

Technical Report

Table of Contents

1	Summary	1
2	Technologies	1
2.1	Frontend.....	1
2.2	Backend.....	1
2.3	Deployment.....	1
3	Flow chart.....	2
4	Source code and demo	2

1 Summary

This application utilizes Google Vertex AI with Gemini models to generate content and messages from media files like images, pdf, etc. It is built using Typescript, Golang and their related technologies.

2 Technologies

2.1 Frontend

- React (18) function components
- Next.js 14 with App router.
- Authentication: NextAuth.js for quick development.
- Server-side rendering: React Server Components.
- Form submissions: use useFormState hook (react-dom) for calling server actions.
- React hooks: useState, useEffect, Context API, useReducer, etc.
- Communication: Axios, Websocket API

2.2 Backend

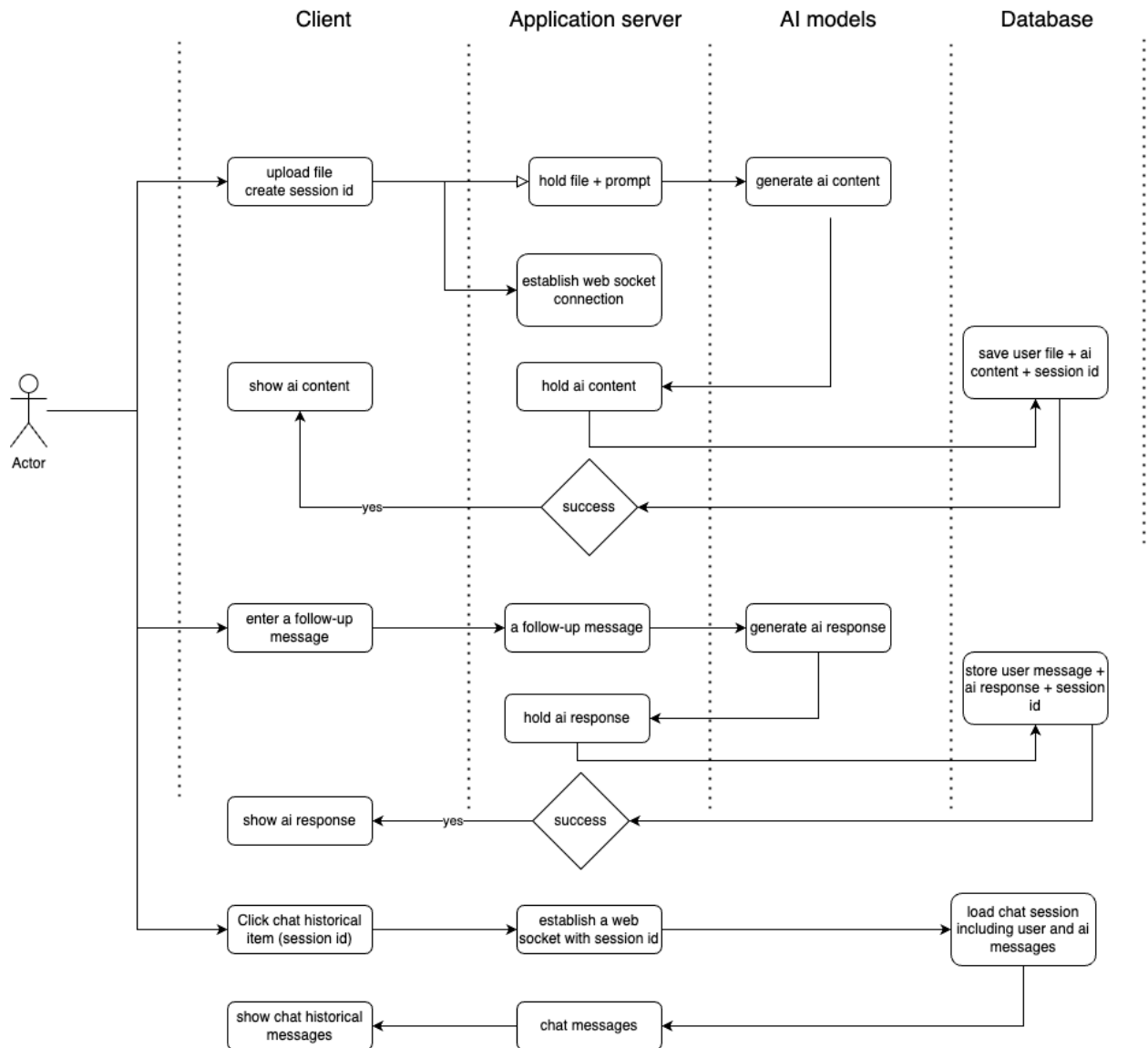
- Go modules:
 - o user service: manage user accounts, authentication.
 - o story service: chat messages, sessions, media files and chat history.
- Golang 1.21.5, Gin Framework.
- Vertex AI Go SDK for GenAI multimodel with Gemini-1.5-Flash.
- Currently application only support file formats: image(png, jpg, and jpeg), and pdf.
- PostgreSQL database for store user accounts, chat sessions, and media files.

2.3 Deployment

- Front-end: Vercel

- Backend: GCP VM with Docker containers, and Nginx.

3 Flow chart



4 Source code and demo

- Frontend: <https://github.com/nhat8002nguyen/Story-of-media-AI>
- Backend: <https://github.com/nhat8002nguyen/story-of-media-be>
- Live app: <https://story-of-media-ai.vercel.app/>
- Demo video: <https://youtu.be/YMQ7PlpP5ic>