Management Tools Web Application

Nathakhun Passornratchakul
Natchakorn Patanakhajorn
Hataicharnok Woravutrangkul

Bachelor of Engineering in
Financial Engineering
School of Engineering
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2022

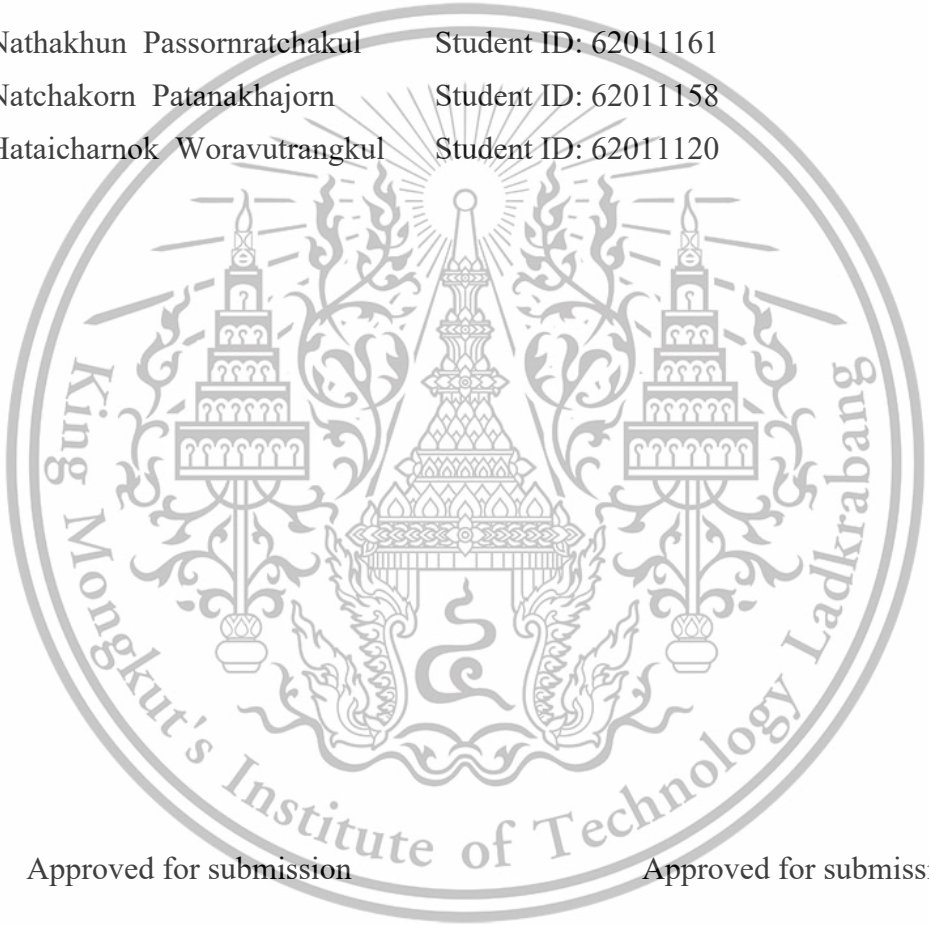
Thesis – Academic Year 2022

Bachelor of Engineering in Financial Engineering
School of Engineering
King Mongkut's Institute of Technology Ladkrabang

Title: Management Tools Web Application

Authors

1. Nathakhun Passornratchakul Student ID: 62011161
2. Natchakorn Patanakhajorn Student ID: 62011158
3. Hataicharnok Woravutrangkul Student ID: 62011120



Approved for submission

.....

(Asst. Prof. Dr. Isara Anantavrasilp)
(KMITL) Advisor

Date 3/6/23

Approved for submission

.....

(Assoc. Prof. Dr. Saran Sarntisart)
(NIDA) Advisor

Date 2/6/23

Acknowledgement

I would like to take this opportunity to express my sincere gratitude to all those who have contributed to the completion of this thesis. Their support and assistance have been invaluable in shaping my ideas and providing me with the resources necessary to produce a scholarly work that meets the standards of academic excellence. First and foremost, I would like to express my deepest appreciation to my thesis advisor, Dr. [name] and Dr. [name], for his guidances, supports, and constants encouragement throughout the entire process. This thesis would not have been possible without his expertise and knowledge in the field. Furthermore, I would like to thank the faculty members at the Financial Engineering Department for creating a challenging yet stimulating learning environment that enabled me to pursue this project. I must also express deep appreciation for the unfaltering encouragement and love offered by my family and friends throughout this process, as it helped me navigate through numerous challenges when researching, programming, and writing a thesis.

Abstract

The aim of this financial engineering project is to develop effective solutions for enhancing the workflow of a companyAAA. To overcome this problem, we must first understand the working practices of the employees in this organization. After discussing with the staff of this company and our advisors, we eventually developed a web application for a management tool that is appropriate for this particular company. We anticipate that this management tool web application will benefit not only the employees of this business but also those who are looking for management solutions to organize their duties, tasks, and projects.

A web application is a computer program that runs on a server and can be accessed by users through web browsers. It provides a web-based user interface that allows users to interact with application servers and deploy the core software components for various business processes. Three components make up a web application: the front-end, the back-end, and the database. The front-end is responsible for presenting the user interface and handling user interactions. The back-end of a web application refers to the part of the application that runs on the web server and is responsible for data processing and storage. It typically communicates with the front-end through an API (Application Programming Interface) and is built using server-side programming languages, while the database is where the application stores and retrieves data.

We proudly propose WEBAAA, a web application for project management, in this project with the hope that it would enhance the workflow of the company.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Problem description and Motivation

We discovered from discussions and research with employees in a company that they want us to create a solution to help them improve their workflow. Employees may struggle with inefficiencies and disarray in their workflow at times. Uncertainty and delays may result from the random way in which tasks are distributed, with inadequate coordination or accountability. Important communications may be lost in the shuffle due to fragmented and irregular staff communication. Furthermore, the absence of a centralized platform for task management and document sharing could lead to errors and duplication of effort.

The development of management tools for an organization is driven by the motivation to enhance efficiency, effectiveness, and productivity in daily operations. These tools are designed to streamline processes, reduce errors, and minimize costs. Moreover, effective management tools are necessary to retain employees and attract new talents by providing opportunities for career growth within the organization. Management tools also play a crucial role in monitoring and measuring the performance of individuals and teams within an organization, helping to identify areas that require improvement, as well as areas of strength that can be harnessed to drive success. Therefore, our aim is to develop a web application that is user-friendly and based on a business workflow. The resulting application increases productivity and organizes collaboration.

1.2 Objectives

The objectives of developing a management tools web application are providing a centralized platform for managing a company's operations, resources, and employees. By utilizing a management tools web application, a company can improve communication, streamline processes, and increase efficiency. To improve employee workflow, decrease errors, and boost productivity, the application can provide task assignment and tracking, document sharing, and real-time chat.

1. The development of a web application for managing company workflows can assist the business prevent errors and oversights by giving tasks and steps a clear and structured framework.

2. The business can conserve time and costs by managing its workflows using a web application instead of using manual, prone-to-error methods. Many workflow-related tasks can be automated by a web application, which can help the company cut down on the effort and time required to complete a process.

3. To enhance coordination and dialogue Workflow management among employees helps enhance teamwork and communication inside the organization. A web application can help team members keep informed and up to date as well as facilitate collaboration and efficient teamwork by offering a central platform for sharing and tracking tasks and stages.

4. The application can facilitate communication among employees, track tasks, and provide valuable data analytics that can inform decision-making.

5. Utilizing a management tools web application is essential for remaining competitive and flexible in the dynamic business environment of today. We anticipate that by utilizing the management tools online application, the business will be able to improve efficiency, maximize its resources, and streamline its operations, keeping it one step ahead of the competition.

1.3 Scope of Work

The scope of this project can be listed as follows:

1. Understand the Information Architecture (IA) of the organization.
2. To develop wireframes for the management tools web application of the company.
3. To develop UX and UI design.
4. To create visual design of the management tools web application.
5. To develop front-end of the web application
6. To develop back-end and database of the web application

1.4 Thesis Structure

This thesis consists of five chapters which are arranged as follows:

- Chapter 1 Introduction - refers to the motivation, objectives, scope of work, and thesis structure of this thesis.
- Chapter 2 Literature Review – proposes the Literature survey, explains the knowledge and technology necessary for developing the web application.
- Chapter 3 Methodology – requirements analysis and system design, presents the requirement of the system, the use case diagram, and the relevant system architecture diagram.
- Chapter 4 System Development – explains the concepts, tools, and techniques that are used in developing the project.
- Chapter 5 Results – refers to the results of a software demonstration, which consists of the user interfaces of the software and the web applications.

- Chapter 6 Conclusion and Future work – is the chapter that talks about the conclusion, future work, and improvements to the project.

Chapter 2
Literature Review

This chapter provides an overview of the Preliminary due diligence process of the companyAAA for making the decision to invest in a startup company and presents the importance of management tools for the workflow of an organization and explains the knowledge and technology necessary for developing the web application.

2.1 Preliminary due diligence of CompanyAAA

CompanyAAA focuses on funding early-stage startups and new technology businesses with the potential to upend the finance industry. Preliminary due diligence is one of the crucial processes in the investment process and entails assessing a potential investment opportunity to determine whether it aligns with requirements and objectives of the company. This step has significance because it aids in identifying potential risks and opportunities and enables the company to decide whether to move forward with further in-depth due diligence.

Preliminary due diligence is an initial assessment or review of a potential investment opportunity. It is typically done before more detailed due diligence is conducted, and its purpose is to determine whether the opportunity warrants further investigation. In the context of companyAAA, this step would involve looking at the company's business model, financial performance, market position, and competitive landscape, among other factors.

The following criterias are among evaluation standards of companyAAA for preliminary due diligence procedure of a startup.

Company and Team Structure

To evaluate the company and team structure of a startup, investors review the company's organizational chart and job descriptions to understand the hierarchy and responsibilities of each position.

Examine the management and leadership qualifications and experienceof the team, paying particular attention to their track record of establishing and leading prosperous enterprises.

In addition, investors could try to grasp the company's culture and beliefs, which might reveal information about the dynamics of the team and how well they might get along.

To gain an understanding of their working methods and problem-solving strategies, this may entail speaking with the management group and other staff members.

Investors also evaluate the company's governance structure, including its board of directors and any committees that have been established. This can provide insight into the company's decision-making processes and the roles and responsibilities of each individual within the organization.

Overall, evaluating the company and team structure is an important part of the due diligence process, as it can provide insight into the company's leadership and decision-making processes, and help investors determine if the team is well-positioned to achieve the company's goals.

Market and Sector Outlook

To evaluate the market and sector outlook of a startup company, investors conduct market research to understand the size and growth potential of the market, as well as the competitive landscape.This can reveal information about the company's potential issues and possibilities as well as the demand for its goods or services.

Investors also analyze industry reports, research, customer surveys, and analyses in addition to conducting market research to gain a general understanding of the outlook for the industry in which the company operates.This can provide insight into the broader economic and regulatory factors that may impact the company's performance, as well as trends and developments in the industry.

