



The Development of Mindiqo, a Note-Taking Application

Jiakun Ge

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements
of the Degree of Master of Science at the University of Glasgow

30th August 2023

Abstract

The purpose of this project is to develop a comprehensive note-taking application-“Mindiqo” designed to alleviate the issue of information overload. The application aims to serve as both a personal knowledge management tool and a community-sharing platform. Its dual nature enables users not only to create, organize, and store their notes but also to share their knowledge in a community.

After a thorough evaluation of existing solutions in the market, Mindiqo was architected to meet stringent functional requirements. To ensure superior performance and reliability, the application is built on a contemporary technology stack consisting of Spring Boot, Vue, MySQL, and Redis. Comprehensive tests, encompassing backend unit tests, API endpoint tests, and user evaluations, were rigorously conducted to assess the application's robustness.

Mindiqo has undergone stringent verification processes to affirm its stability and reliability. Although user experience and interface design have been identified as areas for enhancement, the core functionalities have met with high approval ratings among users, substantiating the application's efficacy.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

Name: Jiakun Ge

Signature: 

Acknowledgements

I would like to express my deepest gratitude to my advisor, Doctor Chigozie Onyekaba, for his continuous support, mentorship, and guidance throughout the course of this research. Your insights and wisdom have been invaluable.

I would also like to extend my heartfelt gratitude to the University of Glasgow for all the support and resources provided, which have been instrumental in the completion of this research.

Lastly, I am grateful to my family and friends for their unending love and encouragement, without which this work would not have been possible.

Contents

Chapter 1	Introduction	1
1.1	Aims and Objectives	1
Chapter 2	Background & Requirements Analysis.....	2
2.1	Background	2
2.2	Requirements Analysis.....	3
2.2.1	Existing Note-Taking App Analysis.....	3
2.2.2	Active Online Communities Analysis	4
2.3	User Stories and MoSCoW Prioritization.....	5
Chapter 3	Design and Implementation	6
3.1	Design.....	6
3.1.1	Technology Stack	6
3.1.2	System Architecture	7
3.1.3	Database Design	8
3.1.4	Application User Interface Design.....	10
3.2	Implementation	10
3.2.1	User Authentication.....	11
3.2.2	Note-taking feature.....	12
3.2.3	Knowledge Community Feature.....	13
Chapter 4	Testing & Evaluation.....	15
4.1	Backend Unit Test	15
4.2	API Endpoint Testing.....	16
4.3	User Evaluation Testing	17
Chapter 5	Conclusion	19
5.1	Summary	19
5.2	Limitations and Challenges.....	20
5.3	Future Work.....	20
Chapter 6	References	22
Appendix A	User Stories	1
Appendix B	Moscow Prioritization.....	17
Appendix C	Wireframe.....	18
Appendix D	User Evaluation Questionnaire	21
Appendix E	User Evaluation Questionnaire Responses.....	24
Appendix F	Ethics Checklist.....	27

Chapter 1 Introduction

The proliferation of information technology has exacerbated the challenge of information overload, a phenomenon wherein individuals are overwhelmed by an excessive amount of information that exceeds their cognitive capacity to process (Roetzel, 2019; Filickova, 2016). This situation not only hampers productivity but also stifles innovation (Dean & Webb, 2011). Given the urgency of the problem, there is a compelling need to find effective strategies to manage, recall, and share the deluge of information available to us.

Personal Knowledge Management (PKM) has emerged as an effective solution to this challenge (Jefferson, 2006). PKM involves the systematic collection, classification, and storage of information, enabling individuals to create their repositories of knowledge. This enhances personal efficiency and sparks creativity. However, successful PKM necessitates an organizational structure and tools tailored for individual growth (Jefferson, 2006).

In the wake of advancements in information technology, online note-taking software has become a cornerstone in PKM (Razmerita et al., 2009). These applications equip knowledge workers with powerful tools to collect, organize, and share information, providing a more flexible and efficient approach to knowledge management for both individuals and teams.

1.1 Aims and Objectives

The primary objective of this project is to develop a note-taking application. The application aims to offer a holistic platform by integrating functionalities for online note-taking and community sharing.

This application has the following key features:

Online Note Editing and Storage: The application allows users to create and store notes in a variety of multimedia formats, including text, images, and audio. It also offers intuitive categorization and a robust full-text search feature, making it easy for users to retrieve information.

Knowledge-Sharing Community: In addition to personal note-taking, the application features a community-sharing module where users can publish their notes or articles. This open forum allows for commenting and appreciation, fostering a culture of knowledge exchange and social interaction.

In summary, this application provides a comprehensive, one-stop solution for tackling the issue of information overload through effective knowledge management and community sharing.

Chapter 2 Background & Requirements Analysis

2.1 Background

In the current era of information explosion, we are continuously surrounded by an ever-growing and constantly updating stream of information. This vast influx of data often results in many experiencing information overload. In such an environment, the precise selection, compilation, management, and apt utilization of this information becomes paramount. This underscores the true value of knowledge management: extracting meaningful insights from the overwhelming flow of information to make more informed decisions(Tiwari, 2022). Faced with this challenge, many individuals turn to note-taking as a crucial method for their personal knowledge management.

With the advancement of the information age, the modes and methods of note-taking have undergone profound transformations. From initial simple records saved solely in plain text (txt) format on personal computers, to today's sophisticated applications stored in cloud-based personal knowledge bases, such as Evernote, OneNote, and Notion, we have witnessed the evolution of technology intertwined with knowledge management. While these applications have been widely embraced and loved by users, their primary design intent was to offer a foundational environment for users to capture and store information. Such software is largely positioned as personal tools, primarily focusing on individual knowledge management, but often overlooking the knowledge-sharing functionalities for individuals or teams.

Nowadays, society is placing an increasing emphasis on openness and collaboration, with knowledge sharing being the key driving force. Both individuals and organizations are increasingly recognizing the importance of this knowledge exchange. In the modern era of information technology, the flow and exchange of knowledge are essential for advancement. GitHub embodies the ethos of knowledge sharing. It's a global hub where developers collaborate, co-create, and exchange insights. Open-source projects on GitHub unite developers worldwide, fostering collective progress. This collaborative spirit accelerates software development. Here, individuals are viewed as nodes in a vast network, not isolated entities. Sharing expertise amplifies its reach and impact, fueling innovation and growth.

In conventional note-taking applications, sharing one's amassed knowledge often necessitates recourse to electronic mail or messaging applications, accompanied by limitations on the number of recipients and efficiency issues. Against this backdrop, the value of a Note-taking application centred on knowledge-sharing community sharing becomes increasingly apparent, which transcends being a mere knowledge storage tool and evolves into a hub for knowledge-sharing and collaboration. On such platforms, users can not only manage and save their notes but also freely publish their notes, inviting others to provide comments and additions, gradually forming a vibrant knowledge community. This approach not only significantly enhances the efficiency of knowledge dissemination but, more importantly, fosters the spread and innovation of knowledge.

2.2 Requirements Analysis

For a note-taking App centred on a knowledge-sharing community, its core features can be primarily divided into two categories:

1. Note-taking functionality: This includes content editing and saving.
2. Community interaction features: This includes content sharing and interaction.

To gain a deeper understanding of these two core features, we plan to adopt the following strategies: Firstly, regarding the note-taking functionality, we will approach from the user's perspective, conducting in-depth testing and analysis of the mainstream note-taking software currently on the market. This will allow us to identify their strengths and weaknesses, which will in turn inform the optimization of our product design. Secondly, for the knowledge community features, we will analyze currently active online communities as references. This will help us understand their operational mechanisms and user behaviour patterns, ensuring that our software can meet users' needs in terms of knowledge sharing and communication.

2.2.1 Existing Note-Taking App Analysis

To assess the features that contribute to an excellent note-taking application, a comparative analysis was conducted on four popular Apps available in application marketplaces. These Apps include Evernote, OneNote, Notion, and Google Keep. While each of these applications presents unique features and advantages, certain commonalities exist among them.

The following **Table 1** delineates the primary functionalities offered by each of these four note-taking applications:

Features/Apps	Evernote	OneNote	Notion	Google Keep
Text Formatting	Yes	Yes	Yes	No
Multi-Platform Support	Yes	Yes	Yes	Yes
Syncing	Yes	Yes	Yes	Yes
Rich Text Editing	Yes	Yes	Yes	Limited
Markdown Support	No	No	Yes	No
Media Insertion	Images, Audio	Images, Audio, Video	Images, Audio, Video	Images
Search Functionality	Yes	Yes	Yes	Yes
Organizational Structure	Tags, Folders	Tabs, Pages	Databases, Pages	Labels, Colors
Sharing & Collaboration	Yes	Yes	Yes	Yes
Backup & Export	Yes	Yes	Yes	Yes

Table 1: Overview of Note-taking App Features

According to **Table 1**, the key factors and functional features that an excellent note-taking app needs to possess can be summarized as follows:

1. Text Formatting: Allows for flexible information management and presentation.
2. Multi-Platform Support and Sync: Ensures that notes can be accessed and edited across various devices and operating systems.
3. Rich Text Editing: Enables the insertion of multimedia elements such as images, audio, and video.
4. Search Functionality: Provides a convenient way to quickly find information within a large collection of notes.
5. Organizational Structure: Offers tools like tags, folders, or databases for categorizing and managing notes.
6. Sharing and Collaboration: Simplifies and enhances teamwork.
7. Backup and Export: Facilitates data migration and secure storage.

2.2.2 Active Online Communities Analysis

To understand the key features that contribute to a successful knowledge-sharing community within a note-taking app, we conducted an in-depth analysis of several well-established online communities, such as Reddit, Stack Overflow, and Quora. Although these platforms are not specifically designed for note-taking, they serve as excellent case studies for understanding the dynamics of effective knowledge-sharing and community engagement. The following **Table 2** shows some common features that these platforms share.

Features/Communities	Reddit	Stack Overflow	Quora
Upvoting/Downvoting	Yes	Yes	Yes
Tags/Categories	Yes	Yes	Yes
Search Functionality	Yes	Yes	Yes
User Profiles	Yes	Yes	Yes
Following Feature	Yes	Yes	Yes
Discussion Threads	Yes	Limited	Limited

Table 2: Features of Active Online Communities

According to **Table 2**, the essential features that contribute to a successful online community within a note-taking app can be summarized as follows:

1. Upvoting/Downvoting: Allows for community-driven quality control of content.
2. Tags/Categories: Facilitates the organization and discovery of relevant topics or questions.
3. Search Functionality: Enables users to quickly find information or discussions that are relevant to their needs.
4. User Profiles: Allows for a personalized experience and enables users to establish credibility within the community.
5. Following Feature: Enables users to keep track of topics, questions, or people that interest them, thereby encouraging ongoing engagement.

