

Low Level Design (LLD)

Notes Keeping App

Ngamlenmang Touthang

Document Version Control

Date Issued	Version	Description	Author
11/01/23	1.0.0	Server initial setup	Ngamlenmang Touhang
12/01/23	1.0.1	User authentication added	Ngamlenmang Touhang
13/01/23	1.0.2	Notes controller and routes added	Ngamlenmang Touhang
14/01/23	1.0.3	Swagger docs added	Ngamlenmang Touhang
15/01/23	1.0.4	Client template with responsive added	Ngamlenmang Touhang
16/01/23	1.0.5	Client side authentication UI added	Ngamlenmang Touhang
18/01/23	1.0.6	Search feature added	Ngamlenmang Touhang
19/01/23	1.0.7	Cookie update	Ngamlenmang Touhang
02/02/23	1.0.8	Documents updated	Ngamlenmang Touhang

Contents

Document Version Control.....	2
Abstract	4
1. Introduction.....	5
1.1. Why this Low-Level Design Document?.....	5
1.2. Scope of the Project.....	5
1.3. Constraints.....	5
2. Technical Specifications.....	6
2.1.Platform.....	6
2.2.Data storage.....	6
2.3.Security.....	6
2.4.User Interface.....	6
2.5.Network.....	6
2.6.Performance.....	6
2.7.Technical Dependencies.....	6
2.8.Deployment.....	7
3. Technology stack.....	8
4. Proposed solution.....	8
5. Design Work flow.....	9
6. User I/O workflow.....	9
7. Key performance indicators (KPI).....	10

Abstract

A web-based application designed to simplify note taking, the app offers an easy-to-use platform for organizing and saving information. Users can create, edit, customize note background and font colour and also search through their notes quickly and efficiently with its title. With the ability to access their notes from any device with the internet connectivity, this application is perfect for both personal and professional use.

1. Introduction

1.1. Why this Low-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model of coding. The document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

1.2. Scope of the project

1. Everyone overlooks crucial items in this hectic world, but adopting a note-taking software can help. Many users will benefit from this app by having a complete list of their chores and stuff in one location
2. The user will feel at ease interacting and using it because to the straightforward but interesting user interface.

1.3. Constraints

The users will only able to use the application with internet connectivity.

2. Technical Specifications

2.1. Platform:

- The keeper notes application web based application and different responsive screen size.
- It will be cross platform.

2.2. Data Storage

- MongoDB Atlas will be used as database management system to store notes and user information.
- The data store in database will be store as Document (NoSql) and mongoose for making the design schema

2.3. Security

- Bcryptjs and Jsonwebtoken is used for encryption and secure the storage of sensitive information (e.g. user passwords)
- User authentication and authorization mechanisms

2.4. User interface

- User interface design and layout with Reactjs library
- User interaction design

2.5. Network

- Data transfer mechanism is mainly done via REST API

2.6. Performance

- Scalability and performance optimization techniques
- Load testing and stress testing procedures

2.7. Technical dependencies

- Some of the third party libraries, frameworks and tools to be used in development are – bcryptjs, cookie-parser, cors, dotenv, express, jsonwebtoken, mongoose, morgan, nodemon, swagger-ui-express, validator, yamlls, reactjs, react-toastify, react-color and axios

2.8. Logging

- The react-toastify log ever success and failure events that occurs in the application

2.9. Deployment

The Notes keeper app is deployed using Railway (backend), Netlify (frontend) and database (MongoDB Atlas) via github which is software version control system



netlify



mongoDB® Atlas

3. Technology stack

Front End	HTML, CSS, React, React-toastify
Backend	Node, mongoDB, Express, mongoose, bcrypts, cookie-parser, cors, dotenv, jsonwebtoken, mongoose, morgan, nodemon, swagger-ui-express, validator, yamls
Database	MongoDB
Data transfer	REST API
Deployment	MongoDB Atlas, Railway, Git and Netlify
Version control	Git

4. Proposed Solution

The proposed solution for a notes-keeping web application include the following features

