Instructions

We are indecisive so give us a properly formatted question with 4 options to vote.  
Remember ask unique questions and options.

Format the questions in this manner:

Question <x>: <question>

1) <option 1>

2) <option 2>

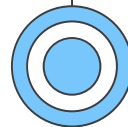
3) <option 3>

4) <option 4>

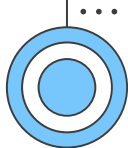
...



...



...



# ChatGPT (Final prompts)

- a. Clear and Specific Instructions
- b. Context Setup
- c. Desired Format
- d. Negative Instructions

We have provided our preferences for the kind of activity we would like to pursue. Based on the voting results, can you recommend us 4 locations in {room\_location} Singapore that we can enjoy our {room\_activity} activities in?

<additional prompt based on room\_activity>

Format the activities in this manner.

Activity 1: Activity name (address)

Activity 2: Activity name (address)

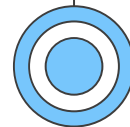
Activity 3: Activity name (address)

Activity 4: Activity name (address)

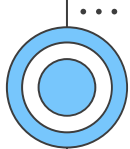
Please provide the location name and address within the same line. Please be specific in the details of the location, namely the name of the location and the address. Remember the location must be in {room\_location} Singapore, and the {room\_activity} activity recommended must be based off our choices. **Do not ask us anymore questions.**



...



...

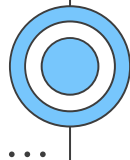


# ChatGPT (Additional prompts)

- a. Clear and Specific Instructions
- b. Context Setup
- c. Desired Format
- d. Negative Instructions

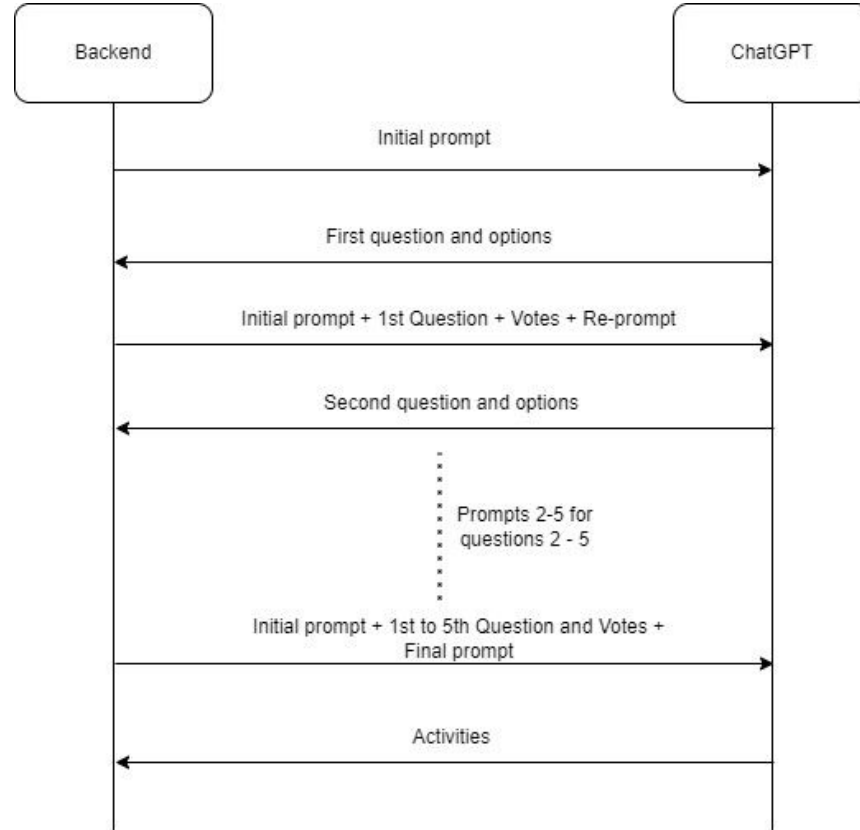
1. Food activity
  - a. Do consider that some of our friends may have dietary restrictions and one of the questions should ask for them (ie. Halal, Vegan).
  - b. Do ask for cuisines from countries located within East Asia and Western Countries or even here in Singapore.
2. Fun activity
  - a. Do ask what type of activity we would like to play.
  - b. Avoid giving an option for virtual activity, the location must be physical.
  - c. Do give a specific address if we are voting for a location.
3. Leisure activity
  - a. Do ask what type of leisure activity we would like to do.
  - b. Do give specific {room\_activity} activity we should do based on the voting results.