6. Discussion Threads: Provides a structured format for dialogue, although this feature is limited in platforms like Stack Overflow and Quora compared to Reddit.

2.3 User Stories and MoSCoW Prioritization

Based on the analysis of existing note-taking applications and vibrant online communities, the proposed note-taking app centred on a knowledge-sharing community can be designed to cater to six distinct user roles.

- Note Creator: They are primarily responsible for creating, managing, and editing notes.
- Poster: This group is tasked with sharing the notes within the community and responding to comments and questions.
- Community Member: They mainly engage in discussions within the community and participate in quality control through upvoting.
- Collaborators: They collaborate with other users to jointly create and edit notes.
- Admin: They are chiefly responsible for managing the community, including content moderation and resolving user issues.
- User: They mainly perform basic operations such as account login and registration.

To gain a more detailed understanding of the needs and expectations of each role, we have provided relevant user stories in **Appendix A**. Additionally, to prioritize various functionalities and requirements, Moscow prioritization is listed in **Appendix B**.

Chapter 3 Design and Implementation

3.1 Design

3.1.1 Technology Stack



Figure 1: Technology Stack

As illustrated in **Figure 1**, the technology stack chosen for this project reflects the prevailing technologies and trends in the current development community. We adhere to the development paradigm of separating the front end and back end to ensure flexibility and maintainability.

Frontend Technologies:

HTML, CSS, and JavaScript: These three technologies form the foundation of modern web development. Specifically, HTML defines the content structure, CSS manages style and layout, and JavaScript provides interactive functionality.

Vue.js: Adopted as a front-end framework, Vue.js is widely embraced due to its lightweight nature, reactive data-binding, component-based design, and user-friendly learning curve. It enables developers to quickly create interactive and maintainable web applications.

Backend Technologies:

Java: This powerful backend programming language allows us seamless integration with various enterprise services such as database connections, messaging queues, and caching.

Spring Boot: An essential component in the Java ecosystem, Spring Boot significantly simplifies backend development. With features like rapid

development, auto-configuration, support for microservices, and deep integration with the Spring framework, it offers a solid foundation for building stable and scalable applications.

Data Management and Storage:

MySQL: As a highly-regarded open-source relational database management system, MySQL offers us a high-performance data storage solution, perfectly meeting the data management needs of the project.

MyBatis: This DAO (Data Access Object) framework offers a simplified way for interactions between Java objects and relational databases. Its flexible mapping, SQL optimization capabilities, and support for dynamic SQL make database operations more efficient.

Redis: As an outstanding in-memory database, Redis delivers fast read-write capabilities and versatile data structures. It also includes rich features for caching and real-time data processing, making it highly suitable for applications that require rapid data access and caching.

In summary, this technology stack provides our project with robust support, ensuring the application's performance and reliability.

3.1.2 System Architecture

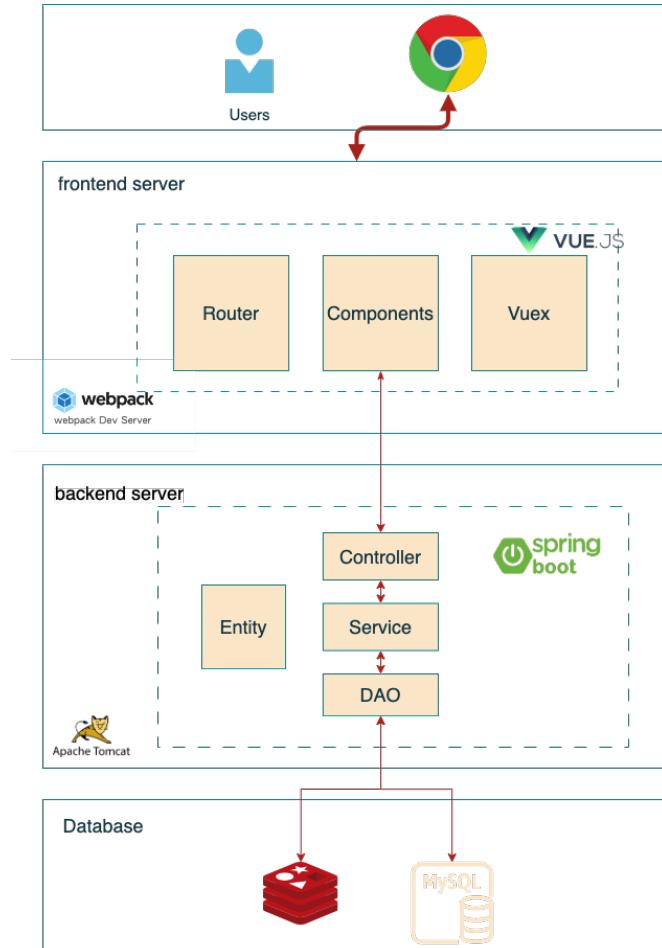


Figure 2: System Architecture

As illustrated in **Figure 2**, a decoupled front-end and back-end architecture is utilised in this project, ensuring clear separation between the UI and business logic. Here's an overview of the system's workflow and key components:

Firstly, When a user interact with the app via a browser. the browser initiates an HTTP request. This request is first routed to the Vue CLI's **Webpack dev server**. Based on its configuration, the server determines if the request should be proxied to the back end. For example, requests beginning with "/api" are forwarded to the backend server through a proxy set in the "vue.config.js".

Secondly, On the back-end side, Spring Boot's in-built server **Tomcat** processes the request from the front-end. Depending on the route of the request, the relevant **Controller** is invoked. The data flow moves from the **Controller** to the **Service** and eventually reaches the database via the **DAO**. It might involve databases like **MySQL** and **Redis**. The system first checks for cached data in Redis. If the cache is not hit, it resorts to querying the **MySQL** database. For efficiency, based on predefined rules, fetched data might be cached to reduce future IO delays, enhancing response speed. Once the data processing concludes, **Spring Boot** generates a response that's sent back to the front-end server.

Lastly, upon receiving the response from the back end, the **Webpack dev server** relays this response to the browser. As the browser processes the incoming data, the UI components, powered by Vue.js, undergo updates based on this fresh data. As a result, users witness updated information or get relevant feedback in their browsers.

3.1.3 Database Design

For a note application centred on a knowledge community, a rational, efficient, and easily scalable database structure is crucial. This database structure is designed based on the core functionality of the software and the needs of the users. It encompasses multiple entities, ensuring the completeness of features and the association between data.

The main entities involved in this application are User, Note, Post Note, Tag, Comment, Like, and Notification(**Figure 3**). There is a field named Content with a data type of LONGTEXT in Note and Post Note. Such a design is based on the following considerations:

- 1)**Rich Text Support:** HTML is used as the storage format, allowing users to flexibly use various text formats in their notes, such as bold, italic, hyperlinks, and images, further enriching the note content.
- 2)**Cross-Platform Consistency:** As a universal markup language, HTML ensures the consistency of content displayed across different devices and platforms.
- 3)**Rendering Efficiency:** HTML content can be quickly rendered by modern web browsers and mobile apps, ensuring a smooth reading experience for users.

The tables involved in the database are:

comment_table: Records note comments, including commenter ID, comment content, and time, etc.

like_table: Tracks user likes on notes.

notification_table: Logs user-related notifications, such as comment or like alerts.

post_note_table and note_table: While both tables store notes, post_note_table primarily holds notes published in the community, whereas note_table leans more towards private notes or drafts.

tag_table: Stores tags related to notes, facilitating user categorization and retrieval.

user_table: Saves basic user information, such as username, password, and email.

note_tag and post_tag: These two junction tables respectively link note_table to tag_table and post_note_table to tag_table, establishing a many-to-many relationship between notes and tags.

The database design of this note application aims to achieve maximum flexibility and scalability.

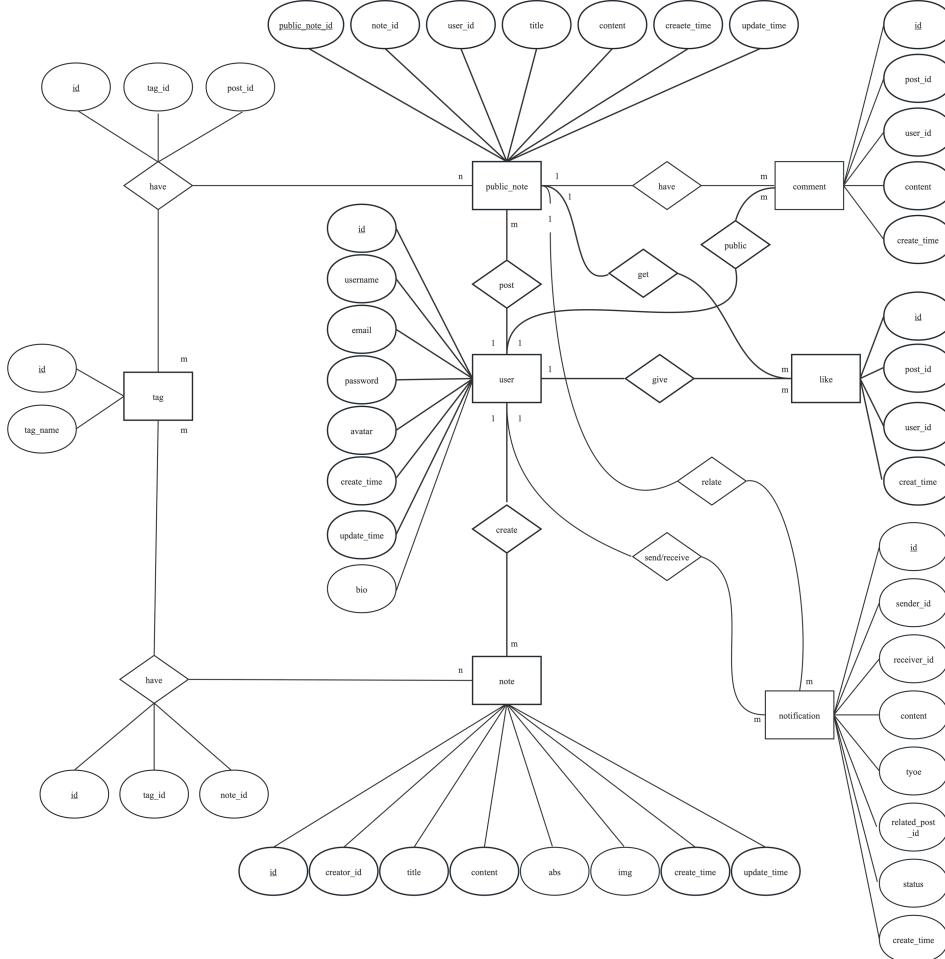


Figure 3: E-R Diagram

3.1.4 Application User Interface Design

Mindiqo emphasizes a design philosophy that's simple and intuitive. An intuitive user interface not only enhances the user experience but also makes creating, editing, and sharing notes effortlessly.