Investors conduct market research to comprehend the size and growth potential of the market, as well as the competitive landscape, in order to assess the market and sector prospects of a startup company. This can provide insight into the potential demand for the company's product or service, as well as the factors that may influence its success.

Business Model & Products Solutions

Investors examine the company's products and market fit to define the competitive advantage, the products' demand in the market regardless of economic conditions, and the uniqueness and not easily replaceable by other products and services.

The product pipeline and new product development are aspects of a company's growth strategy and business model and product solution evaluation, as they can help the company stay competitive and generate additional revenue. The product pipeline typically includes a list of products or product enhancements that the company is currently developing or plans to develop in the future. This may include new products, as well as updates or improvements to existing products.

Technology and Intelligence Property

To evaluate the technology and intellectual property of a startup company, investors review the company's technology, operational processes, workflow, trademarks, and other intellectual property assets. This can provide insight into the company's ability to protect its innovations, scalability, security management and maintain its competitive advantage.

To evaluate the technology, investors review the company's technology stack, including the platforms, frameworks, intellectual property and tools it uses to develop and deliver its products or services. They may also review the company's technical capabilities, including its engineering team, development processes, and testing procedures.

Tractions and KPI

To evaluate the traction and key performance indicators (KPIs) of a startup company, investors review the company's traction and KPI, including quality of traction, sales pipeline and sales metrics, Net Promoter Score (NPS) and other performance metrics. This can provide insight into the company's growth, profitability, and overall performance.

The NPS is a widely used metric in customer satisfaction research, as it provides a quick and easy way to gauge the level of customer satisfaction and loyalty, and help identify areas for improvement.

Financial and Projections

To evaluate the financial projections of a startup company, investors review the company's financial statements, including historical financial performance. They also review the company's business plan, use of funds, cash conversion cycle and financial projections, which outline the company's expected revenue, expenses, profits, company's liquidity and ability to generate cash from its operations over a given period of time.

Data Collection & Analytics

In the evaluation of a startup company, investors review the company's data collection and analytics capabilities. This can provide insight into the company's ability to generate and use data to drive decision-making and improve its operations.

To evaluate the company's data collection and analytics capabilities, investors review the company's data sources, including its internal and external data sources, as well as the quality, integrity and reliability of its data.

They may also assess the company's data management and storage systems, including its data warehousing, data lakes, and data marts.

In addition, investors may review the company's analytics tools, techniques, and processes, including its data visualization, predictive modeling, machine learning capabilities and plan of implementation of the compliance policy such as PDPA. This can provide insight into the company's ability to generate insights and actionable recommendations from its data.

Synergy with Bank (CompanyAAA)

The synergy between a startup company and a bank can refer to the benefits and value that the company and the bank can generate by working together. This includes the competency and collaboration of the company with the bank, as well as the interest and capability to work together.

Term of Investment

To evaluate the terms of investment for a startup company when raising funds with a bank, investors review the term sheet, which refers to the investment policy and justifiable valuation of the company. They may also assess the conditions and milestones that the company must meet in order to receive the funding, as well as any rights or privileges that the bank may have as an investor.

In addition, investors seek to understand the potential benefits and drawbacks of the terms of investment for the company, including the potential impact on the company's equity ownership, control, and future financing options.

Company Governance

Investors also review the company governance, including shareholding structure and affiliate and Employee Stock Ownership Plan or ESOP which can be a useful tool for engaging leadership and management in a startup company. By giving employees a financial stake in the company through stock ownership, an ESOP can help align the interests of management and employees, and can motivate employees to work toward the success of the company.

Additionally, by providing a way for employees to share in the company's success, an ESOP can help foster a sense of ownership and accountability among leadership and management. Overall, an ESOP can be a powerful tool for building a strong, engaged, and motivated leadership team in a startup company.

In order to evaluate how well a startup company is being governed, it is also useful to look at the company's board of directors and company advisors that can provide valuable guidance and oversight to the leadership team, and can help ensure that the company is operating in the best interests of all stakeholders.

2.2 Importance of management tools

Management tools are essential for the workflow of an organization as they facilitate effective communication, coordination, and organization of tasks among employees. By providing a centralized platform for task management, document sharing, and messaging, management tools help employees stay focused, productive, and engaged. By ensuring that activities are assigned, recorded, and done effectively, these technologies can also streamline procedures and lower error rates. Organizations can identify areas for improvement and make data-driven decisions through the important data and analytics provided by management tools. Ultimately, management tools enhance the workflow of an organization, improving efficiency, productivity, and overall performance. Without management tools, organizations may struggle with disorganization, lack of accountability, and reduced productivity, leading to lower quality work output, higher costs, and a less competitive position in the market.

2.3 Information Architecture (IA)

Information architecture (IA) refers to the process of structuring, organizing, and labeling content in a way that makes it simple for users to find and use. It is a crucial component of web development and design. It is the art and science of creating a structure for information that allows user to navigate and interact with it effectively.

As it provides the underlying framework for how information is organized and presented, IA is frequently regarded as the foundation of a website or application. Good

IA can improve the user experience, enhance accessibility, and make it easier for users to achieve their goals.

There are several key components of IA that we considered:

Organization: The organization of content refers to how it is grouped and structured. Content should be organized in a way that makes sense to users and reflects their needs and interests. This includes hierarchical structures and Chronological order

Hierarchies are a tree-like structure, content is arranged into categories and subcategories, with the broader categories at the top and more specialized categories branching out beneath them. In graphic design, hierarchies or hierarchical structures are an excellent method to show how the components of the task interact with one another. It is a system of classifying things into several categories and subdivisions, and it is frequently used to arrange content on websites.

Size, color, contrast, brightness, alignment, and repetition are some of the fundamental concepts of hierarchy in graphic design that are used to highlight particular elements of the design. It takes into account such elements to demonstrate their significance to the design as a whole.

The content is organized logically and clearly in hierarchical frameworks, which makes it simpler for users to navigate and locate what they're looking for. This can be very helpful for applications or websites with a lot of complex information or material.

Chronological order is a way of organizing and presenting content in information architecture (IA) that is based on time or date. The content is arranged in a linear form in chronological order, with the earliest or oldest content appearing at the top and the most recent or most current stuff appearing at the bottom.

For time-sensitive content, such as new tasks, notes, or current discussions and comments, chronological order is employed. It enables users to simply follow the progression or evolution of content over time and to be informed of new information.

One benefit of chronological order in IA is that it offers a simple, logical structure for content organization that is simple for users to comprehend and follow. For visitors looking for the most recent information, it also makes it simple to see what content is new or just published.

Navigation: Navigation refers to the system of links and menus that users utilize to move through a website or application. Clear labeling and logical groups should be present in the navigation, making it intuitive and simple to utilize.

Labeling: Labeling describes the words and phrases used to define the navigation and content elements. Labels should be understandable to the target audience and include language that is clear, succinct, and consistent.

IA is an iterative process that includes testing, research, and analysis to make sure that the content is organized and structured in a useful and user-friendly way. To construct a unified and efficient information architecture that satisfies expectations of users, collaboration is needed between designers, developers, content producers, and other stakeholders.

2.4 Wireframes

A wireframe is a graphic depiction of a web page or application interface that demonstrates the organization and layout of the material without the inclusion of design components like colors, typography, or images. Before a digital product is built, a low-fidelity prototype is used to plan and communicate the design and functionality of the product.

Wireframes can be made with a variety of techniques, including paper sketches, digital wireframing software, or even basic drawing applications. They are a crucial component of the design process. A wireframe frequently includes the basic parts of a web page or application, such as headers, footers, menus, content areas, and user interface elements like buttons, forms, and input fields.

A wireframe is used to illustrate the design and functioning of a digital product in an easy-to-understand manner. A wireframe enables designers and stakeholders to concentrate on the structural and functional features of the product, such as the placement of content, the flow of navigation, and the hierarchy of information, by omitting design elements like colors and typography. In order to test and improve their design ideas, recognize and fix usability issues, and effectively convey their concepts to stakeholders and developers.

2.5 User Experience (UX)

UX design focuses on the total interaction that users have with a digital product. It includes every stage of the user experience; from the first time a consumer learns about a product to their last engagement with it. The objective of UX design is to produce a good and fulfilling user experience that satisfies the demands and objectives of users.

Several techniques, including user research, persona development, user journey mapping, and usability testing, are used by UX designers to comprehend the needs and objectives of users. They make use of this data to create user-friendly, intuitive interfaces that are simple to use and add value for users.

2.6 User Interface (UI)

UI design focuses on the visual appeal and interaction features of a digital product, such as buttons, forms, and menus. The aim of UI design is to produce an intuitive, visually appealing, and engaging user experience.

Visual design elements like color theory, typography, and layout are used by UI designers to provide a unified and aesthetically pleasing design that enhances every aspect of the user experience. Additionally, they employ interactive design components like animations and micro-interactions to give users a sense of feedback and delight.

2.7 Graphical user interfaces (GUIs)

A graphical user interface (GUI) is a type of user interface that enables consumers to interact with digital products using graphical components like menus, buttons, and icons. GUIs provide users a way to interact with a digital product using a combination of mouse, keyboard, and touch-based inputs, and they are designed to be intuitive and user-friendly.

GUIs are utilized in a wide range of digital products, including desktop and mobile applications, websites, and video games. They are made to simplify difficult processes by dividing them into smaller, easier-to-manage chunks and giving users immediate, understandable feedback on their actions.

2.8 Relationship Between UX and UI Design

Although UX and UI design are distinct disciplines, they are closely related and frequently collaborate. While UI designers create the visual and interactive components that support that experience, UX designers define the overall user experience.

In practice, UX and UI design are iterative processes in which designers collaborate to develop and improve the product. While UI designers create high-fidelity mockups and prototypes to perfect the visual and interactive components of the design, UX designers may use wireframes and prototypes to test and validate design concepts

2.9 Hypertext Markup Language (HTML)

HTML (Hypertext Markup Language) is a markup language for creating and structuring web content. It serves as the foundation of web development by offering a uniform set of rules and instructions for developing web pages. The layout, content, and structure of web pages are specified using HTML.

The structure and content of a web page are specified by a set of tags called tags in HTML. The headings, paragraphs, lists, tables, forms, photos, links, and other parts of a web page are all created using these tags. HTML tags can contain attributes that provide more details about the element, and they are written using angle brackets (>).

Some common HTML tags include:

<html>: Defines the start and end of an HTML document.
<head>: Contains metadata about the web page, such as the title and links to stylesheets.
<body>: Contains the main content of the web page.
<h1>, <h2>, <h3>, etc.: Defines headings of various sizes.
<p>: Defines a paragraph of text.
 and <ol>: Defines unordered and ordered lists, respectively.
<table>: Defines a table.
<form>: Defines a form that allows users to input data.
<img>: Defines an image.
<a>: Defines a hyperlink.

To build dynamic and interactive web sites, HTML is frequently combined with JavaScript and CSS (Cascading Style Sheets). JavaScript is used to bring dynamic behavior and interactivity to the page, whereas CSS is used to style the page's content and layout.

2.9.1 HTML5

HTML5 is the most recent version of HTML, the markup language used to produce and arrange web content. Since its initial release in 2014, it has evolved into the industry standard for web development. A variety of new features and enhancements brought by HTML5 increase the usability and efficiency of web pages.