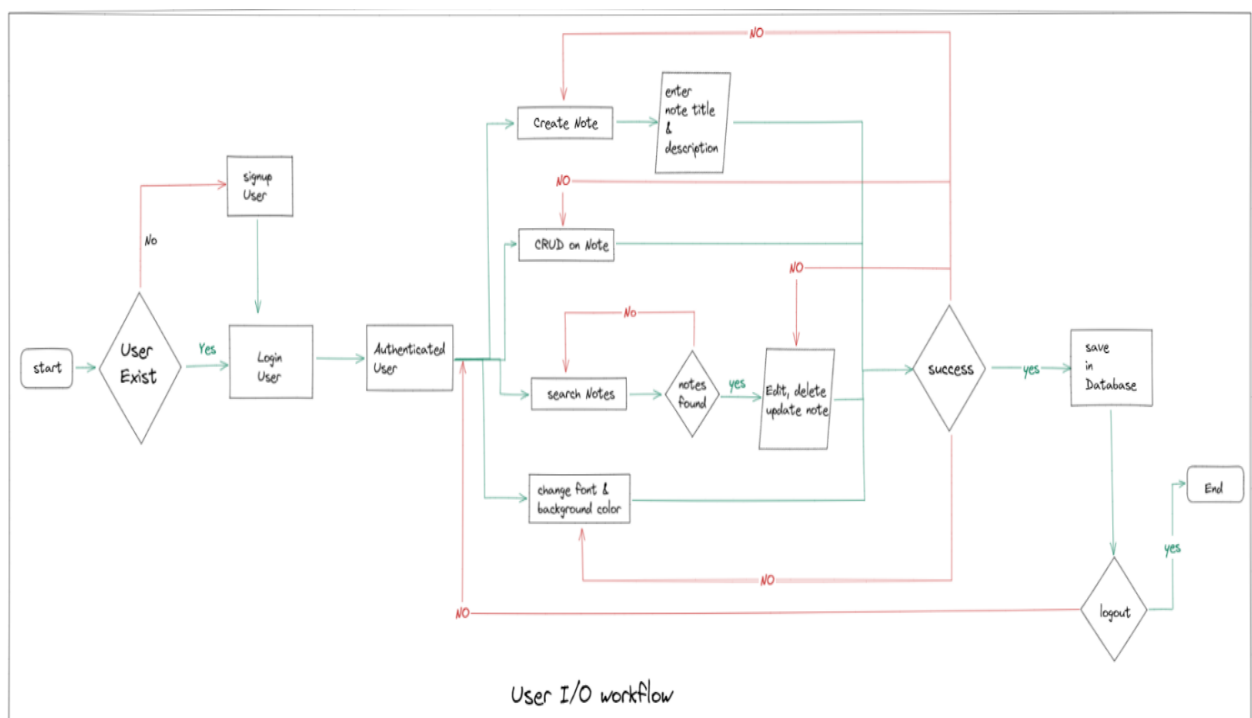
1. User Accounts: Allow users to create an account and sign in to access their notes from any device.
2. Note creation and editing: Provide an intuitive and user-friendly interface for users to create, edit and delete notes.
3. Customizable background and font colour: Provide modifiable note card background and font colour.
4. Search functionality: implement a search bar to help users quickly find specific notes base on the notes title name.
5. Mobile Compatibility: Ensure that the application is accessible and functional on both desktop and mobile devices.
6. Security and privacy: Implement robust security measures to protect user's data and ensure the privacy of their notes.
7. Timestamp: to help the user better understand when the note was written, the creation and modification dates is provided to every note.

5. Design Workflow

Low Level user flow



6. User I/O workflow



7. Key performance indicators (KPIs)

Key Performance Indicators (KPIs) are metrics used to measure the success and effectiveness of a product or system. For The notes keeping app,

1. User Retention: The percentage of users who continue to use the app over time, indicating satisfaction with its performance and features
2. Active User Count: The number of unique users who access the app regularly, indicating its popularity and usefulness
3. Note Creation and Editing Activity: The number of notes created and edited by users, indicating their level of engagement and use of the app
4. Search Usage: The frequency and effectiveness of the search functionality, indicating the usefulness of these features for users
5. Feedback and Rating: user feedback, ratings and reviews can provide valuable insight into the app's performance and areas of improvement.
6. Data Privacy and security: Measures to ensure the safety and security of user data, such as encryption and secure storage are critical to the success of the app.