As is shown in **Figure 4**, the main interface of the app is structured. On the left side, there's a sidebar listing commonly used features like "My Notes" and "Home". This layout allows users to quickly locate and use their frequently needed functions. At the top, there's a permanent search bar, flanked by the user's avatar and notification icon. This design ensures users can easily search for content, access their profile, and view notifications promptly.

To design a user-friendly interface, a series of wireframes (**Appendix C**) are drawn out. On the "Home" page, users can freely browse, filter, and sort notes and also interact with notes shared by others. The "My Notes" page serves as the primary workspace, equipped with a feature-rich rich-text editor that supports various formatting tools like bold and italic. Moreover, to simplify note management and retrieval, the app incorporates a tagging system, making note categorization and searching efficient and straightforward.

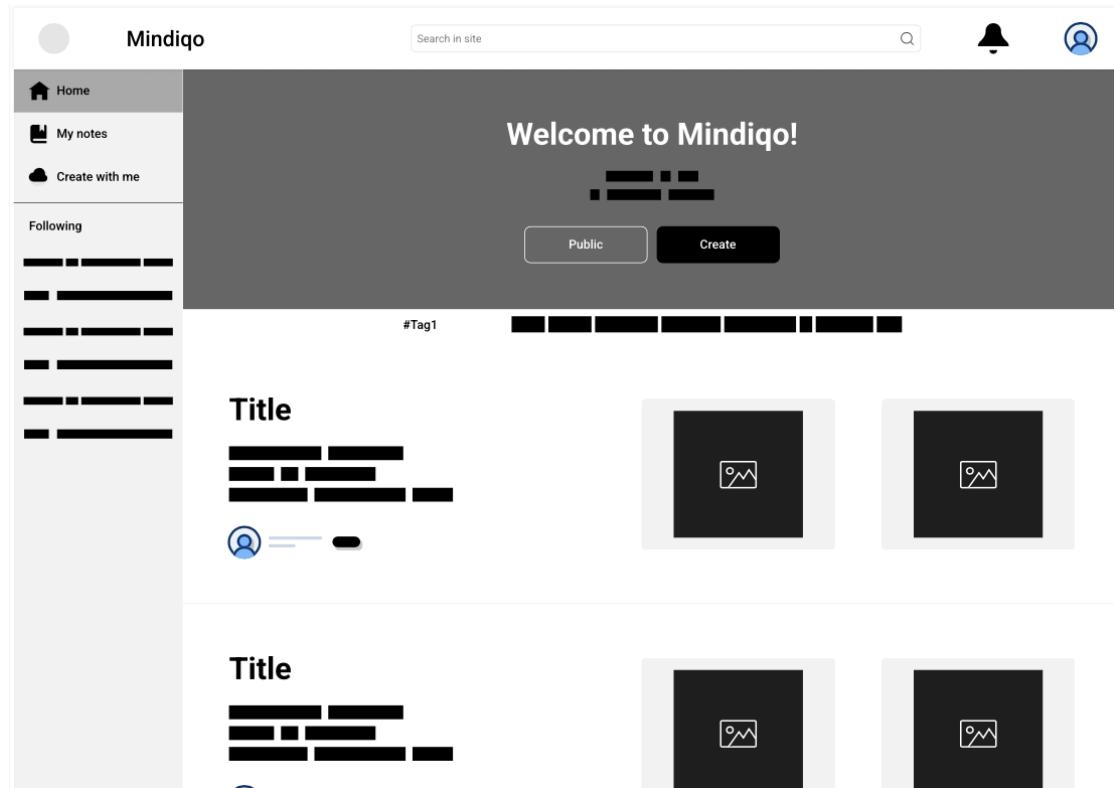


Figure 4: Home page wireframe

3.2 Implementation

The development of Mindiqo encompasses a wide range of aspects. Due to space constraints, this paper will not delve deeply into every detail but will highlight how the core functionalities of Mindiqo are implemented. Readers interested in the development code can access our source code directly from the project repository on GitLab.

3.2.1 User Authentication

User register: In the RegisterView.vue front end, there's a user registration form. Each input is validated with validateState('field'). If inputs are inadequate (e.g., weak password or long username), submission is blocked with an alert. Upon clicking register, the app sends a POST request to the backend server. If the database confirms no duplicate username, a success message returns; otherwise, it sends a "username already exists" error.

User login: On the login page, the frontend sends the username and password to the backend via a POST request. Upon receiving the data, the AuthController on the backend processes it. If the username and password match, a JWT-based token is generated and returned to the front end. JWT (JSON Web Token) is a compact, URL-safe credential that allows information to be passed between network requests and verifies the sender's HTTP (JSON Web Tokens, 2019). This token contains the username and avatar. Once the front end receives this token, it parses it to extract and store non-sensitive data like username and avatar within the app. The token is also stored in the browser's local storage. To ensure security, this token is included in the header of subsequent requests for backend authentication and validation.

User Permission Interception: In the main.js file of the frontend Vue project, a route guard is configured(**Figure 5**). Whenever a route change occurs, the system checks if the target page requires authentication. If the target page requires a login and the user is not logged in, the system redirects to the login page. Otherwise, if no login is needed or the user is already logged in, the system completes the route transition.

```
100 | //router guard
101 | router.beforeEach((to, from, next) => {
102 |   // check if the route requires authentication and user is not logged in
103 |   if (to.matched.some(record => record.meta.requiresAuth)) {
104 |     const userToken = localStorage.getItem('userToken');
105 |     if (!userToken) {
106 |       // if user is not logged in, redirect to login page.
107 |       next({
108 |         path: '/login',
109 |         query: { redirect: to.fullPath } // Store the full path to redirect the user to after login
110 |       });
111 |     } else {
112 |       // User is logged in, continue to route
113 |       next();
114 |     }
115 |   } else {
116 |     // Doesn't require auth, continue to route
117 |     next();
118 |   }
119 | });


```

Figure 5: Router guard

In the backend Spring Boot project, an interceptor named AuthInterceptor (**Figure 6**) is defined to intercept specific requests. The interceptor extracts the user's Token from the request header and validates it using a static method in JwtUtils. The request is only allowed if the Token is valid (i.e., it can be decoded and hasn't expired). If the Token fails to decode or is expired, the system will return an error message: "NOT LOGIN" or "TOKEN EXPIRED".

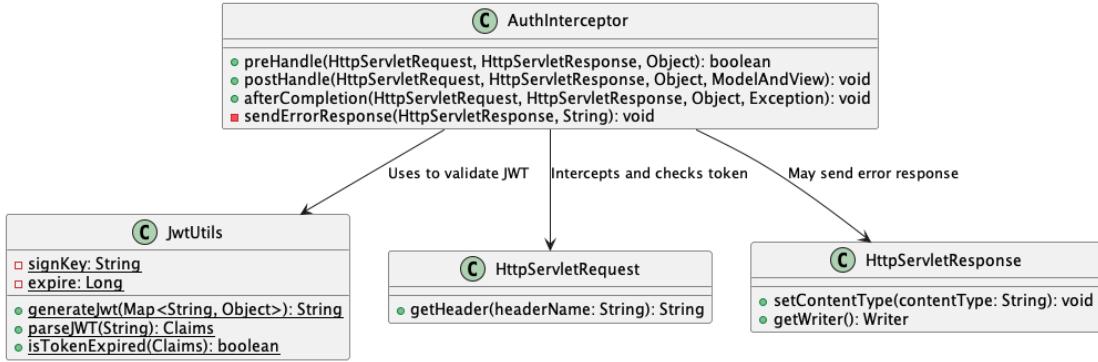


Figure 6: AuthInterceptor UML

3.2.2 Note-taking feature

Create: On the “/my_notes” page, there's a “New Note” button. Clicking it activates the pre-set method to create a new note, prompting a dialogue box for the title and summary input. After confirmation, the front end sends this information to the backend via a POST request to the “api/notes/add” API. The backend, upon receiving this data, creates a new entry in the notes database. During this, the backend uses MyBatis's primary key backfilling technique to populate the note's primary key generated by the database back into the note object. This note object, especially its primary key, is then returned to the front end along with a Result object. The front end, upon receiving the response, redirects to the note's editing page at “/my_notes/edit/{id}”, where “{id}” is the primary key of the new note.

Edit: In the frontend project's “EditorComponent”, an open-source editor named “element-tiptap” has been integrated. “Element-tiptap” is a rich-text editor built on Vue.js, harmoniously blending the strengths of both Element UI and tiptap libraries. It offers an extensive suite of text editing and formatting tools, encompassing common operations such as setting titles, bolding, italicizing, and linking (Leecason, 2022).

Save: After editing, the front end converts the note content to HTML format. On choosing to save, the front sends this HTML formatted content to “api/notes/{id}/save” via a POST request. The backend, upon receiving the request, uses the provided note “{id}” to update the corresponding item in the database.