Some of the key features of HTML5 include:

Improved multimedia support: New tags for audio and video content embedding make it simpler to add multimedia to web pages without the use of plugins.

Semantic markup: New semantic tags, such as <header>, <footer>, <article>, and <section>, that provide a better structure for web pages and make it simpler for search engines to interpret the content of the page.

Improved form controls: New form controls that simplify user data entry, such as date pickers, sliders, and search fields.

Improved offline support: New APIs that allow web pages to function offline, making it possible to create web applications that work even when the user is not connected to the internet.

Improved performance: New features that increase web page performance, such as lazy loading of images and the ability to prefetch resources.

Canvas and SVG support: Support for both Canvas and SVG, making it easier to create complex graphics and animations directly in the web browser.

HTML5 is a potent tool for web developers that offers several of essential features and benefits, making it simpler to design interesting and interactive web apps that function on various platforms and devices.

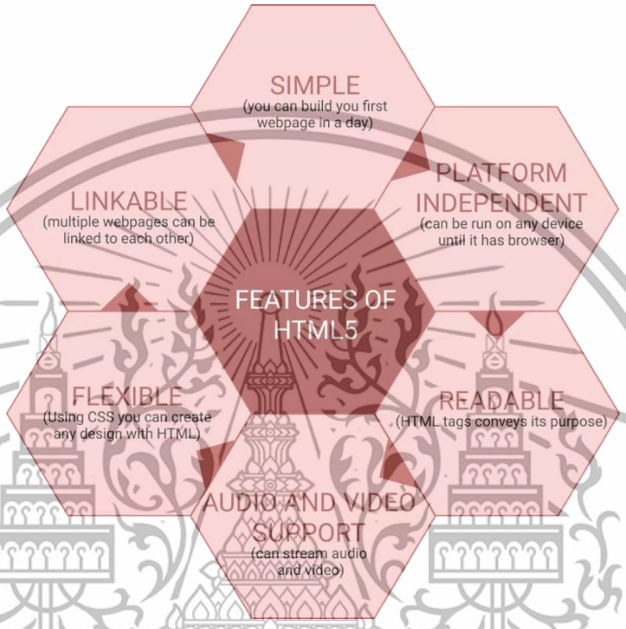


Figure 1. Features of HTML5

[source: https://www.tutorialstonight.com/html/html-introduction/]

2.10 Cascading Style Sheets (CSS)

CSS (Cascading Style Sheets) is a stylesheet language used to describe the display and layout of HTML (Hypertext Markup Language) and XML (Extensible Markup Language) documents. CSS is used to separate the content of a web page from its presentation and is used to style the visual look of web pages, including fonts, colors, layouts, and other graphical components. This separation enables developers to produce stylish and visually appealing web pages that are simple to maintain and update.

CSS operates by establishing guidelines for how HTML parts must be rendered. These rules are made up of selectors, which identify the HTML elements that need styling, and declarations, which outline the styling requirements for each element. CSS rules

can be defined either inline within the HTML document, in a separate stylesheet file, or in the head section of an HTML document.

In conclusion, CSS is a potent style language used to specify how HTML and XML documents are presented and laid out. It is simpler for developers to maintain and update web pages and makes them more accessible to users with diverse needs when they are given the option to separate the content and structure of a document from its display. Advanced layout techniques are supported by CSS as well, enabling the development of intricate and responsive web designs.

2.11 JavaScript

JavaScript is a high-level programming language used to bring dynamic behavior to websites and web applications. It is one of the three fundamental technologies used in the creation of websites, together with HTML and CSS.

Because JavaScript is a client-side scripting language, it executes within the user's web browser on their computer, rather than on a server. This makes it possible to employ JavaScript for a range of activities like form validation, developing interactive user interfaces, and changing web page content without necessitating a complete page reload.

One of the primary advantages of utilizing JavaScript in web development is its ability to deliver a rich user experience. JavaScript can make web pages more interactive and dynamically behave so that people are more likely to interact with them and come back to a site.

However, if JavaScript is not used properly, it can potentially pose security problems. JavaScript, for instance, can be used by hostile actors to steal user information, inject code into a web page, and carry out other negative deeds. To safeguard their JavaScript code and avoid security issues, developers should follow best practices. It is crucial that

developers adhere to recommended practices for protecting their JavaScript code, such as employing secure coding methods and keeping their code updated with security patches.

2.12 Application Programming Interface (API)

Application Programming Interface is commonly referred as the API. It is a collection of protocols, routines, and tools for developing software applications that specify how software components should interact with one another. In order to facilitate communication between various software programs or services, APIs are frequently employed in web development.

APIs can provide a wide range of functionality, such as:

Data retrieval and storage: To retrieve and store data from databases, file systems, and other sources.

Authentication and authorization: To authenticate users and authorize access to protected resources.

Integration with third-party services: To integrate with third-party services such as social media platforms, payment gateways, and messaging services.

Functionality as a service: To provide a specific functionality as a service, such as image recognition or natural language processing.

Communication between applications: To enable communication between several apps, such as a web application and a mobile app.

APIs are critical to the development of modern software applications because they enable developers to construct applications more quickly and easily by reusing existing functionality and developing on top of existing systems and platforms. Additionally, they offer a standardized way to conduct interaction and integration between various systems and applications, which is essential for creating scalable and reliable programs.

2.13 Firebase

The Firebase Realtime Database is a cloud-hosted database. Data is synchronized in real-time to every connected client and stored as JSON. All of your clients share a single Realtime Database instance and are automatically updated with the most recent data when you create cross-platform apps using our Apple platforms, Android, and JavaScript SDKs.

A platform for building, maintaining, and deploying applications, Firebase offers a variety of tools and services. It was established in 2011 and was subsequently acquired by Google in 2014. Firebase offers a range of services that help developers from worrying about infrastructure, server maintenance, or backend development, allowing them to create applications more quickly and simply.

Firebase provides a wide range of services, including:

Real-time Database: A real-time database that can be used to sync and store data across clients and servers in real-time. This makes it simple to create tools like chat applications and collaborative tools that require real-time updates.

Authentication: Authentication services that make it simple for developers to add user authentication into their programs. This includes support for social login, multi-factor authentication, and email and password authentication.

Cloud Messaging: Cloud messaging services that enable developers send notifications and messages to users across many platforms, such as Android, iOS, and the web.

Cloud Storage: Cloud storage services that enable developers to store and serve user-generated content, such as images and videos.

Hosting: Hosting services that enable developers to deploy and host their web applications on Firebase. This includes support for custom domains and SSL certificates.

Functions: A serverless functions platform that enables developers to run backend code in response to events triggered by Firebase services or HTTP requests.

Analytics: Analytics services that enable developers to track user behavior and usage of their applications.

Test Lab: A test lab that enables developers to test their applications on real devices, across different versions of operating systems and screen sizes.

Firebase offers a variety of tools and services that help developers create applications more quickly and easily. It is also designed to be simple to use. Developers use it to create a variety of applications, from simple mobile apps to large-scale web applications.

Firebase provides a range of APIs that can be used for different purposes, including:

- Authentication API
- Real-time Database API
- Cloud Storage API
- Hosting API
- Cloud Messaging API
- Functions API
- Remote Config API

These APIs are made available by Firebase as a set of tools that can be used to create and run contemporary apps. As a result, Firebase isn't just one kind of API; rather, it is a group of APIs and services that can be combined to create and run mobile and web applications.

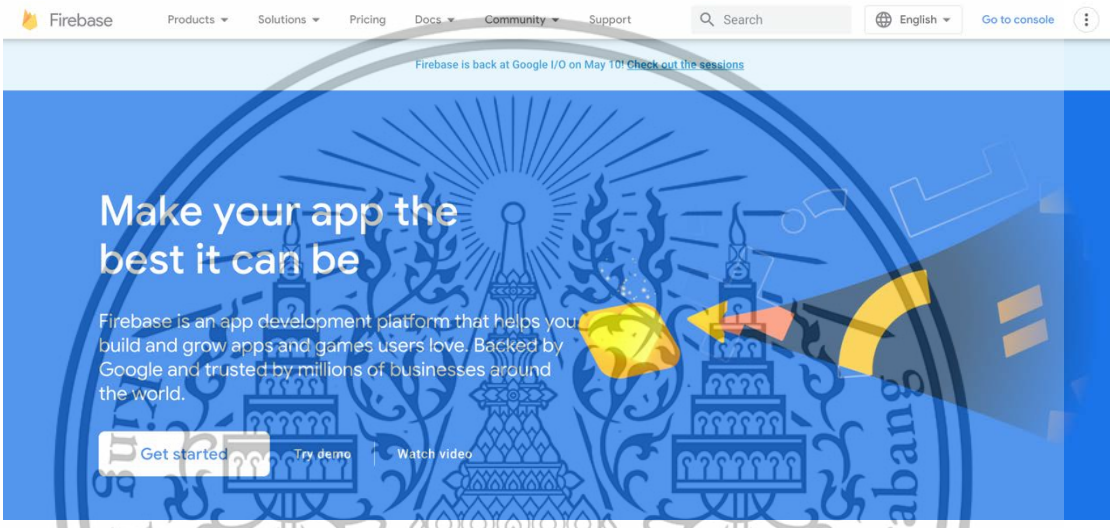


Figure 2. Firebase

Chapter 3

Methodology

In this chapter, explains the concepts, tools, and techniques that are used in developing the project.

3.1 Requirements

3.1.1 Functional Requirements

- The system must provide a way to authorize employees that want to use our system.
- The system must provide a way for employees to connect with projects and tasks through the management tools web application.
- The system must provide a way for employees to see the information of the projects, tasks, discussions and notes that are created and published on the web application.
- The system must provide a way for employees to see the comments that are written or created by the teammates.
- The system must provide a way for employees to see the list of tasks that they have been assigned
- The system must provide a way for employees to see the assignee of all tasks in the project.
- The system must provide a way for employees to publish projects, tasks, notes and discussions.
- The system must provide a way for employees to upload files for each task in each project.
- The system must provide a way for employees to manage the status of the tasks.

- The system must provide a way for employees to filter projects that are in active and completed status.
- The system must provide a way for employees to filter tasks that are in planning, pending, in progress, review and completed status.
- The system must provide a way for employees to filter tasks that have been assigned in My Work page.

3.1.2 Non-Functional Requirements

- The system should not disclose any data to the unintended users.
- The system should be simple enough that the users will be able to understand easily.
- The system should be simple enough that the user will be able to create and publish a project, task, discussion and note in a few steps.
- The backend application should be implemented with Firebase framework.
- The system must save all the information and File uploaded in the database.
- The system must be user-friendly.