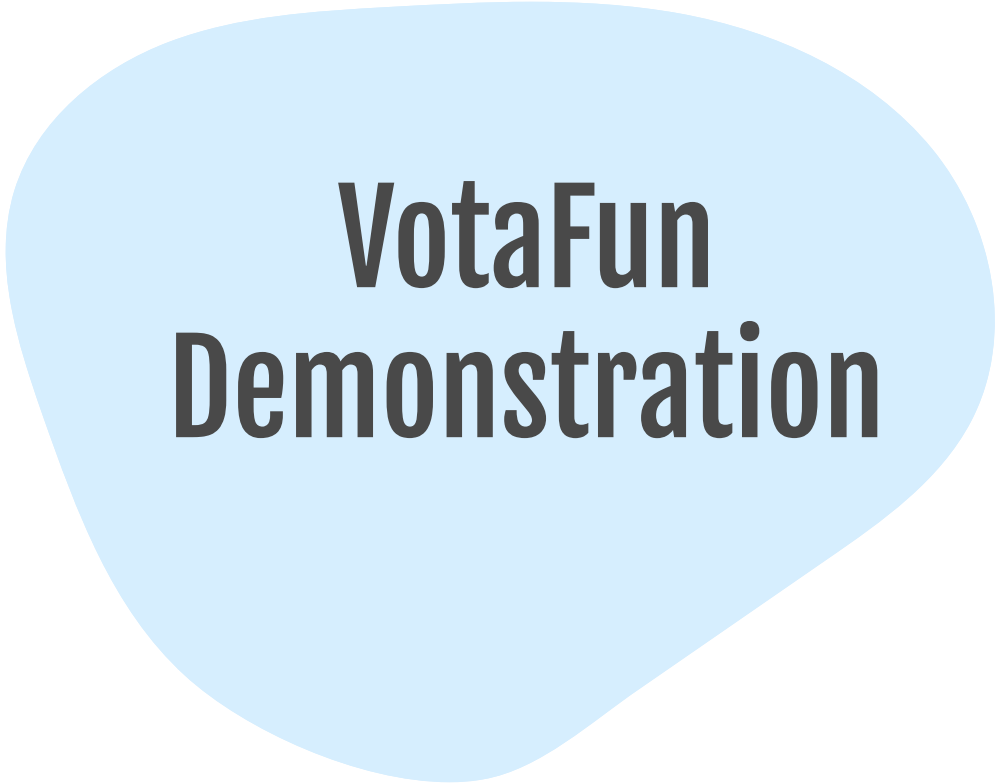

...



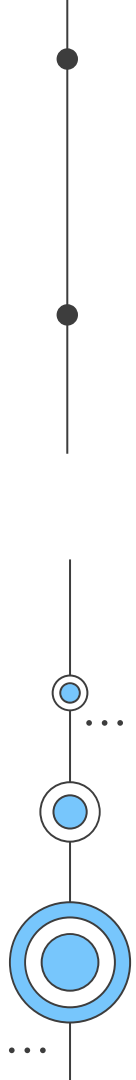


# ChatGPT process





# VotaFun Demonstration





# VotaFun Releases



# Initial Release (Version 1.0.0)

Release Version 1.0.0	
Functionality	Description
Front-end Interactive Webpage: <ul style="list-style-type: none"><li>- Create/ join room</li><li>- Lobby page</li><li>- Session page</li></ul>	Main functions implemented as defined in initial SRS
Socket integration	Fully implemented as defined in initial SRS
Redis database	Fully implemented as defined in initial SRS
ChatGPT prompt engineering	Fully implemented as defined in initial SRS

...