Publish: When a user decides to publish a note, the front end first ensures the current content is saved. Once securely stored, it invokes the publishing method. This method creates a full copy of the current note and sends it to “api/notes/{id}/public” via POST. The backend, upon receipt, merges this copy with user data, forming a new “PostNote” object. This object, besides having the note's content, also carries additional related data like the publisher, tags, cover image, and summary. After its creation, the backend saves this “PostNote” object's data in the database.

	id	creator_id	title	content	create_time
1	21	8	juejuezi	<p>	2023-08-05 11:58:03
3	43	1	Microsoft AI	<p></p><p>	
 <id property="id" column="id"/>
 <result property="noteId" column="note_id"/>
 <result property="userId" column="user_id"/>
 <result property="title" column="title"/>
 <result property="content" column="content"/>
 <result property="abs" column="abs"/>
 <result property="createTime" column="create_time"/>
 <result property="updateTime" column="update_time"/>
 <result property="img" column="img"/>
 <result property="viewCount" column="view_count"/>
 <result property="likeCount" column="like_count"/>
 <result property="commentCount" column="comment_count"/>
 <association property="creator" javaType="com.mindigo.backend.entity.User">
 <id property="id" column="user_id"/>
 <result property="username" column="username"/>
 <result property="avatar" column="avatar"/>
 </association>
</resultMap>
```

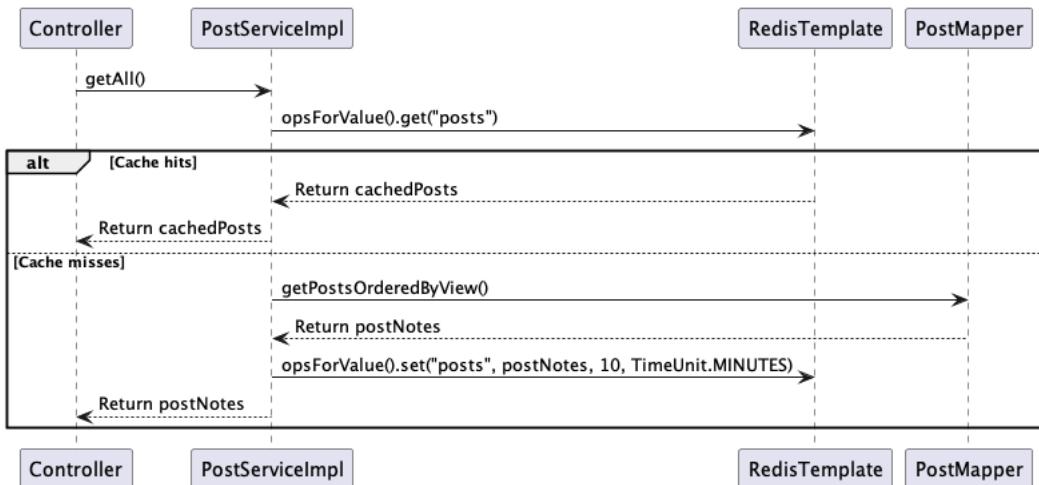
```

<select id="getPostsOrderedByView" resultMap="postNoteResultMap">
 SELECT pn.id,
 pn.note_id,
 pn.user_id,
 pn.title,
 pn.abs,
 pn.create_time,
 pn.update_time,
 pn.img,
 pn.view_count,
 u.username,
 u.avatar,
 (SELECT COUNT(*) FROM like_table lt WHERE lt.post_id = pn.id) as like_count,
 (SELECT COUNT(*) FROM comment_table ct WHERE ct.post_id = pn.id) as comment_count
 FROM post_note_table pn
 JOIN user_table u ON pn.user_id = u.id
 ORDER BY pn.view_count DESC, pn.create_time DESC
</select>
```

**Figure 7:** “resultMap” in Mybatis XML

**Post View:** When users visit “/home/post/{id}”, they will be presented with a specific post. As the page loads, the front end automatically triggers the “loadpost( )” method to fetch detailed content from the backend. It's noteworthy that the post's “content” is stored as HTML in the database. To display it correctly on the front-end, use the “v-html” directive of Vue, denoted as “v-html=“ post. content ””. This ensures the HTML content is rendered accurately on the page. Besides, the page also loads associated comments and like information. It checks if the user has already liked the post, determining how the like button or icon should be displayed.

**Redis Cache Integration:** To optimize the homepage response time, the backend service integrated Redis for data caching since the homepage content is frequently accessed by users. This avoids the overhead of constantly fetching data from the primary database and has significantly sped up data access. Specifically, in the backend service layer "PostServiceImpl", methods like getAll(), getById(), getMostViewed(), getMostLiked(), getRecent(), and getTags() all utilize Redis caching. If data exists in the cache, it's directly retrieved and returned; if not, it's fetched from the database and then stored in the Redis cache with a 10-minute expiration. A detailed workflow of the PostServiceImpl is illustrated in **Figure 8**.



**Figure 8:** Redis Cache Sequence Diagram

# Chapter 4 Testing & Evaluation

## 4.1 Backend Unit Test

In our Spring Boot backend project, to ensure the quality of the code, we have meticulously designed tests for each Service and Controller. Test classes were created for each of them, providing comprehensive testing for all functional methods.

To guarantee the purity of our unit tests and to eliminate interference from external dependencies, we utilized the “Mockito” mocking framework. This enabled us to simulate potential responses from external services or databases. This strategy focuses our tests on the core performance of the functional units, ensuring the code's behaviour is correct under predefined return scenarios.

For the controller layer, “MockMvc” was employed to simulate HTTP requests. “MockMvc” can not only mimic various HTTP requests like GET and POST, but it also offers thorough verification of their responses, examining aspects like HTTP status codes, response content, and other related details. This ensures the controller behaves as expected when faced with different HTTP requests.

To enhance the testing framework, assertion capabilities from Spring MVC Test have been integrated into the project. This addition allows for more than just basic method response verification; it enables a detailed examination of the structural elements within JSON response objects, as illustrated in **Figure 9**.

Finally, utilizing the testing features in IntelliJ IDEA to scan all the test classes within Spring Boot, conducted unified tests, and generated detailed test reports(**Figure 10**).

```
@Test
public void testLogin_Success() throws Exception {
 User testUser = new User();
 testUser.setUsername("exampleUser");
 testUser.setPassword("examplePassword");
 testUser.setId(123);
 testUser.setAvatar("exampleAvatar");
 when(authService.login(any(User.class))).thenReturn(testUser);