3.2 System Architecture

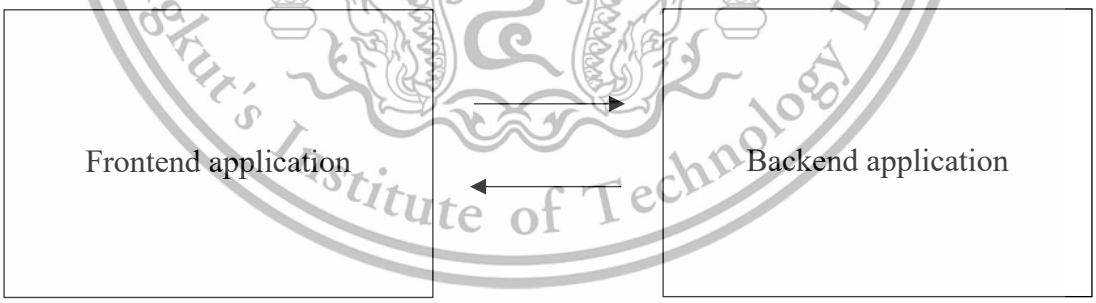


Figure 3. System Architecture Diagram

3.2.1 Frontend Component

The user-side of the project management tool has been implemented as a responsive website application. Therefore, the website application can be opened by both laptops and PCs as it is fully responsive. A user authentication system has been implemented

to ensure secure access and authorization. Authentication is required before users may access an application's features. Within the application, users can create projects, tasks, discussions, notes, and comments after gaining authorization. A seamless and uninterrupted user experience is provided by automatically saving all user interactions within the application. Projects, tasks, discussions, notes, and comments, as well as any other project-related entities, are created and modified by sending backend requests. Server-side processing of these requests leads to a database being securely stored with the information. The user will be required to authorize the request, the backend database is updated automatically every time a user updates or adds new information to published projects, tasks, discussions, notes, or comments. All users accessing the application will have access to the latest information. Users can work together effectively and keep up with project developments through this synchronization mechanism. It is important to highlight that for any changes or new actions performed within the application, user authorization is required. By doing so, the integrity and security of the data can be maintained, and unauthorized access and modifications can be prevented.

3.2.1.1 User Interface Design

In order for any application to succeed, user interface design is essential. For our project management tool, careful consideration was given to designing a user interface that optimizes usability and productivity. This involved a systematic approach, including wireframing, prototyping, and iterative improvements. By incorporating user-centered design principles, creating an intuitive interface with clear information hierarchy, seamless navigation, and visually appealing layouts to enhance the overall user experience.

3.2.1.2 Interactivity with User Experience

Interactivity is key to engaging users and improving their experience with our project management tool. Leveraging JavaScript and modern web technologies, this web application implemented various interactive elements and features to facilitate task management and collaboration. For instance, the users could generate as much as startups, projects, tasks, discussions, and notes. These include drag-and-drop functionality for effortless task organization and real-time updates to enable teams to

work synchronously. By emphasizing responsiveness and providing immediate feedback, we aimed to enhance user satisfaction and efficiency.

3.2.1.3 Accessibility

Accessibility Considerations in a Project Management Tool. Ensuring accessibility in our project management tool was a fundamental aspect of its development. To address visual accessibility, we employed several techniques. Ensuring appropriate color contrast ratios between foreground and background elements to make content readable for individuals with low vision. Additionally, avoided relying solely on color to convey important information, utilizing clear labels, icons, and other visual cues. By incorporating accessibility best practices, we aimed to create an inclusive environment that allows all users to effectively utilize our tool.

3.2.1.4 Integration with Back-End

To store data in the prototype project management tool, Firebase was used as the primary storage solution. A comprehensive suite of services is available through Firebase, including integration with real-time databases and file storage. Therefore, it is a proper choice for storing images, text, and files. The Firebase Realtime Database has been employed to store and retrieve dynamic data within the prototype project management tool. This NoSQL database allows for real-time synchronization. The front-end application connects to the Firebase Realtime Database, enabling seamless interaction with and modification of data in real-time. The application gains the ability to sync and manipulate data in real-time, enhancing user experience and making it dynamic.

A prototype project management tool has been developed utilizing HTML, CSS, and JavaScript, enabling seamless integration with Firebase. A set of libraries and APIs provided by the Firebase JavaScript SDK enable front-end applications to secure interactions with Firebase services. With Firebase SDK methods and API calls, the prototype tool can create, read, update, and delete data stored in Firebase, ensuring reliable and consistent data management.

3.2.2 Backend Component

A web application's server-side is referred to as the backend component. This includes the code, databases, and servers in charge of handling data management, processing user requests, and producing responses that are then transmitted back to the web browser. The backend component is comprised of five components and is listed below:

Server: The server is responsible for processing incoming requests from the client-side of the application and sending responses back. It executes the backend code that manipulates data, makes use of databases, and produces responses that are transmitted back to the user.

Database: Data for the application is stored and managed in the database. User data, application data, and any other data needed by the application are all stored there. The database is managed by a database management system (DBMS), which provides tools for generating, updating, and querying data.

Backend programming languages and frameworks: Backend programming languages, JavaScript is used to write the code that runs on the server-side of the application.

APIs: APIs (Application Programming Interfaces) are used to facilitate communication among various application components. They enable the integration of services and applications from third parties as well as communication and data sharing between the frontend and backend.

Authentication and security: The backend component of a web application is responsible for handling user authentication and ensuring the security of user data. This entails controlling user access, encrypting private information, and putting security controls in place to guard against unauthorized access to the application.

3.3 Use Case Diagram

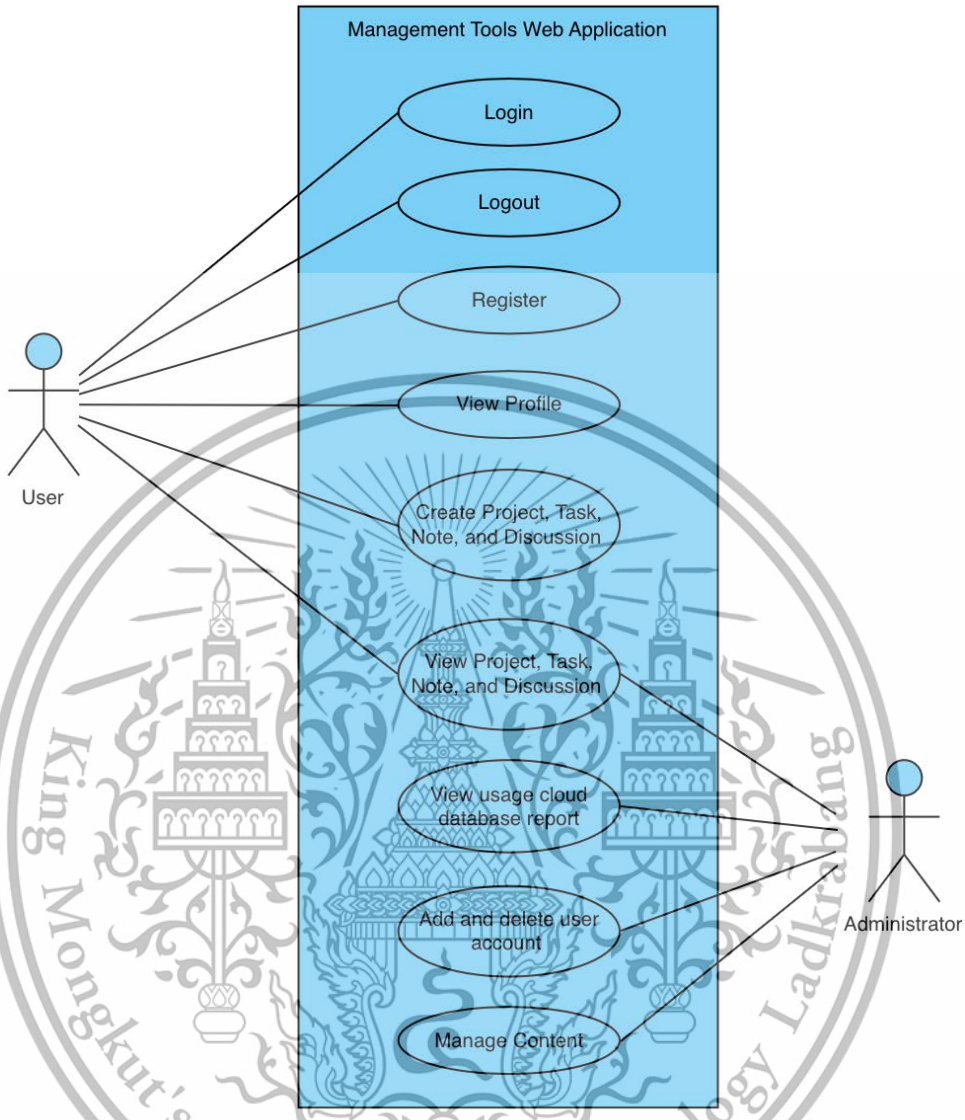


Figure 4. Use Case Diagram

3.4 Database Schema

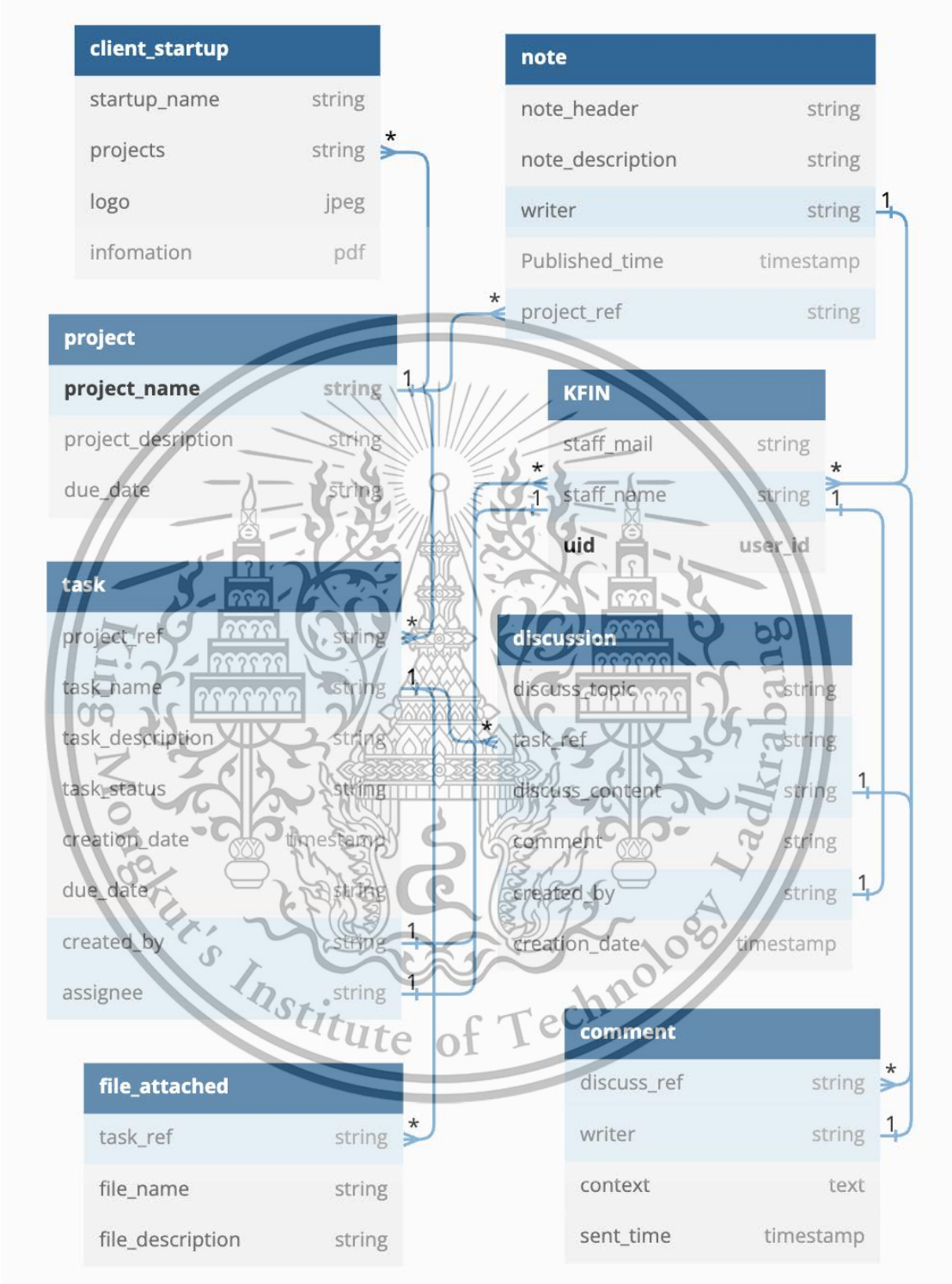


Figure 5. Database Schema

Chapter 4

System Development

4.1 HomepageData.js

This file serves as the central hub of our program, as it is the first page users encounter upon logging in. Here, users will find a list of existing projects and a button to create new ones. Additionally, the page includes essential elements such as a logo, functionality to retrieve documents from Firebase, and the ability to retain user input when creating new projects.