# Minor Release (Version 1.1.0)

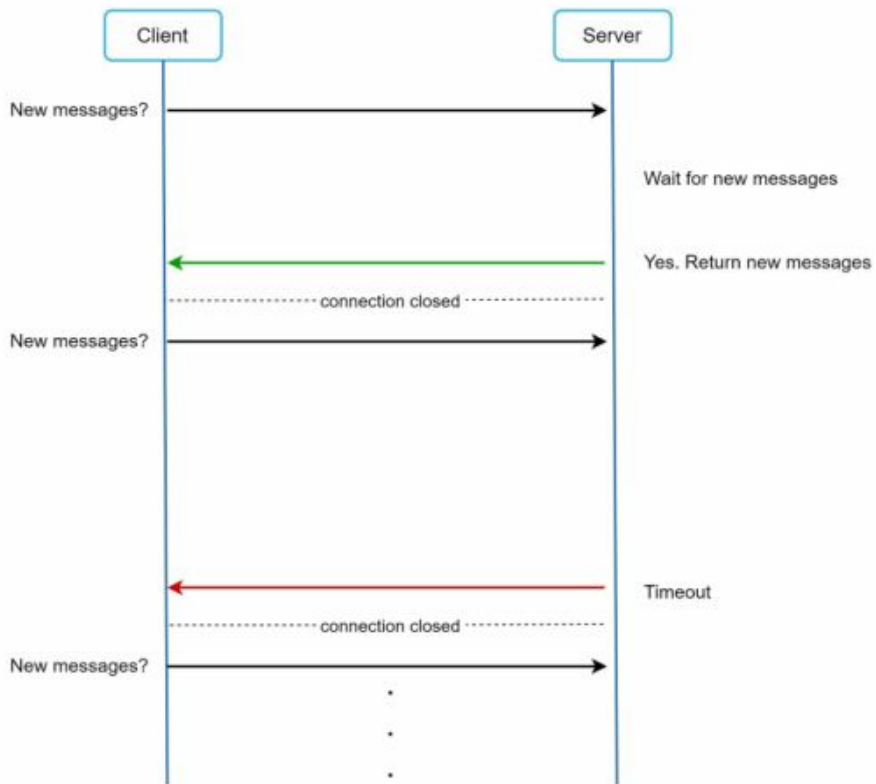
Release Version 1.1.0	
Functionality	Description
Front-end Interactive Webpage: <ul style="list-style-type: none"><li>- Lobby page</li><li>- Session page</li></ul>	Bug fixes and minor improvements. <ul style="list-style-type: none"><li>• Leave and close room buttons.</li><li>• Users can set the max capacity of rooms.</li></ul>
ChatGPT prompt engineering	Minor improvements <ul style="list-style-type: none"><li>• Improvements of prompt engineering</li></ul>

# Major Release (Version 2.0.0)

Release Version 2.0.0	
Functionality	Description
Front-end Interactive Webpage: <ul style="list-style-type: none"><li>- Summary Page</li><li>- Regenerate question and options</li></ul>	Introduction of new major features or system overhaul <ul style="list-style-type: none"><li>• Summary page which summarises the votes for all questions.</li><li>• Users can regenerate questions and options if they are not satisfied.</li></ul>
Socket integration	Introduction of new major features or system overhaul <ul style="list-style-type: none"><li>• Attempt to reconnect to the same session if user disconnect due to unreliable internet connection</li></ul>
Database	Introduction of new major features or system overhaul <ul style="list-style-type: none"><li>• Improve redundancy and multiple backups</li><li>• Addition of durable storage (ie traditional DB)</li><li>...</li></ul>
New large language models (LLMs)	Introduction of new major features or system overhaul <ul style="list-style-type: none"><li>• New LLMs for users to use instead of only ChatGPT.</li></ul>

A decorative graphic consisting of blue circles of varying sizes connected by thin black lines, forming a network-like structure around the central text. Some circles have concentric rings. Ellipses (...) are placed at several points where the lines end or branch.

**Thank  
You**



Keep-alive +  
watch/ poll??





# Alternatives to websockets



1. HTTP long polling
  - Client holds connection open until new messages are available/ timeout is reached
  - Once client receives message, it immediately sends another request, restarting the long polling
  - Issues:
    - Inefficient, long polling still makes periodic connections after timeout even when user is not chatting
    - Repeated opening and closing of connections is not ideal