 mockMvc.perform(post(urlTemplate: "/api/login")
 .contentType("application/json")
 .content("{\"username\": \"exampleUser\", \"password\": \"examplePassword\"}")
 .andExpect(status().isOk())
 .andExpect(jsonPath(expression: "$.code").value(expectedValue: 1))
 .andExpect(jsonPath(expression: "$.msg").value(expectedValue: "success")));
}
```

**Figure 9:** Example of Test Method

## Test Results & Analysis

java in backend: 65 total, 65 passed		3.81 s
<a href="#">Collapse</a>   <a href="#">Expand</a>		
<b>BackendApplicationTests</b>		569 ms
<b>contextLoads()</b>	passed	569 ms
<b>NotificationControllerTest</b>		1.70 s
<b>testGetReadNotifications()</b>	passed	1.59 s
<b>testGetUnreadNotificationsCount()</b>	passed	17 ms
<b>testDeleteANotification()</b>	passed	15 ms
<b>testMarkANotificationAsRead()</b>	passed	65 ms
<b>testGetUnreadNotifications()</b>	passed	12 ms
<b>CommentControllerTest</b>		99 ms
<b>testComment()</b>	passed	69 ms
<b>testGetComments()</b>	passed	30 ms
<b>AuthControllerTest</b>		130 ms
<b>testLogin_Success()</b>	passed	51 ms
<b>testLogin_Failure()</b>	passed	23 ms
<b>testRegister_Success()</b>	passed	17 ms
<b>testResetPassword()</b>	passed	21 ms
<b>testRegister_UsernameExists()</b>	passed	18 ms
<b>PostServiceTest</b>		33 ms
<b>test GetById()</b>	passed	12 ms
<b>test GetTags()</b>	passed	4 ms
<b>test GetMostViewed()</b>	passed	4 ms
<b>test Delete()</b>	passed	5 ms
<b>test GetAll()</b>	passed	4 ms
<b>test Search()</b>	passed	4 ms

**Figure 10:** Unit Test Report

**Controller Tests:** The outcomes here indicate that all HTTP request handlers function as expected, with no significant flaws or anomalies.

**Service Tests:** Being pivotal for business logic and data handling in the application, these tests have also passed successfully, affirming the stability and reliability of the Service layer.

Given the above results, all unit tests have been successfully passed, signifying the application's stability and reliability under the current testing conditions.

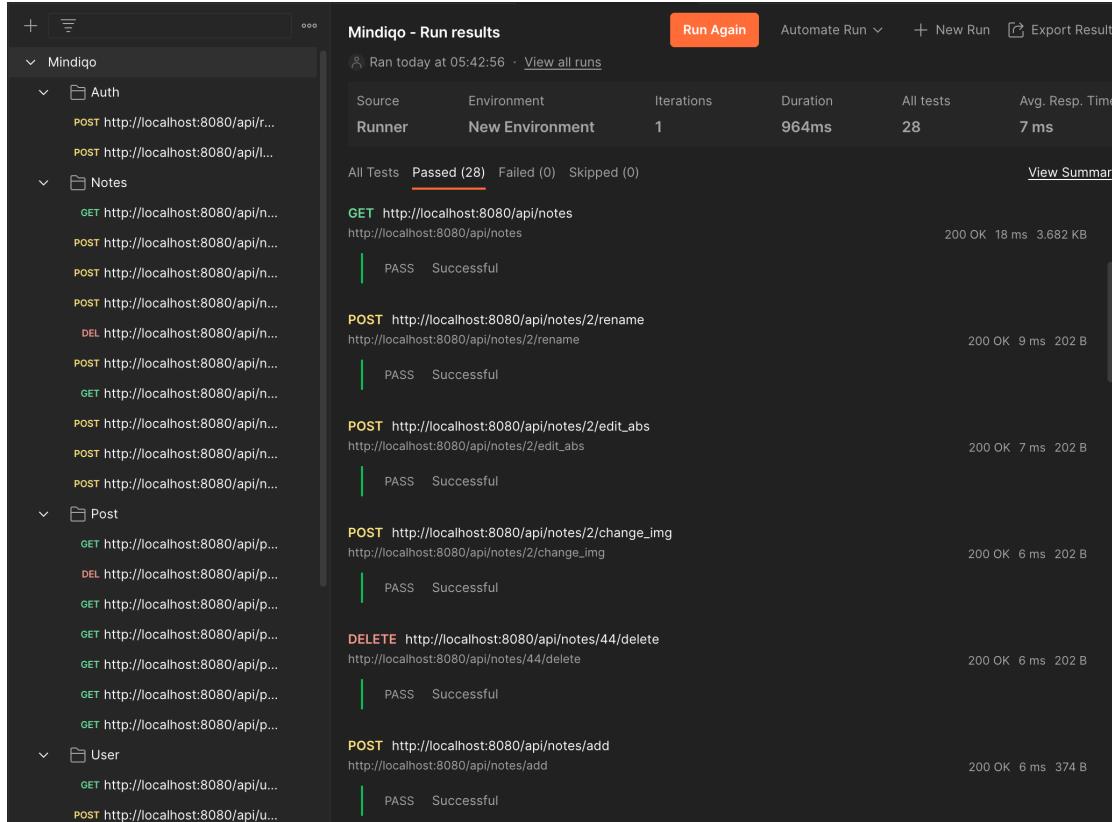
## 4.2 API Endpoint Testing

In this project, we employed Postman as the tool for API interface testing to ensure the accuracy and robustness of the code logic. Based on the existing API documentation, each request was configured to match the predetermined HTTP request methods, request URLs, and request bodies.

For those requests that necessitated tokens in the HTTP header, the feature of environment variables in Postman was adopted. This approach not only made management easier but also enhanced the degree of automation in our testing workflow. A valid token was stored in Postman's environment variables and uniformly used across all requests that required it.

Going further, meticulous verification tests were written for each request within Postman. These scripts were mainly aimed at checking the returned HTTP status codes, validating the content of the response bodies, and making other related test assertions.

Finally, by using the "Runner" feature in Postman, all pre-configured requests were executed in a single run, and detailed test results were generated accordingly.



**Figure 11: API Endpoint Test Result**

### Test Results & Analysis:

Throughout the testing phase, 28 API endpoints were assessed, covering all primary functionalities and data interaction points in the application. All tests successfully passed. In terms of response speed, an average response time of 7ms highlights the backend service's efficiency and degree of optimization.

### 4.3 User Evaluation Testing

In this project, nine participants were invited to test the Mindiqo application in a local environment. To ensure consistency and control variables during the testing, Chrome v116 was chosen as the sole browser for the evaluation.

Participants, following a predefined design, individually completed a series of tasks within the Mindiqo app. These tasks encompassed account registration, note editing, browsing and commenting on content in the knowledge community, web content searches, publishing notes, and editing personal profile information. These

tasks were designed to assess the application's performance across various functionalities.

After the testing was concluded, an online survey (**Appendix D**) was designed to gain deeper insights into evaluations and feedback on Mindiqo. This survey not only assessed Mindiqo's functionality, user experience, and interface design but also compared it with previously used applications. Particular attention was given to Mindiqo's performance in areas such as note-taking features, knowledge community functions, user interface design, and app interactivity. Furthermore, a specific section at the end of the survey was set up to gather suggestions and opinions on potential areas for improvement in the application. **Appendix E** contains the questionnaire responses.

### **Test Results & Analysis:**

The survey results illustrate that Mindiqo, a note-taking application, has generally been well-received by its users. A majority of respondents have prior experience with other note-taking apps. When compared with these apps, most users rated Mindiqo as 'better' or 'excellent'.

Crucially, its core functionality, which is the note-taking feature, was predominantly seen in a positive light. Additionally, the app's knowledge community feature stood out, with a significant number of users lauding it. This sense of community was further evident as many expressed willingness to share their notes within this platform.

Regarding usability, while the user interface (UI) design received mostly favorable feedback, the interaction with the app was seen as 'somewhat natural', indicating potential areas of improvement for a more seamless experience.

Some participants reported issues with the app. One comment mentioned that if an image in the notes is too wide, it might cause display problems.

Mindiqo is a good note-taking app praised for its community focus and user-friendly interface. While its core features are well-liked, some users have reported issues, suggesting potential areas for improvement

# Chapter 5 Conclusion

The primary objective of this project was to develop a note-taking application designed to serve as a comprehensive platform, incorporating features for online note-taking and community knowledge sharing. To align with this overarching goal, a Minimum Viable Product (MVP) has been successfully constructed and evaluated. The MVP encapsulates all core functionalities, including note creation, editing, storage, and community knowledge sharing. The project followed a rigorous lifecycle that encompassed design, implementation, testing, and evaluation phases. Thus, it successfully delivered on its initial aim of offering a holistic platform for note-taking and community engagement.

## 5.1 Summary

In the design and implementation phases, the aim of this project is to offer a robust note-taking software equipped with knowledge community and collaborative features, to facilitate superior knowledge management and sharing among users. Utilizing the current mainstream technology stack of Spring Boot, MySQL, and Vue, a Minimum Viable Product (MVP) has been successfully constructed. Core functionalities aligned with major user requirements have been implemented, as **Figure 12 and 13** indicated by the completion status of user stories.

No	User Category	Title	Priority	Cost/ day
1	Note Creator	Viewing Notes I Created	Must	0.5
2	Note Creator	Creating a New Note	Must	1
3	Note Creator	Editor	Must	3
4	Note Creator	Editor: Text Format Control	Could	1
5	Note Creator	Editor: List	Could	0.5
6	Note Creator	Editor: Markdown Support	Should	1.5
7	Note Creator	Editor: Insert Image	Should	0.5
8	Note Creator	Editor: Insert External Link	Could	0.5
9	Note Creator	Add Tags to Note	Should	0.5
10	Note Creator	Save Note	Must	3
11	Note Creator	Renaming a Note	Should	0.5
12	Note Creator	Deleting a Note	Must	0.5
13	Note Creator	Organizing Notes	Could	1.5
14	Note Creator	Searching for Notes	Should	1.5
15	Note Creator	Posting a Note	Must	3
16	Note Creator	Initiating Collaboration	Would	3
17	Poster	Viewing Posted Notes	Must	1
18	Poster	Canceling Post	Must	0.5
19	Community Member	Editing Profile	Could	0.5
20	Community Member	Viewing Profiles	Could	0.5
21	Community Member	Browsing Note Summaries	Must	0.5

**Figure 12: List of User Stories(1)**

22	Community Member	Viewing Specific Note Contents	Must	0.5
23	Community Member	Viewing Interactions on a Note	Could	1
24	Community Member	Commenting on a Note	Could	1
25	Community Member	Replying to a Comment	Could	0.5
26	Community Member	Like a Note	Could	0.5
27	Community Member	Message Notifications	Could	0.5
28	Community Member	Searching for Information	Should	0.5
29	Community Member	Searching for Information	Should	0.5
30	Collaborator	Viewing Collaboration Requests	Would	2
31	Collaborator	Editing a Collaborative Note	Would	3
32	Collaborator	Viewing Changes	Would	2
33	Admin	Managing Users	Should	2
34	Admin	Moderating Community Posts	Should	0.5
35	User	User Registration	Must	1
36	User	User Login	Must	0.5
37	User	Password Reset	Must	0.5

completed: 

**Figure 13: List of User Stories(2)**

The application underwent rigorous testing, including backend unit tests, API endpoint tests, and user evaluation questionnaires. The results indicate a high level of stability, reliability, and user satisfaction. The note-taking and knowledge community features were particularly well-received, although some areas for improvement were identified, such as UI/UX enhancements.

## 5.2 Limitations and Challenges

While the project has achieved its primary objectives, there are some limitations and challenges:

1. Scalability: As the user base grows, the system may require further optimizations to handle increased traffic and data storage needs.
2. Mobile Responsiveness: The application is primarily designed for web browsers, and its adaptability to mobile devices has not been fully explored.
3. Feature: Additional features like real-time collaboration, offline access, and advanced search capabilities could further enhance the application.

## 5.3 Future Work

In the face of time constraints, not all desired features could be implemented. Future work should focus on these areas:

**Note Collaboration:** In upcoming iterations, priority should be given to the implementation of note collaboration features.

**Display Issues:** Refinement of the visual presentation of wide images within the notes section to improve user experience.

**Community Features:** Additional functionalities such as upvoting, downvoting, and "following" should be incorporated to cultivate a more interactive and engaging community experience.

## Chapter 6 References

- Dean, D., & Webb, C. (2011). Recovering from information overload.
- Filickova, M. (2016). DANIEL J. LEVITIN, THE ORGANIZED MIND: THINKING STRAIGHT IN THE AGE OF INFORMATION OVERLOAD, PENGUIN BOOK 2015, SS. 528. *Civitas et Lex*, 12(4), 91-94.  
<https://doi.org/10.31648/cetl.2345>
- Leecason. (2022, July 8). element-tiptap: A Vue 2 wrapper of tiptap a renderless rich-text editor [Computer software]. GitHub.  
<https://github.com/Leecason/element-tiptap>
- Jefferson, T. L. (2006). Taking it personally: personal knowledge management. *VINE*, 36(1), 35-37. <https://doi.org/10.1108/03055720610667345>
- Razmerita, L., Kirchner, K., & Sudzina, F. (2009). Personal knowledge management. *Online Information Review*, 33(6), 1021-1039.  