Table 1. List of functions in HomepageData.js

Function	Description
onAuthStateChanged(auth: userInformation, (user) => {..})	Retrieving user data and creating elements to document.
createUserStorage(user: data)	Creating an array “myWork” in the Firestore Database for new users.
logout_yes();	Toggling the page to be blurred via CSS
gettingDropDown();	Get the data from startup lists which are created by the users in Firestore Database.
showDropDown(value : string);	Creating drop down in document
keepProjectData();	Retrieving project data from users.

keepProjectNames(projectName : string);	Add project name to the Firestore Database.
gettingExistingProject();	Get the name from Firestore Database.
createContainer(projectName: string);	Create containers on the web for users.
gettingLogo(startup: string, projectContainer: element);	Get the path of the logo to refer to the start up being chosen.
gettingDocIDofStartUp(id: string , startUp: string, projectContainer: element)	Get the document path from Firestore Database to get the logo image.
gettingActualLogo(logoFileName: string, startUp: string, projectContainer: element)	Creating logo images for each project.

4.2 TaskData.js

In this file, it is the first page after users clicked into each project, which shows tasks of that project. Moreover, users are able to create new tasks and attach user's files into each task. Additionally, there are five fixed columns which are Planning, Pending, In progress, Review and Completed. After the task has been created, it will automatically append to the Planning column. For each task, they can be dragged and dropped in any column to change the status of the task. Furthermore, users are able to click on each task, and there will be a popup that allows them to see the description, files reference, due date, task created by.

Table 2. List of functions in TaskData.js

Function	Description
callHeader();	Get the header of the page.
deleteHeaderTask();	Erase the header.
closeTask();	Hide the task when the user clicks on the button.

uploadFileInTask();	Update file description and actual file to Firestore and Storage.
createTask();	Retrieve task data from the user input and update to the Firestore Database.
getting(projectName: string)	Getting the existing tasks and adding it into each column, also allows drag and drop between columns.
getTaskDesc(value: string, dataid: string);	Getting task description and adding elements to the document.
getAsigneeName(id: string, valueClicked: string);	Get the assignee data according to the task being clicked.
getAsigneeNameField(id: string, valueClicked: string);	Get the field in Firestore Database, in order to receive the data
getAsigneeNameField2(id: string, valueClicked: string);	Append the elements which receive from the Firestore Database into the document.
getFileDesc(item: string, value: string);	Getting file description from the Firestore and appending it into the document.
getFileName(fileNames: string, value: string);	Getting file names from the database and appending them into the document.

4.3 DiscussionData.js

This file represents the discussion page which is after clicked on the projects. There is a button that allows users to create discussions in which users can choose which task they want to refer to. After they create discussions, there will be containers for users to click on it. When they click on the container, there will be a popup that shows the

information inside, including Discussion's header, description and discussion created by. Moreover, users are able to comment on that discussion.

Table 3. List of functions in DiscussionData.js

Function	Description
callHeader();	Get the header of the page.
deleteHeaderDiscussion();	Erase the header.
gettingDropDown();	Get the data from startup lists which are created by the users in Firestore Database.
showDropDown(value : string);	Creating drop down in document
closeProject();	Hide the task when the user clicks on the button.
createDiscussion();	Retrieving the data of discussion from users input.
createDiscussionConatiner(item: string);	Create a container on the document, allowing the user to see the actual container on the web.
discussionLists.forEach((container) => {..})	Getting the path of the comment in each discussion.
createDescContainer(getIdContainer: string)	Retrieving the comments section data from Firestore and appending to the document.
createCommentsContainer(orderComment: string, discussionContainer: string);	Create containers for the data that are received.

4.4 Files.js

For File.js file, this page shows all the files that refer to every task created. Also the files from startup which we have already created earlier. There will be no button that allows users to add a new file into a non-reference task. Moreover, users are able to download any files from this page.

Table 4. List of functions in Files.js

Function	Description
callInfo();	Get the client name and pass on the value.
gettingFileStartUpInfo(startUp: string);	Get the path of each document in Firestore.
gettingInfoFileName(docID: string, startUp: string)	Get the data from Firestore which is the name of the file.
gettingFileInfoFileStorage(fileInfoName: string, startUp: string);	Get the actual file data from Firebase Storage. Also, append the elements to the document
gettingInfoFileIcon(fileInfoName: string, eachDesFileElement: element);	Get the file name and consider the type of the file and append the icon.
callHeader();	Get the header of the page.
deleteHeaderFiles();	Erase the header.
gettingExistingFile();	Pass on the name of the project that the user is in.
gettingTaskList(projectName);	Get the existing array area in Firestore's field.

gettingPath(taskName: string);	Get all the documents in collection in Firestore and pass on that value.
gettingField(field: string, taskName: string);	Get the description of the file and file's name
gettingFilePath(fileName: string, taskName: string, eachDesFileElement: element);	Get the actual file from Firebase Storage via file's name.
gettingFileIcon(fileName: string, eachDesFileElement: element)	Consider the type of the file and append the icon to the document.

4.5 Notes.js

In this file, it is referred to the Notes page, which will show all the notes that members have created. Moreover, users are able to create new notes from a button on this page. Each note shows the details which are the note's header, description, and note's created time.

Table 5. List of functions in Notes.js

Function	Description
callHeader();	Get the header of the page.
deleteHeaderNotes();	Erase the header.
closeProject();	Hide the task when the user clicks on the button.
createNotes();	Set the data from user input into Firestore.
gettingExistingNotes(proJectName: string);	Get the existing data from the database.

createNotesContainer(notesName: string);	Create containers for existing notes for user to see.

4.6 MyWork.js

Table 6. List of functions in MyWork.js

Function	Description
columns.forEach((column)=> {..})	Adding a checkbox for users, if their user's information does not include that task.
test(taskName: string);	Set and update the document in Firestore.

4.7 MyWorkPage.js

In this file, it is connected to Task.js file. As mentioned, users are able to click on each task. After the popup popped, there will be a check button under the "Add to my work". Each user can click on it once, and it will disappear. It will keep the data in Firestore of that task and show in another sidebar tab called "My Work".

Table 7. List of functions in MyWorkPage.js

Function	Description
gettingMyWorkList();	Getting existing tasks that users have.
gettingMyWorkDoc(taskName: string);	Get all the docs according to the user's id.
gettingMyWorkField(taskName: string, docID: string);	Get the value according to the key in the Firestore document field. Also append the data to the document.

getStatusTask(taskName:string, proJectName: string, dataKeyNew: string);	Getting value according to key and append to the document.
getStatusTaskWhenNooColumn(taskName: string, projectName: string);	Returning the value in array according to the taskName.
toggleCol();	Change the CSS using toggle

4.8 Homepage.js

In this Javascript file, it does not include the path to the database. Mean that it is just about the function of the mainpage which is sidebar tab functions and blurring the page functions.

Table 8. List of functions in Homepage.js

Function	Description
addBlur();	Adding blurring using activate and set CSS
removeBlur()	Removing blur using activate and set CSS
logout_blur();	Toggle activate to the document.
closeProject();	Toggle activate and call function.
moveActivateTab();	Calculate the pixel for the tab.
changeLink();	Set activity for class when click tab.
changeNav();	Set activity when click on navigate tab.

4.9 Login.js

For Login.js, this file does not include the path to the database, in other words, Login.js includes only the function for users to type into the login page. For example, clicking on the eye icon to see the password as a readable text instead of dots, or moving the HTML elements after users click on the text input, it will transition beautifully.

Table 9. List of functions in Login.js

Function	Description
addcl();	Adding "focus" to the element that is clicked which is parentNode.
remcl();	Remove "focus" to the element from parentNode.
passwordInput.addEventListener('input', ()=>{..})	Check if the eye icon for viewing the password
eyeIcon.addEventListener('click',()=>{..})	Change class for the eye icon to be opened eye or closed eye.

4.10 LoginData.js

For this file, it will check the email and password of the users, if it existed in the database or not. This file only uses library document from Firebase called "signInWithEmailAndPassword".

Table 10. List of functions in LoginData.js

Function	Description
loginForm.addEventListener("submit", (e)=>{...})	Using document called "signInWithEmailAndPassword(auth, email, password)" to check the existing

4.11 Signup.js

In this Signup.js file, it contains the path to the Authentication Firebase which will create new users from the signup page. Moreover, this file also contains the function for users in the sign up page such as clicking on the eye icon to see the readable password text.

Table 11. List of function in Signup.js

Function	Description
signUpForm.addEventListener('submit', ()=>{...})	Get the value input form user and update the display name according to variable.
createUserStorage(user: user.data);	Set the array into document for users.
moveh5();	Adding "focus" to parentNode
backh5();	Removing "focus" from parentNode
passwordInput.addEventListener('input', ()=>{..})	Check if the eye icon for viewing the password
eyeIcon.addEventListener('click',()=>{..})	Change class for the eye icon to be opened eye or closed eye.

4.12 CreateStartUp.js

For the CreateStartUp.js file, this is connected to the HomePageData.js file. In other words, this file will function on the HomePageData.js which will allow users to create startups into the database. After users have created startups, users are then able to create projects that refer to startups that already exist. Moreover, when users are creating new startups, they could add an icon of the startup and file reference to the startup. This will keep data in Firebase Storage.

Table 12. List of functions in CreateStartUp.js

Function	Description
fileInput.addEventListener(‘change’, ()=>{..})	Push the file that is selected and append it to the document.
selectedLogo.addEventListener(‘click’, ()=>{..})	Let the user click to choose the file they want.
uploadData();	Set the data according to the value from user's input and also upload the file to Storage
cancelCreateSTU();	Removing the “activate” from class to close the input startup for users.

