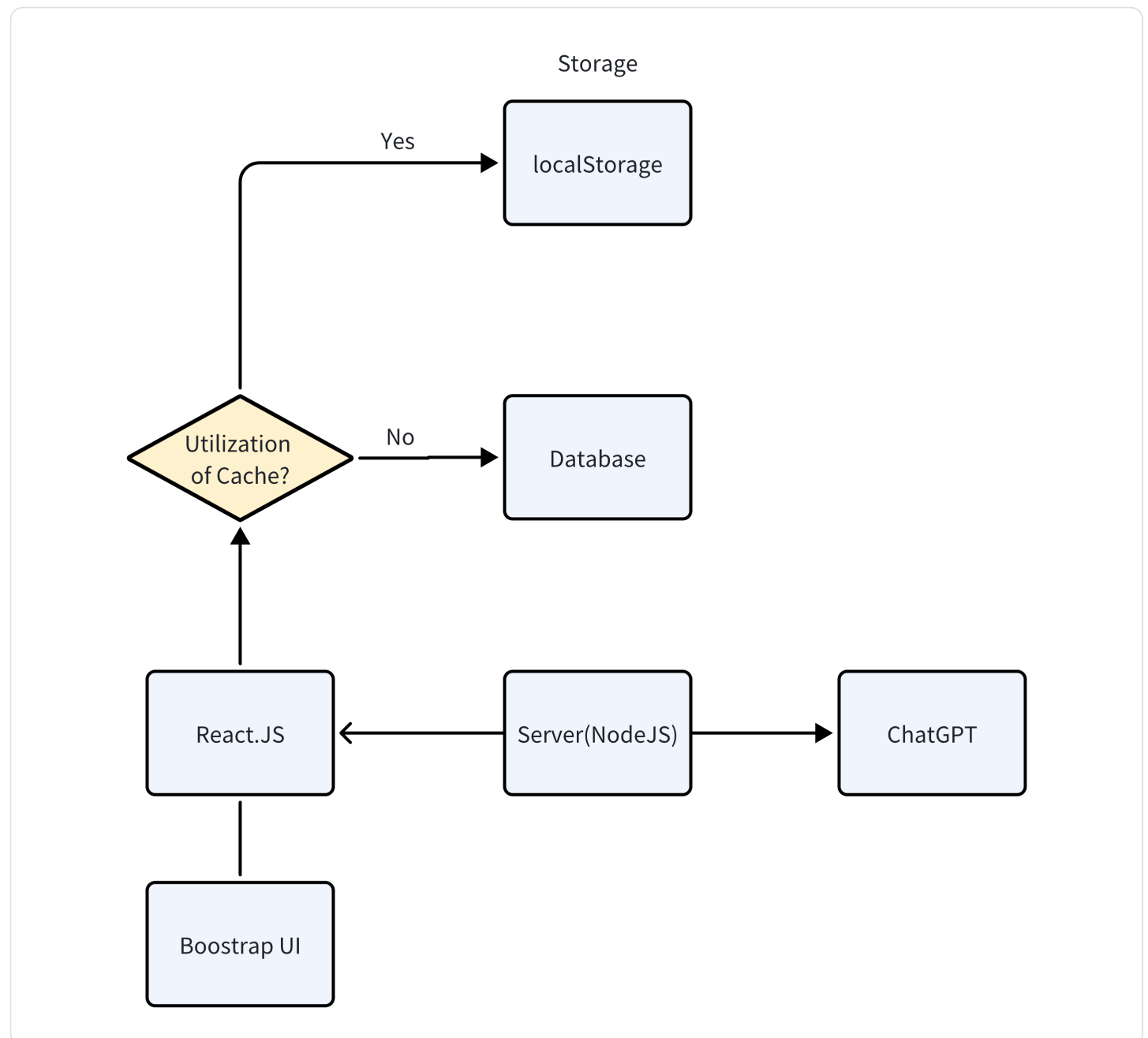


AnimalChat

Architecture Design

Design v1.0



Frontend

React

React components for different parts of the frontend

- User Input Component: For account validation and User Text Input Message

- b. Animal Selection component: To allow user to choose animals
- c. Chat Display Component: For showing the conversation between the user and the selected animal.
- D. Navigation Components: If you have multiple pages or sections, define navigation components.

Bootstrap

Although Materialize, Foundation and other frameworks are applicable choices for our ChatApp, Bootstrap is chosen because of its rich documentation, adoption and community resources. This could reduce complexity in our development easier when there is a need for seeking online resources.

Backend

Storage: localStorage, Cache, Cookie

LocalStorage

LocalStorage is chosen for temporarily storing user-related profile information in the interim stage. Due to its limited storage capacity, cache or cookie are in our choices for future storage options.

Besides, tokens with timestamp could be used for setting the lifecycle of localStorage data, for good practice of user data security.

NodeJS

Node.js is designed to be non-blocking and asynchronous, making it well-suited for handling real-time applications like chatbots. It can handle multiple incoming requests concurrently, allowing our chatbot to respond quickly to users.

Given limited project duration, deploying Node.js applications is relatively straightforward. We can host our chatbot server on our own server, depending on our specific needs.

Database

1. User:

- Fields:
 - `id` (Generated unique identifier for each user)
 - `email` (Unique identifier for users)
 - `password` (Encrypted password)
 - Other user profile information (e.g., name, profile picture, etc.)

2. Chat:

- Fields:

- `id` (Generated unique identifier for each chat)
- `userId` (Foreign key referencing the User entity)
- `animalId` (Foreign key referencing the Animal entity)
- `messages` (Array or text field to store the chat messages)
- `created_at` (Timestamp for when the chat was created)

3. Animal:

- Fields:

- `id` (Generated unique identifier for each animal)
- `name` (Name of the animal)
- `soundUrl` (URL for the animal's sound)
- `imageUrl` (URL for an image of the animal)

Users	
id	int
Email	vachar(255)
Password	varchar(255)
Other profile	TBC

ChatId
ChatId
UserId
AnimalId
Messages
created_at

Animals
AnimalId
AnimalName
AnimalImageURL
AnimalSoundURL