<https://doi.org/10.1108/14684520911010981>
- Roetzel, P. G. (2019). Information overload in the information age: a review of the literature from business administration, business psychology, and related disciplines with a bibliometric approach and framework development. *Business Research*, 12(2), 479-522. <https://doi.org/10.1007/s40685-018-0069-z>
- JSON Web Tokens. (n.d.). Retrieved August 21, 2023, from <https://jwt.io>
- Tiwari, 2022, KNOWLEDGE MANAGEMENT STRATEGIES AND EMERGING TECHNOLOGIES - AN OVERVIEW OF THE UNDERPINNING CONCEPTS

## Appendix A User Stories

#1	Viewing Notes I Created	Priority : Must	Cost : 0.5 day
<b>Story Description:</b>			
<p><b>As a Note Creator,</b> <b>I want</b> to be able to view my created note, <b>So that</b> I can easily access and manage my notes.</p>			
<b>Acceptance Criteria:</b>			
<p><b>Given</b> that I am on the "My Notes" page, <b>When</b> I look at the sidebar, <b>Then</b> I should see a list of all the notes I have created. <b>When</b> I click on a note from the sidebar, <b>Then</b> the system should navigate me to the specific page of that note.</p>			

#2	Creating a New Note	Priority : Must	Cost : 1 day
<b>Story Description:</b>			
<p><b>As a Note Creator,</b> <b>I want</b> create a note, <b>So that</b> I can document my thoughts or ideas.</p>			
<b>Acceptance Criteria:</b>			
<p><b>Given</b> that I am on the "My Notes" page, <b>When</b> I click on 'New', <b>Then</b> the app should present me with a blank note template. <b>Then</b> the note template should provide a title field where I can enter a title for the note or not.</p>			

#3	Editor	Priority : Must	Cost : 3 day
----	--------	-----------------	--------------

**Story Description:**

**As a Note Creator,**  
**I want** to be able to input and edit text within the opened note page,  
**So that** I can generate and refine the content of my note.

**Acceptance Criteria:**

**Given** that I am on an open note page,  
**When** I start typing or modifying existing text,  
**Then** the app should allow me to input and edit the text content of the note.

#4

Editor: Text format Control

Priority : Could

Cost : 1 day

**Story Description:**

**As a Note Creator,**  
**I want** to be able to change the style of the text in my note,  
**So that** I can make my note more readable, organized, and visually appealing.

**Acceptance Criteria:**

**Given** that I am editing a note,  
**When** I select a portion of the text,  
**Then** I should be able to change the style of the selected text (such as bold, italic, underline, color, etc.) by click the buttons on the toolbar or shorcuts.

#5

Editor : List

Priority : Could

Cost : 0.5 day

**Story Description:**

**As a Note Creator,**  
**I want** to be able to create a series of lists,  
**So that** I can organize and structure my content effectively.

#### Acceptance Criteria:

**Given** that I am editing a note,  
**When** I select the "bulleted list"/"numbered list" option from the toolbar and enter an item,  
**Then** the item should be displayed as a bullet/numbered point in a list, and pressing enter should create a new bullet/numbered point on a new line,  
**When** I press enter after a list item,  
**Then** a new line should be created, and the list should automatically update

#6	Editor: Markdown Support	Priority : Should	Cost : 1.5 day
----	--------------------------	-------------------	----------------

#### Story Description:

**As a Note Creator,**  
**I want** the editor to support Markdown syntax  
**so that** it can help me quickly input formatted information.

#### Acceptance Criteria:

**Given** that I am editing a note  
**When** I type the corresponding markdown keywords,  
**Then** the editor will automatically update the text to the corresponding format.

#7	Editor: Insert Image	Priority : Should	Cost : 0.5 day
----	----------------------	-------------------	----------------

#### Story Description:

**As a Note Creator,**  
**I want** to be able to insert images into the editor at specific positions,  
**So that** I can enriching the content of notes

#### Acceptance Criteria:

**Given** that I am editing a note  
**When** I click on the "Insert Image" button in the toolbar,  
**Then** a dialog box should appear, allowing me to choose an image file from my device or the URL of an image from the web.

#8	<b>Editor: Insert External Link</b>	Priority : Could	Cost : 0.5 day
----	-------------------------------------	------------------	----------------

#### **Story Description:**

**As a Note Creator,**  
**I want** to be able to insert external links into the editor,  
**So that** reference relevant information from other sources.

#### **Acceptance Criteria:**

**Given** that I am editing a note  
**When** I select a portion of text,  
**And** I click on the "Insert Link" button in the toolbar,  
**Then** a dialog box should appear, allowing me to enter the URL for the external link.

#9	<b>Add Tags to Note</b>	Priority : Should	Cost : 0.5 day
----	-------------------------	-------------------	----------------

#### **Story Description:**

**As a Note Creator,**  
**I want** to be able to add tags to my notes.  
**So that** I can organize and categorize my notes based on common or custom labels.

#### **Acceptance Criteria:**

**Given** that I am editing or viewing a note,  
**When** I click on the "Add Tag" button,  
**And** I click on the "Insert Link" button in the toolbar,  
**Then** a dialog box should appear, allowing me to enter a new tag or select an existing tag.

#10	<b>Save Note</b>	Priority : Must	Cost : 3 day
-----	------------------	-----------------	--------------

#### **Story Description:**

**As a Note Creator,**  
**I want** the ability to save my notes,  
**so that** I can access them later and keep them securely.

#### Acceptance Criteria:

**Given** that I am editing or creating a note,  
**When** I make any changes to the note's content,  
**Then** the changes should be automatically saved to the server in the background, , overwriting the previous version of the note.  
**When** the automatic saving process is in progress,  
**Then** a visual indicator should be displayed to show that the note is being saved.  
**When** the note was saved,  
**Then** the visual idicator should be removed.  
**When** there is a connectivity issue or server error during the automatic saving process,  
**Then** an error message should be displayed, notifying the user that the note could not be saved.

#11	Renaming a Note	Priority : Should	Cost : 0.5 day
-----	-----------------	-------------------	----------------

#### Story Description:

**As a Note Creator,**  
**I want** to be able to rename my note,  
**So that** I can update the name of my note as per the content changes or for better organization.

#### Acceptance Criteria:

**Given** that I am on the "My Notes" page,  
**When** I click on the "..." button next to a note icon and select "rename",  
**Then** I should be able to input a new name for the note and save it.

#12	Deleting a Note	Priority : Must	Cost : 0.5 day
-----	-----------------	-----------------	----------------

#### Story Description:

**As a Note Creator,**  
**I want** to be able to delete my note,  
**So that** I can manage my notes and remove the ones I no longer need.

**Acceptance Criteria:**

**Given** that I am on the "My Notes" page,

**When** I click on the "..." button next to a note icon and select "delete",

**Then** the note should be removed from my list of notes after a confirmation prompt.

#13	Organizing Notes	Priority : Could	Cost : 1.5 day
-----	------------------	------------------	----------------

**Story Description:**

**As a Note Creator,**

**I want** to categorize and archive my notes,

**So that** I can better organize my notes and make it easier to retrieve specific information.

**Acceptance Criteria:**

**Given** that I am on a note page or the "My Notes" page,

**When** I create a link within a note to another note or click on the "..." button next to the note icon and select "move",

**Then** the system should allow me to nest notes hierarchically and move my notes to the specified location, enabling me to view the hierarchical relationship of my notes by clicking the downward arrow in the "My Notes" view.

#14	Searching for Notes	Priority : Should	Cost : 1.5 day
-----	---------------------	-------------------	----------------

**Story Description:**

**As a Note Creator,**

**I want** to be able to search for my notes by inputting keywords,

**So that** I can quickly find the note I need.

**Acceptance Criteria:**

**Given** that I am on the "My Notes" page,

**When** I input keywords into the search bar,

**Then** the system should return results that match the title, tags, or content of the notes with the inputted keywords.

#15	Posting a Note	Priority : Must	Cost : 3 day
-----	----------------	-----------------	--------------

#### Story Description:

**As a Note Creator,**  
**I want** to be able to post my own notes in the community,  
**So that** I can share my knowledge with the community.

#### Acceptance Criteria:

**Given** that I am on the "My Notes" or "Knowledge Community" page,  
**When** I select a note and choose to post it or click on the "+" icon to post a note I've created,  
**Then** the system should publish the selected note in the community, preventing further edits to the published version but allowing me to edit a copy of the original note.

#16	Initiating Collaboration	Priority : Would	Cost : 3 day
-----	--------------------------	------------------	--------------

#### Story Description:

**As a Note Creator,**  
**I want** to be able to initiate a collaboration,  
**So that** I can invite other users to co-create and enrich the content of my notes.

#### Acceptance Criteria:

**Given** that I am on the current note page or the "Create with me" page,  
**When** I select a note and choose to collaborate or click on "+" and select a note to initiate a collaboration,  
**Then** the system should allow me to invite other users to co-create the selected note with me.

#17	Viewing Posted Notes	Priority : Must	Cost : 1 day
-----	----------------------	-----------------	--------------

#### Story Description:

**As a Poster.**  
**I want** to be able to view my posted notes by clicking on my avatar in the "Knowledge Community" page,  
**So that** I can keep track of what I have shared with the community.

**Acceptance Criteria:**

**Given** that I am in the "Knowledge Community" section,  
**When** I click on my avatar,  
**Then** I should be redirected to my personal profile where I can see all the notes that I have posted.

#18	Canceling Post	Priority : Must	Cost : 0.5 day
-----	----------------	-----------------	----------------

**Story Description:**

**As a Poster,**  
**I want** to be able to delete a note that I've previously shared with the community,  
**So that** I can manage the content that I have made public.

**Acceptance Criteria:**

**Given** that I am viewing my posted notes,  
**When** I select a note and choose to unpost,  
**Then** that note should be removed from the public view in the community after a confirmation prompt.

#19	Editing Profile	Priority : Could	Cost : 0.5 day
-----	-----------------	------------------	----------------