Chapter 5

Results

5.1 Web Application

Users can view the projects that have been created, the startup firm name, and the due date of the projects on the home page. In addition, the user can see the buttons for adding new projects and startup businesses.

Sign up page for the web application that enables users to create account entering their first name, last name, email address, and password.

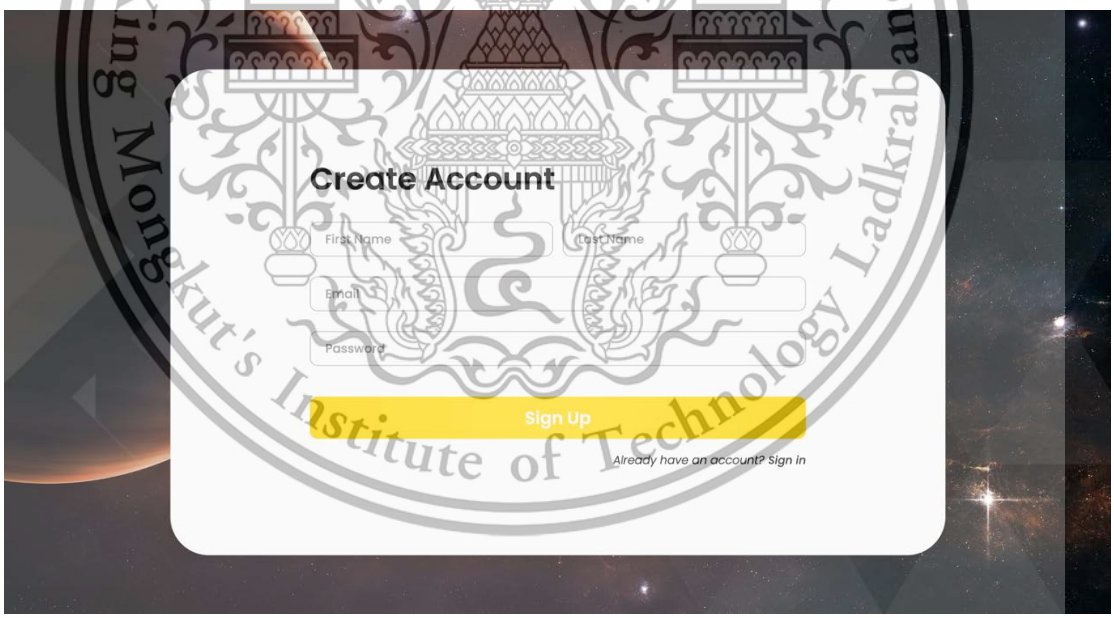


Figure 6. Sign up page

Log in page of the web application allows users to access the website by entering their username, which is their email address, and password.

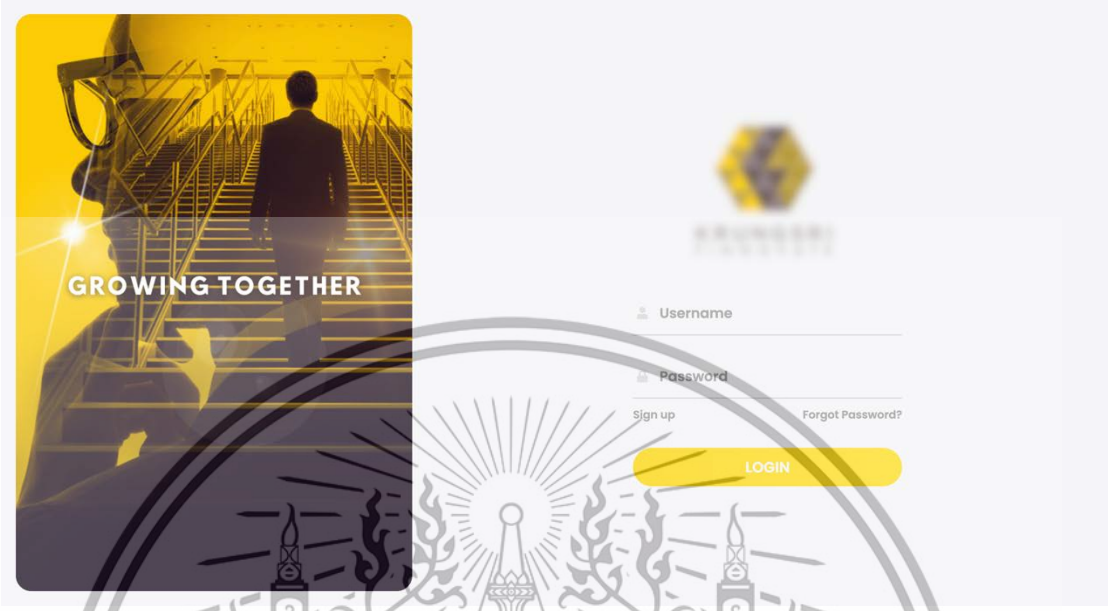


Figure 7. Log in page

Users can view the projects that have been created, the startup firm name, and the due date of the projects on the home page. In addition, the user can see the buttons for adding new projects and startup businesses.

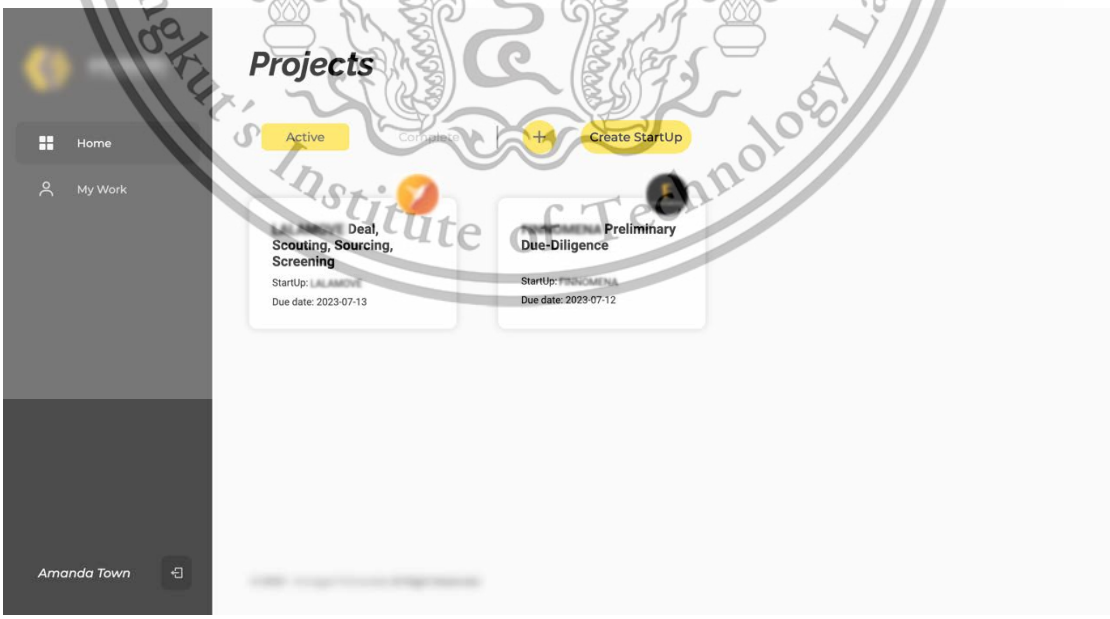


Figure 8. Home page

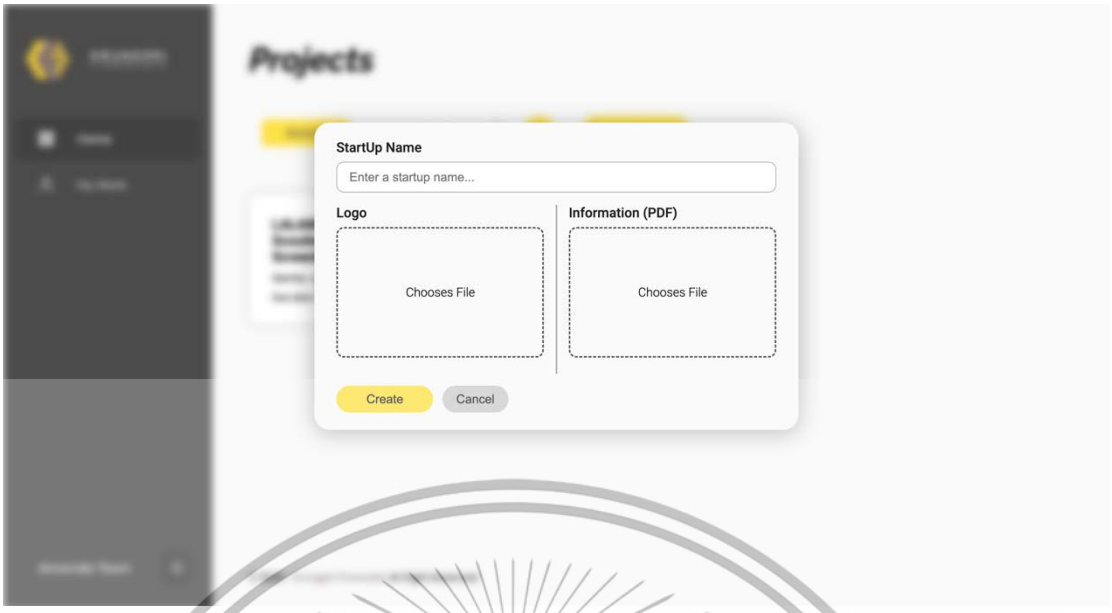


Figure 9. Create startup

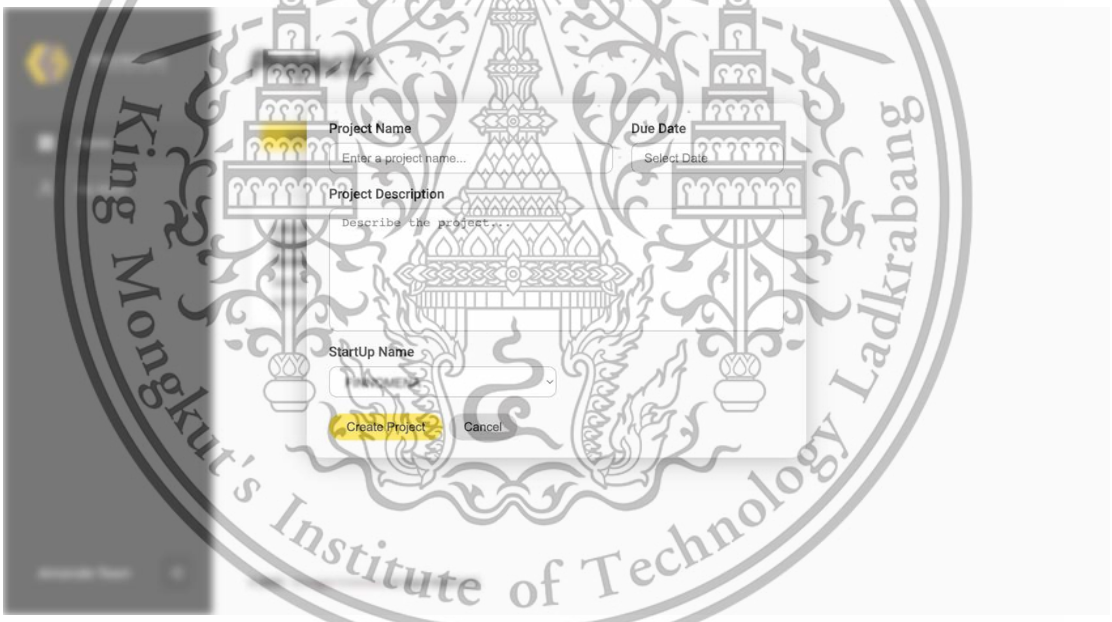


Figure 10. Create project

The active tasks for the selected project are visible to users on the task page. Each work will be categorized into one of five statuses: planning, pending, in progress, review, and completed. Users can clearly observe the status of each task as well as an overview of the tasks for this project on this page. Users can add new tasks, view details for already-existing tasks, add the task to their work and modify the status of tasks by dragging and dropping them into the desired column.

