**CS673 Software Engineering** 

**Team5 - NoteAnt**

**Software Design Document**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Frankie | Requirement leader | *Guancheng Huang* | Oct 19 |
| Chris | Design and Implementation leader | Wenhao Tian (Chris) | Oct 19 |
| Nicholas | Team Leader | *Nicholas Narmada* | Oct 19 |
| Wayne | Configuration leader | *Yichen Li* | Oct 19 |
| Siyuan Wan | Security Leader | *Siyuan Wan* | Oct 19 |
| Yibo(Wilbur) | QA Leader | *Yibo Wang* | Oct 19 |
|  |  |  |  |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| **iter1** | **Siyuan Wan，**  **Chris，**  **Wayne** | **19 Oct, 2023** | **First** |
|  |  |  |  |

[Introduction](#_87t9hln2vjz0)

[Software Architecture](#_3ipvmjgn6clp)

[Class Diagram](#_ky60nv8suxxm)

[UI Design (if applicable)](#_7ucksmkf6rzx)

[Database Design (if applicable)](#_tcmuor4nl1kz)

[Security Design](#_x18fj36s1121)

[Business Logic and/or Key Algorithms](#_mtfbusfb0eq3)

[Design Patterns](#_9zvwkmc4luo5)

[Any Additional Topics you would like to include.](#_15tmymhipvdv)

[References](#_50ojo9i46ytq)

[Glossary](#_8n34lvocupub)

# Introduction

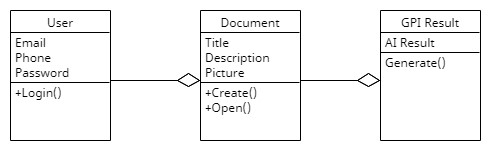
We all take notes at some point, either in class or in a meeting. One of the popular methods of note taking is typing down your thoughts while listening. A lot of times the notes we took can be hard to read or confusing.

To solve this problem, our project aims to develop a ChatGPT-like application that combines text editing, to provide users with a real-time, interactive content generation and editing experience. The application is architected to facilitate user interaction, data processing and integration with the OpenAI API.

# Software Architecture

We first divide tasks into 3 pages: Login, user documents page, and document page. The login page handles user authentication and assigning tokens. User documents page displays all the documents belonging to this user. It also handles creating documents and delete documents requests. Then, when the user clicks onto one of the documents, the document id will be sent to the third page, document page. The document page will grab the document from MongoDB to present to the user. After the user finishes editing and clicks save, a post request will be sent to the document page for update. Another function on this page is generate. After the user clicks generate, document content will be collected and sent to OpenAI. OpenAI will return the summarized info back to the server.

# Class Diagram



# UI Design (if applicable)

Our UI design part is divided into 3 front end pages.

The first page is login page. It use GSI(Google Sign In), allow users login their account by their google account.

The second page is user document page. User can manage their personal info, their documents and create new note (document) in this page. Our document card shows document pictures, document title and description. Then user double click one of their documents, they can be navigated to the third page. (GPI page)

The third GPI page like Word. You can change your document style, like font size, etc. And you can choose your note, click generate button. The GPT will return the summarized content or something you want about your document.

# Database Design (if applicable)

For this project, we will use 2 data models: User and Documents.

The User model only contains : user\_id and username

The Document model contains : title, document.\_id, body, summary, last\_modified, and author, which reference to user.id.

# Security Design

Our software system prioritizes security by leveraging Google’s OAuth 2.0 protocol for authentication and authorization. This ensures that user’s passwords are never stored or seen by our application. Instead, Google authenticates the user and provides our application with a secure token. All communications are encrypted using HTTPS, and we only request necessary permissions from users, minimizing data access. Regular monitoring, logging and updates further bolster our commitment to safeguarding user data.

# 

# Business Logic and/or Key Algorithms

Json Web Token(JWT): JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way to securely transmit information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed with a secret (using the HMAC algorithm) or with a public/private key pair using RSA or ECDSA. It used for our Authentication in our project.

# Design Patterns

# Any Additional Topics you would like to include.

# References

# Glossary