**Story Description:**

**As a Community Member,**  
**I want** to be able to edit my personal profile,  
**So that** I can manage and update my personal information and display preferences in the community.

**Acceptance Criteria:**

**Given** that I am on my personal profile page,  
**When** I choose to edit my profile,  
**Then** I should be able to update my personal information and save the changes.

#20	<b>Viewing Profiles</b>	Priority : Could	Cost : 0.5 day
-----	-------------------------	------------------	----------------

#### **Story Description:**

**As a Community Member,**  
**I want** to be able to view the personal profiles of other members, including their personal introductions and the notes they have published,  
**So that** I can better understand their interests and knowledge areas.

#### **Acceptance Criteria:**

**Given** that I am browsing the community,  
**When** I click on the avatar of another member/myself,  
**Then** I should be directed to their personal profile where I can view their introduction and the notes they have published.

#21	<b>Browsing Note Summaries</b>	Priority : Must	Cost : 0.5 day
-----	--------------------------------	-----------------	----------------

#### **Story Description:**

**As a Community Member,**  
**I want** to be able to browse the summaries of notes shared by others on the homepage,  
**So that** I can get an overview of the shared knowledge and decide which notes I want to read in full.

#### **Acceptance Criteria:**

**Given** that I am in the "Knowledge Community" page,  
**When** I browse through the shared notes, ,  
**Then** I should be able to see a list of notes with a image and tags and titles .

#22	<b>Viewing Specific Note Contents</b>	Priority : Must	Cost : 0.5 day
-----	---------------------------------------	-----------------	----------------

**Story Description:**

**As a Community Member,**  
**I want** to be able to click on a note to view its full content,  
**So that** I can fully understand and learn from the knowledge shared in the note.

**Acceptance Criteria:**

**Given** that I am browsing shared notes,  
**When** I click on a specific note,  
**Then** I should be directed to a page that displays the full content of the note.

#23

Viewing Interactions on a Note

Priority : Could

Cost : 1 day

**Story Description:**

**As a Community Member,**  
**I want** to be able to click on a note to view the interactions associated with it, such as comments and likes by other members,  
**So that** I can gauge the community's response to the note and engage in the discussion if I choose.

**Acceptance Criteria:**

**Given** that I am viewing a specific note,  
**When** I scroll to the bottom of the page,  
**Then** I should be able to see comments and likes by other members on that note

#24

Commenting on a Note

Priority : Could

Cost : 1 day

**Story Description:**

**As a Community Member,**  
**I want** to be able to comment on a note at the bottom of the note content page,  
**So that** I can share my thoughts, ask questions, or provide feedback to the author and engage with the community.

**Acceptance Criteria:**

**Given** that I am viewing a note,  
**When** I scroll to the bottom of the note content page,  
**Then** I should be able to input and post a comment.

#25	Replies to a Comment	Priority : Could	Cost : 0.5 day
-----	----------------------	------------------	----------------

**Story Description:**

**As** a Community Member,  
**I want** to be able to reply to comments made by others,  
**So that** I can engage in discussion, ask further questions, or provide additional insights.

**Acceptance Criteria:**

**Given** that I am viewing comments on a note,  
**When** I select a specific comment and click reply,  
**Then** I should be able to post a reply to that specific comment.

#26	Like a Note	Priority : Could	Cost : 0.5 day
-----	-------------	------------------	----------------

**Story Description:**

**As** a Community Member,  
**I want** to be able to like a note,  
**So that** I can show my appreciation for the note and provide positive feedback to the author.

**Acceptance Criteria:**

**Given** that I am viewing a note,  
**When** I click on the "like" button at the bottom of the note,  
**Then** my "like" should be registered and the total count of likes for the note should increase.

#27	Message Notifications	Priority : Could	Cost : 0.5 day
-----	-----------------------	------------------	----------------

**Story Description:**

**As a Community Member,**  
**I want** to be notified when I receive a new message(likes, comments, replies).  
**So that** I can keep up with the community interactions involving me.

**Acceptance Criteria:**

**When** there are new interactions related to me,  
**Then** the notification center should update with these interactions and display a red dot with the number of unread messages.

#28	Searching for Information	Priority : Should	Cost : 0.5 day
-----	---------------------------	-------------------	----------------

**Story Description:**

**As a Community Member,**  
**I want** to be able to view all messages related to me in the notification center,  
**So that** I can keep track of all interactions related to my activities in the community.

**Acceptance Criteria:**

**When** I click on the notification button,  
**Then** I should be accessed to notification centre where i can see all messages related to me,  
**When** I click on a specific notification,  
**Then** I should be redirected to the relevant comment or note that the notification is referring to.

#29	Searching for Information	Priority : Should	Cost : 0.5 day
-----	---------------------------	-------------------	----------------

**Story Description:**

**As a Community Member,**  
**I want** to be able find the information I need,  
**So that** I can efficiently navigate and access relevant notes, tags, and users within the community.

**Acceptance Criteria:**

**Given** that I am on "Knowledge Community",  
**When** I enter keywords in the searching bar and initiate a search,  
**Then** the platform should return the matching notes, tags, and users based on the keyword.

#30	Viewing Collaboration Requests	Priority : Would	Cost : 2 day
-----	--------------------------------	------------------	--------------

**Story Description:**

**As** a Collaborator,  
**I want** to be able to view collaboration requests in the "Create with me" page,  
**So that** I can manage and respond to the requests for collaboration on various notes.

**Acceptance Criteria:**

**When** I am in the "Create with me" page,  
**Then** I should be able to view all the collaboration requests I have received, including details of the note and the user who sent the request.

#31	Editing a Collaborative Note	Priority : Would	Cost : 3 day
-----	------------------------------	------------------	--------------

**Story Description:**

**As** a Collaborator,  
**I want** to be able to edit the note I am collaborating on,  
**So that** I can add my insights and knowledge to improve the note.

**Acceptance Criteria:**

**Given** that I am a collaborator on a note,  
**When** I access the note,  
**Then** I should be able to edit and save changes to the note.

#32	<b>Viewing Changes</b>	Priority : Would	Cost : 2 day
<b>Story Description:</b>			
<p><b>As a Collaborator,</b>  <b>I want</b> to be able to see the changes made by other collaborators,  <b>So that</b> I can understand their contributions and build on them.</p>			

#33	<b>Managing Users</b>	Priority : Should	Cost : 2 day
<b>Story Description:</b>			
<p><b>As an Admin,</b>  <b>I want</b> to be able to view, edit, and manage user accounts,  <b>So that</b> I can ensure the smooth functioning of the community.</p>			

#34	<b>Moderating Community Posts</b>	Priority : Should	Cost : 0.5 day
-----	-----------------------------------	-------------------	----------------

#### **Story Description:**

**As** an Admin,  
**I want** to be able to moderate community posts,  
**So that** I can ensure that all content aligns with the community guidelines.

#### **Acceptance Criteria:**

**Given** that I am logged in as an admin,  
**When** I review community posts,  
**Then** I should have the ability to delete posts, hide posts, or flag posts that violate the community guidelines.

#35	User Registration	Priority : Must	Cost : 1 day
-----	-------------------	-----------------	--------------

#### **Story Description:**

**As** a new user,  
**I want** to be able to register an account,  
**So that** I can have personal access to the application and its features.

#### **Acceptance Criteria:**

**Given** that I am a new user,  
**When** I go to the registration page,  
**Then** I should be able to enter my personal details (like email, password, etc.) and upon successful validation and submission, an account should be created for me.

#36	User Login	Priority : Must	Cost : 0.5 day
-----	------------	-----------------	----------------

#### **Story Description:**

**As** a registered user,  
**I want** to be able to log into my account,  
**So that** I can access my personal data and interact with the application.

**Acceptance Criteria:**

**Given** that I am a registered user,  
**When** I go to the login page,  
**Then** I should be able to enter my login credentials and, upon successful validation, access my personal account.

#37	Password Reset	Priority : Must	Cost : 0.5 day
-----	----------------	-----------------	----------------

**Story Description:**

**As** a registered user,  
**I want** to be able to reset my password,  
**So that** I can regain access to my account if I forget my password.

**Acceptance Criteria:**

**When** I go to the login page and select the 'Forgot Password' option,  
**Then** I should be able to enter my email and receive a password reset link, and upon clicking the link and entering a new password, my password should be reset.

## Appendix B Moscow Prioritization

No	User Category	Title	Priority	Cost/ day
1	Note Creator	Viewing Notes I Created	Must	0.5
2	Note Creator	Creating a New Note	Must	1
3	Note Creator	Editor	Must	3
4	Note Creator	Editor: Text Format Control	Could	1
5	Note Creator	Editor: List	Could	0.5
6	Note Creator	Editor: Markdown Support	Should	1.5
7	Note Creator	Editor: Insert Image	Should	0.5
8	Note Creator	Editor: Insert External Link	Could	0.5
9	Note Creator	Add Tags to Note	Should	0.5
10	Note Creator	Save Note	Must	3
11	Note Creator	Renaming a Note	Should	0.5
12	Note Creator	Deleting a Note	Must	0.5
13	Note Creator	Organizing Notes	Could	1.5
14	Note Creator	Searching for Notes	Should	1.5
15	Note Creator	Posting a Note	Must	3
16	Note Creator	Initiating Collaboration	Would	3
17	Poster	Viewing Posted Notes	Must	1
18	Poster	Canceling Post	Must	0.5
19	Community Member	Editing Profile	Could	0.5
20	Community Member	Viewing Profiles	Could	0.5
21	Community Member	Browsing Note Summaries	Must	0.5
22	Community Member	Viewing Specific Note Contents	Must	0.5
23	Community Member	Viewing Interactions on a Note	Could	1
24	Community Member	Commenting on a Note	Could	1
25	Community Member	Replying to a Comment	Could	0.5
26	Community Member	Like a Note	Could	0.5
27	Community Member	Message Notifications	Could	0.5
28	Community Member	Searching for Information	Should	0.5
29	Community Member	Searching for Information	Should	0.5
30	Collaborator	Viewing Collaboration Requests	Would	2
31	Collaborator	Editing a Collaborative Note	Would	3
32	Collaborator	Viewing Changes	Would	2
33	Admin	Managing Users	Should	2
34	Admin	Moderating Community Posts	Should	0.5
35	User	User Registration	Must	1