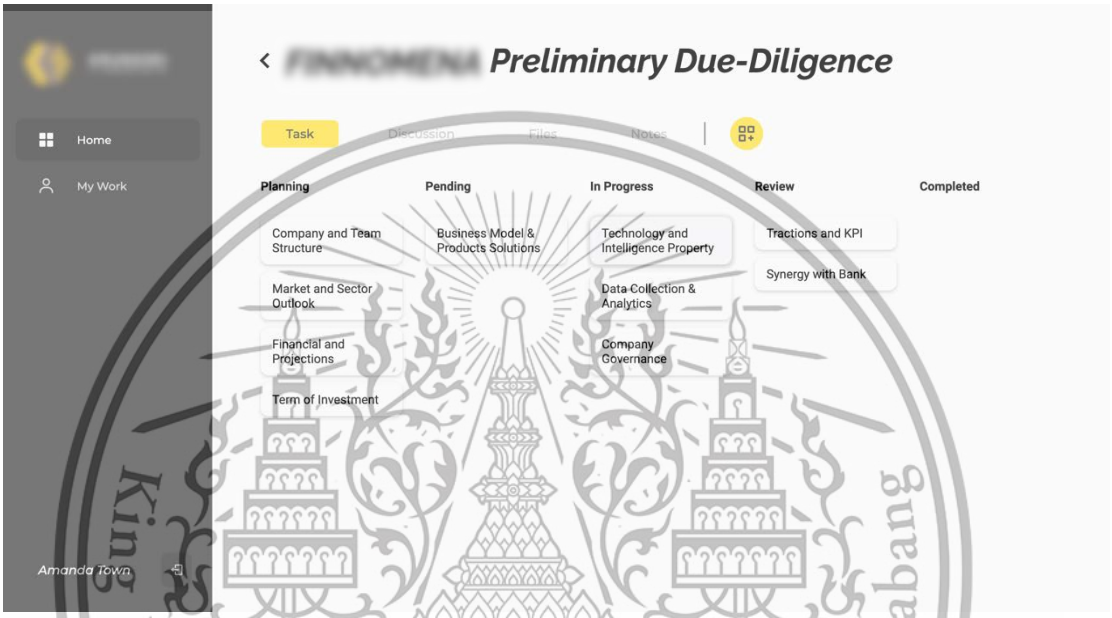


Figure 11. Task page

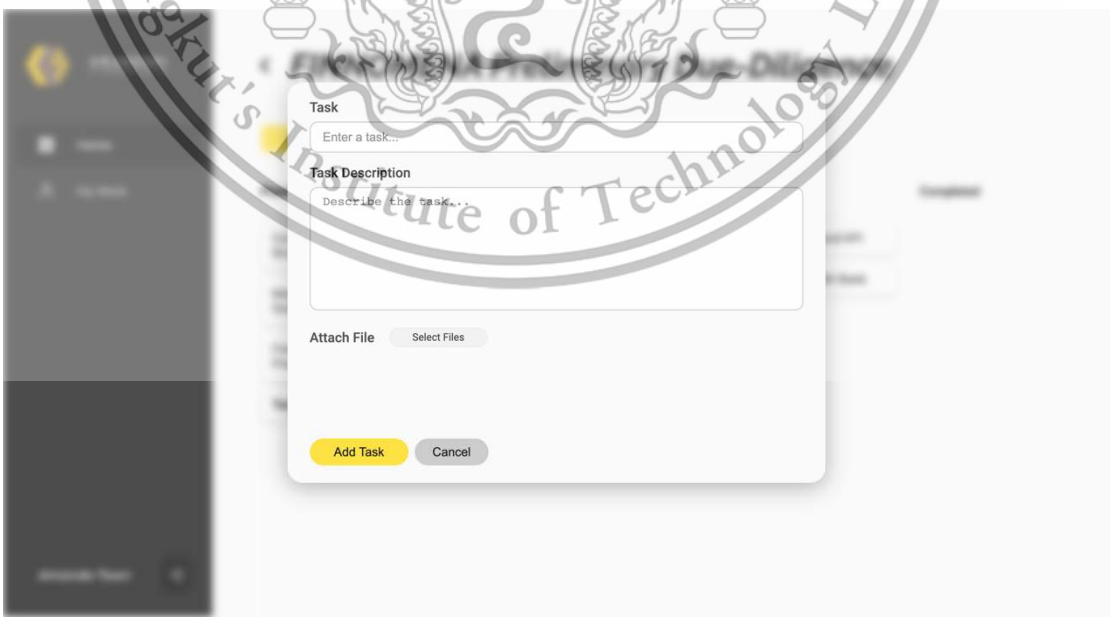


Figure 12. Create task

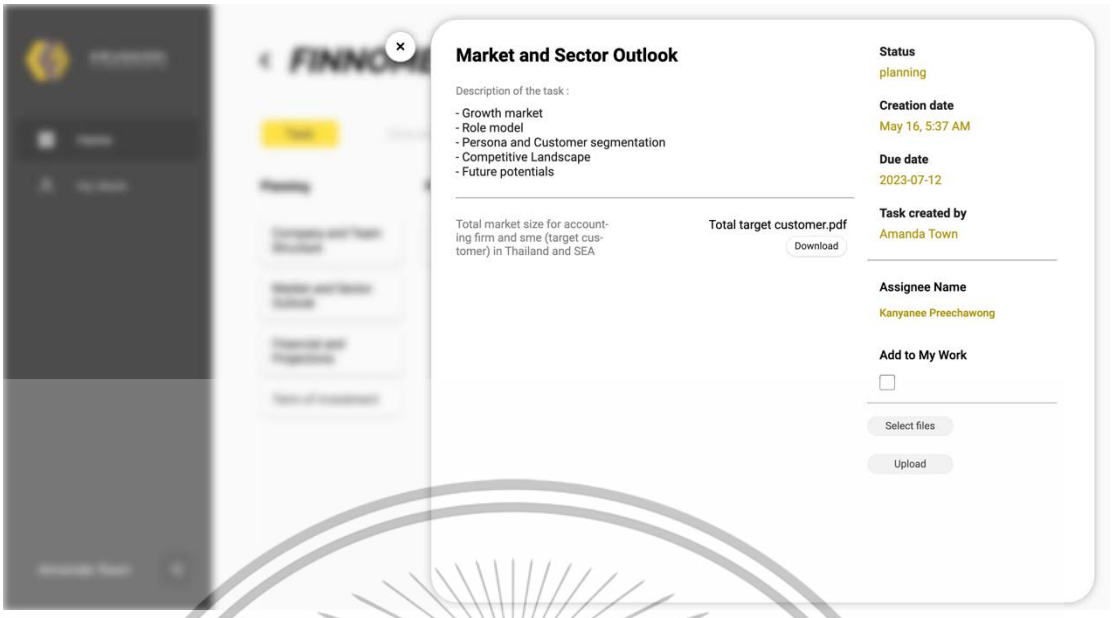


Figure 13. Details of the task

On the discussion page, the user can view all of the conversation topics for this project. By selecting the discussion box, the user can view the description, reference task, creation date, and name of the creator, as well as contribute more comments. The user can create a new discussion subject by clicking the "Create Discussion" button and providing the topic, description, and reference task.