36	User	User Login	Must	0.5
37	User	Password Reset	Must	0.5

## Appendix C Wireframe

Home page without login

This wireframe shows the layout of the Mindiqo home page for users who are not logged in. The top navigation bar includes the 'Mindiqo' logo, a search bar, and a user icon. On the left, a sidebar contains links for 'Home', 'My notes', 'Create with me', and a 'Login' button. The main content area features a 'Welcome to Mindiqo!' message, a user profile placeholder, and two large cards labeled 'Title' with small icons. Below these are two more 'Title' cards.

Home page

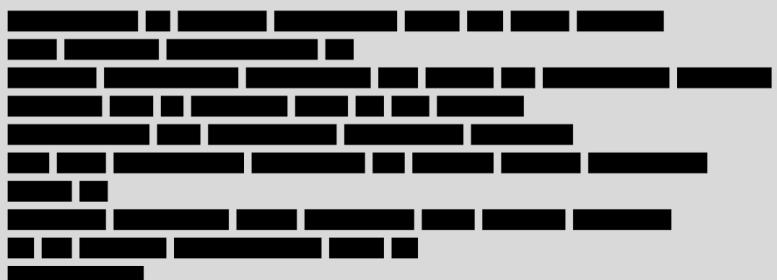
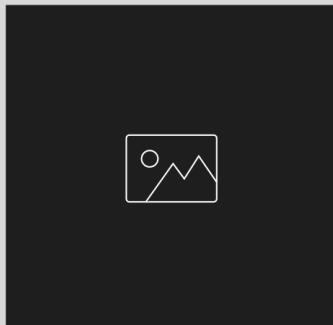
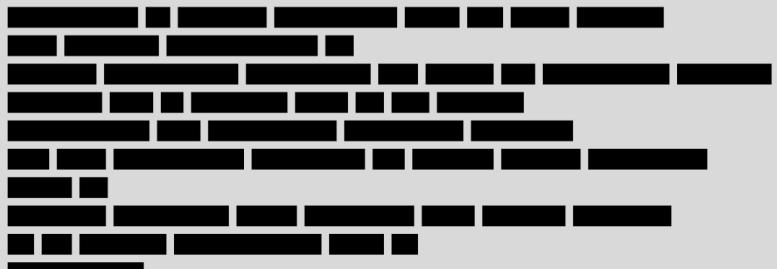
This wireframe shows the layout of the Mindiqo home page for logged-in users. It includes a user profile picture in the top navigation bar, along with a search bar and a user icon. The sidebar now includes a 'Following' section showing a list of items. The main content area remains similar to the non-logged-in version, featuring a 'Welcome to Mindiqo!' message, a user profile placeholder, and two large cards labeled 'Title' with small icons. Below these are two more 'Title' cards.

## Post details

Mindiqo

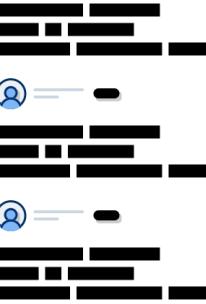
Home My notes Create with me Following

Note1 #tag1 #tag2 6 July 2023



15

Comments 3



## Note editing

The screenshot shows the Mindiqo interface for note editing. At the top, there's a navigation bar with 'Mindiqo' and a search bar labeled 'Search my notes'. Below the navigation is a header with the title 'Untitled'. To the right of the title are buttons for 'Enter or select a tag' and 'Create'. On the left, a sidebar menu includes 'Home', 'My notes' (which is selected and highlighted in grey), and 'Create with me'. Under 'My notes', there's a section titled 'Bookshelf' containing several placeholder note cards. The main area is currently empty, representing the 'Untitled' note.

## User profile

The screenshot shows the Mindiqo user profile page for 'John Doe'. At the top, there's a navigation bar with 'Mindiqo' and a search bar labeled 'Search in site'. Below the navigation is a header with the user's name 'John Doe' and an 'Edit' button. To the right of the name is a user icon. On the left, a sidebar menu includes 'Home', 'My notes' (selected and highlighted in grey), and 'Create with me'. Under 'My notes', there's a section titled 'Following' containing several placeholder note cards. The main area features a large heading 'Publish notes' and two sections of note cards. Each card has a title ('Title'), a preview image placeholder (a black square with a small icon), and a three-dot menu icon. The first section has one card, and the second section has two cards.

# Appendix D User Evaluation Questionnaire

2023/8/24 05:06

Mindiqo User Evaluation Questionnaire

## Mindiqo User Evaluation Questionnaire

This survey is designed to gather your feedback and on Mindiqo. Your responses will play a crucial role in development of future applications. **All responses will be kept highly confidential and will not be shared with third parties without your explicit consent.** We greatly appreciate your time and valuable input.

\* Required

1. What note-taking app have you used? \*

- Evernote
- Onenote
- Google Keep
- Notion
- Typora
- Never used note-taking app before
- Other

2. Compared to other note-taking apps you've used, how would you rate Mindiqo? \*

- Better
- Excellent
- Average
- Worse
- Equal

3. How do you rate the note-taking functionality in Mindiqo? \*

- Excellent
- Good
- Average
- Poor
- Very good

4. How do you rate the knowledge community feature in Mindiqo? \*

- Excellent
- Good
- Average
- Poor
- Very poor

5. How do you rate the user profile and settings in Mindiqo? \*

- Excellent
- Good
- Average
- Poor
- Very poor

6. Would you sharing your note in the knowledge community? \*

- Yes
- No
- Maybe

7. How would you rate the user interface (UI) design of Mindiqo? \*

- Excellent
- Good
- Average
- Poor
- Very poor

8. How natural and intuitive did you find interacting with the Mindiqo app? \*

- Very natural
- Somewhat natural
- Neutral
- Somewhat unnatural
- Very unnatural

9. Have you encountered any problems or issues while using Mindiqo? \*

- Yes (Please specify in the next question)
- No

10. If you answered "Yes" to the previous question, please describe the problem or issue you faced.

11. Overall, how satisfied are you with Mindiqo? \*

- Very satisfied
- Satisfied
- Neutral
- Unsatisfied
- Very unsatisfied

---

This content is neither created nor endorsed by Microsoft. The data you submit will be sent to the form owner.

 Microsoft Forms

# Appendix E User Evaluation Questionnaire

2023/9/1 01:18

Mindiqo User Evaluation Questionnaire

## Mindiqo User Evaluation Questionnaire

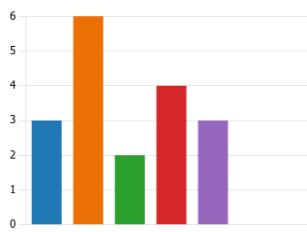
9  
Responses

01:37  
Average time to complete

Active  
Status

1. What note-taking app have you used?

Evernote	3
Onenote	6
Google Keep	2
Notion	4
Typora	3
Never used note-taking app bef...	0
Other	0



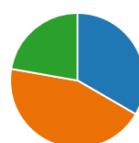
2. Compared to other note-taking apps you've used, how would you rate Mindiqo?

Better	5
Excellent	2
Average	2
Worse	0
Equal	0



3. How do you rate the note-taking functionality in Mindiqo?

Excellent	3
Good	4
Average	2
Poor	0
Very good	0



4. How do you rate the knowledge community feature in Mindiqo?

Excellent	6
Good	2
Average	1
Poor	0
Very poor	0



[https://forms.office.com/Pages/DesignPageV2.aspx?origin=NeoPortalPage&subpage=design&id=KVxybjp2UE-B8i4ITwEzyN0HYL\\_JjsVHomqwJIFg7mpUMF...](https://forms.office.com/Pages/DesignPageV2.aspx?origin=NeoPortalPage&subpage=design&id=KVxybjp2UE-B8i4ITwEzyN0HYL_JjsVHomqwJIFg7mpUMF...) 1/3

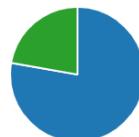
5. How do you rate the user profile and settings in Mindiqo?

<span style="color: blue;">●</span> Excellent	4
<span style="color: orange;">●</span> Good	3
<span style="color: green;">●</span> Average	2
<span style="color: red;">●</span> Poor	0
<span style="color: purple;">●</span> Very poor	0



6. Would you share your note in the knowledge community?

<span style="color: blue;">●</span> Yes	7
<span style="color: orange;">●</span> No	0
<span style="color: green;">●</span> Maybe	2



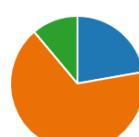
7. How would you rate the user interface (UI) design of Mindiqo?

<span style="color: blue;">●</span> Excellent	1
<span style="color: orange;">●</span> Good	7
<span style="color: green;">●</span> Average	1
<span style="color: red;">●</span> Poor	0
<span style="color: purple;">●</span> Very poor	0



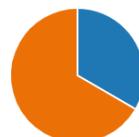
8. How natural and intuitive did you find interacting with the Mindiqo app?

<span style="color: blue;">●</span> Very natural	2
<span style="color: orange;">●</span> Somewhat natural	6
<span style="color: green;">●</span> Neutral	1
<span style="color: red;">●</span> Somewhat unnatural	0
<span style="color: purple;">●</span> Very unnatural	0



9. Have you encountered any problems or issues while using Mindiqo?

<span style="color: blue;">●</span> Yes (Please specify in the next q...)	3
<span style="color: orange;">●</span> No	6



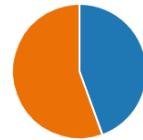
10. If you answered "Yes" to the previous question, please describe the problem or issue you faced.

1  
Responses

Latest Responses

11. Overall, how satisfied are you with Mindiqo?

Very satisfied	4
Satisfied	5
Neutral	0
Unsatisfied	0
Very unsatisfied	0



# Appendix F Ethics Checklist

**School of Computing Science  
University of Glasgow**

## **Ethics checklist form for 3<sup>rd</sup>/4<sup>th</sup>/5<sup>th</sup> year, and taught MSc projects**

This form is only applicable for projects that use other people ('participants') for the collection of information, typically in getting comments about a system or a system design, getting information about how a system could be used, or evaluating a working system.

**If no other people have been involved in the collection of information, then you do not need to complete this form.**

If your evaluation does not comply with any one or more of the points below, please contact the Chair of the School of Computing Science Ethics Committee ([matthew.chalmers@glasgow.ac.uk](mailto:matthew.chalmers@glasgow.ac.uk)) for advice.

If your evaluation does comply with all the points below, please sign this form and submit it with your project.

---

1. Participants were not exposed to any risks greater than those encountered in their normal working life.

*Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that occur outside usual laboratory areas, or that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, that use sensory deprivation (e.g. ear plugs or blindfolds), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback*

2. The experimental materials were paper-based, or comprised software running on standard hardware.

*Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, laptops, iPads, mobile phones and common hand-held devices is considered non-standard.*

3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project.

*If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, or the data is to be*

*published), then signed consent is necessary. A separate consent form should be signed by each participant.*

*Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script.*

4. No incentives were offered to the participants.

*The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.*

5. No information about the evaluation or materials was intentionally withheld from the participants.

*Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.*

6. No participant was under the age of 16.

*Parental consent is required for participants under the age of 16.*

7. No participant has an impairment that may limit their understanding or communication.

*Additional consent is required for participants with impairments.*

8. Neither I nor my supervisor is in a position of authority or influence over any of the participants.

*A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.*

9. All participants were informed that they could withdraw at any time.

*All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script.*

10. All participants have been informed of my contact details.

*All participants must be able to contact the investigator after the investigation. They should be given the details of both the student and module co-ordinator or supervisor as part of the debriefing.*

11. The evaluation was discussed with all the participants at the end of the session, and all participants had the opportunity to ask questions.

*The student must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation. In cases where remote participants may withdraw from the experiment early and it is not possible to debrief them, the fact that doing so will result in their not being debriefed should be mentioned in the introductory text.*

12. All the data collected from the participants is stored in an anonymous form.

*All participant data (hard-copy and soft-copy) should be stored securely, and in anonymous form.*

---

**Project title** The Development of Mindiqo, a Note-Taking Application

**Student's Name** \_\_\_\_\_ Jiakun Ge\_\_\_\_\_

**Student Number** \_\_\_\_\_ 2653432\_\_\_\_\_

**Student's Signature** \_\_\_\_\_ 

**Supervisor's Signature** \_\_\_\_\_

**Date** \_\_\_\_\_ 30 August 2023\_\_\_\_\_

Ethics checklist for projects