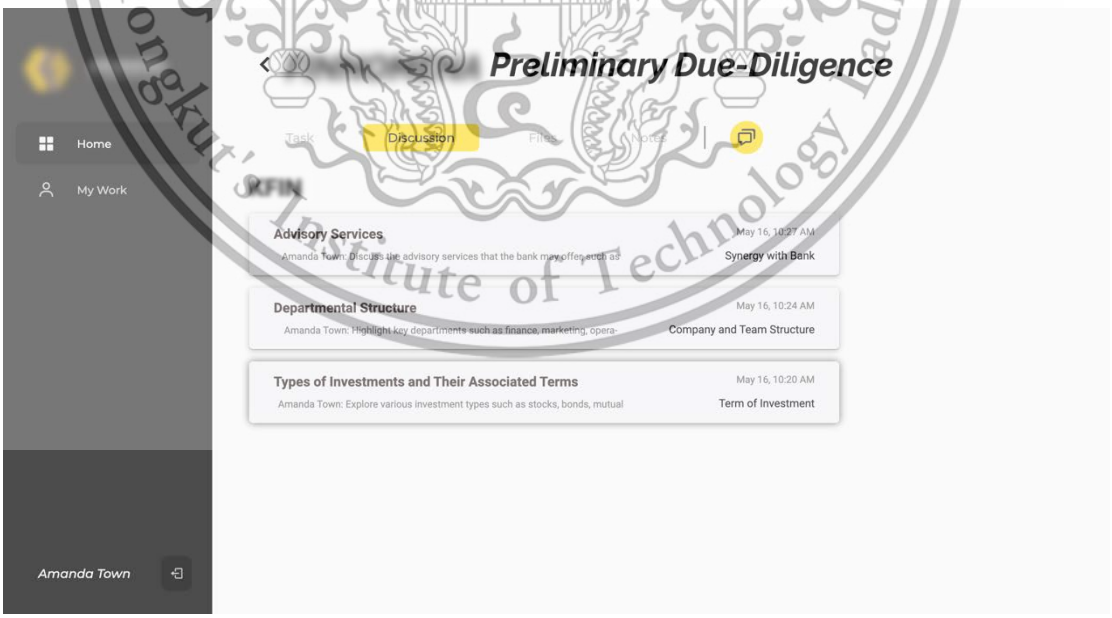


Figure 14. Discussion page

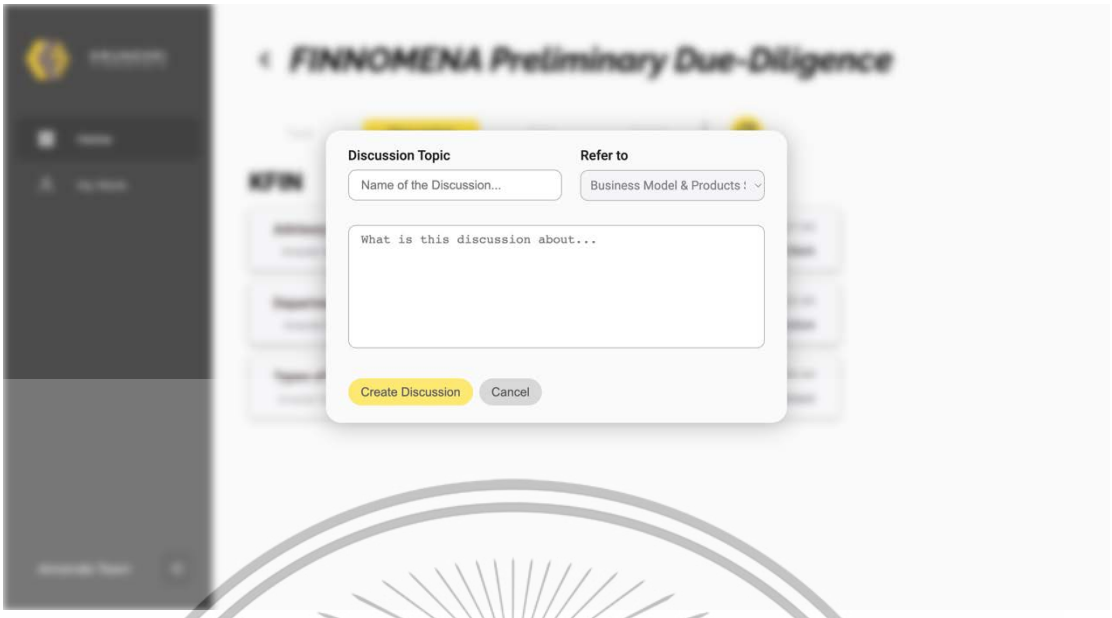


Figure 15. Create discussion

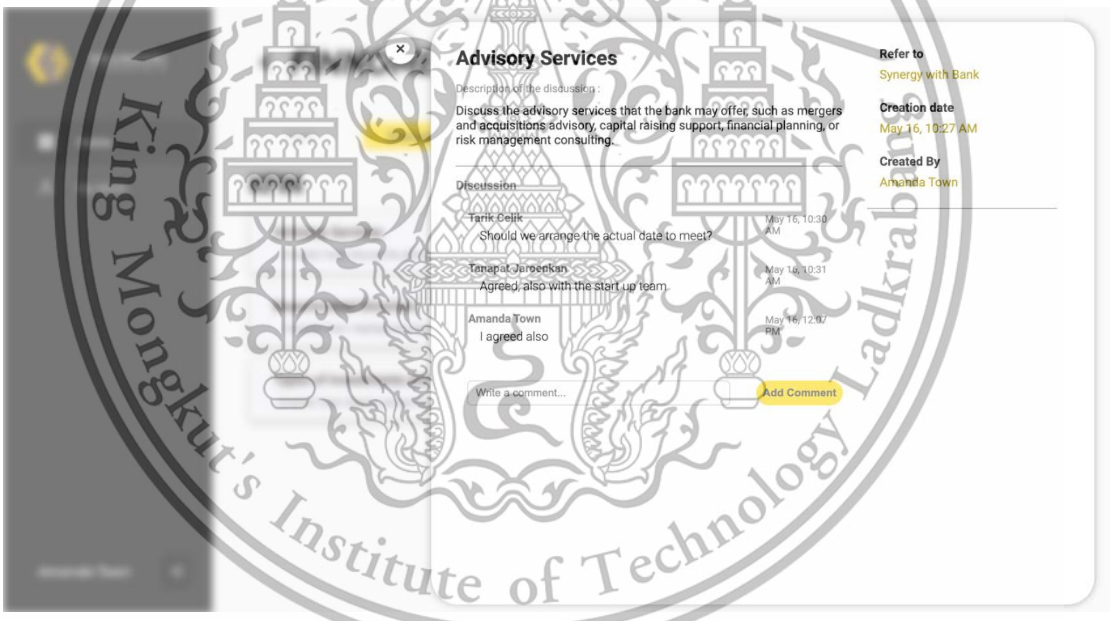


Figure 16. Details of discussion

The user can view every file that has been uploaded to the project, along with the name and description of the file, on the file page. By clicking the download icon, the user can download the file to their local device.

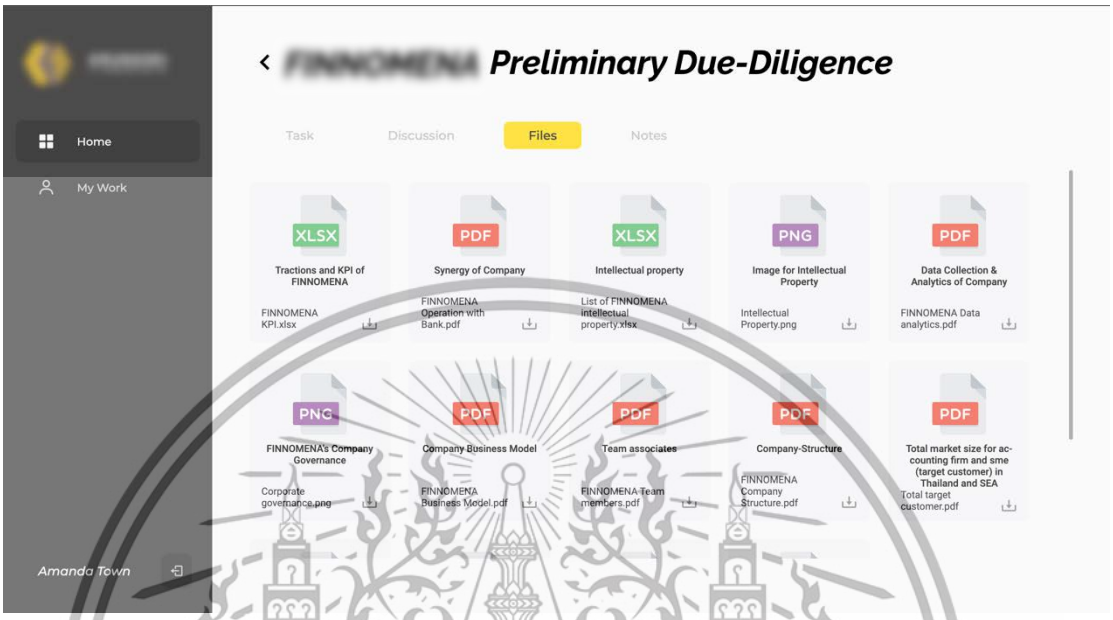


Figure 17. Files page

The user can view all notes that have been generated along with their header, contents, creator, and creation time on the note page. When the user clicks the create button, a new note can be created by typing the note header and description.

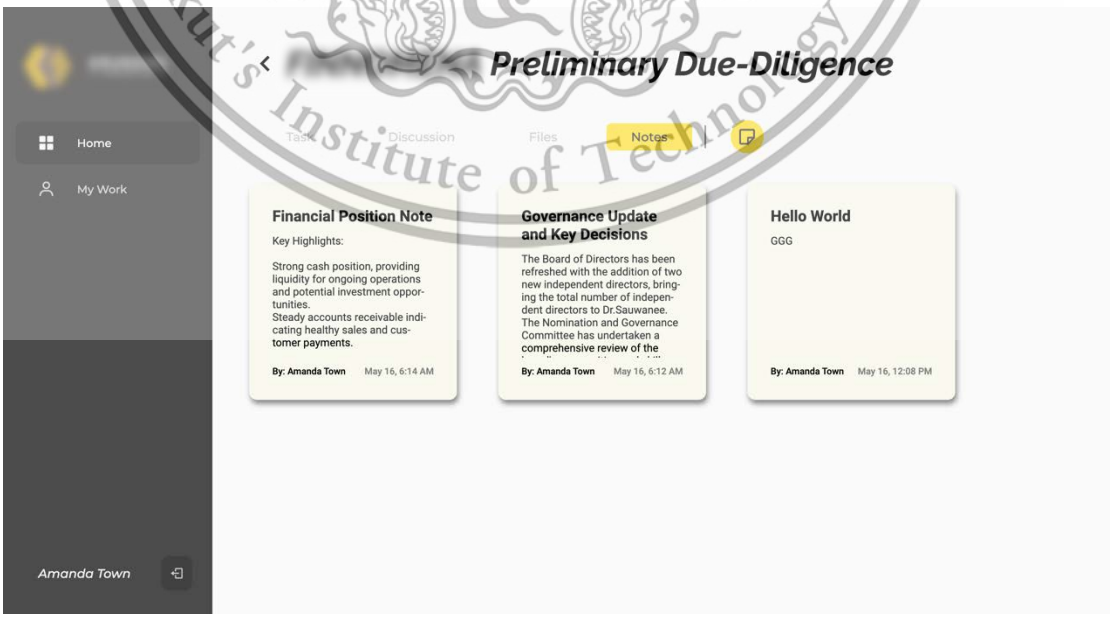


Figure 18. Notes page

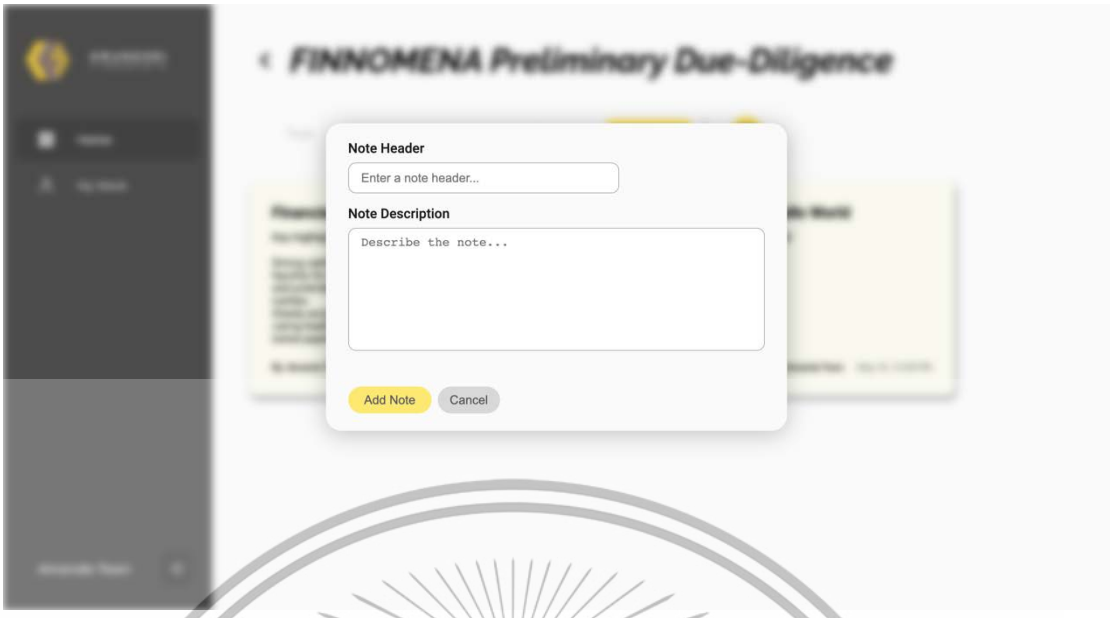


Figure 19. Create note

All tasks that have been assigned to the user are displayed on the my work page, organized by project name. The user can view the task name along with the task status by expanding the details of the project.

Figure 20. Page my work

Chapter 6
Conclusion and Future work

6.1 Conclusion

The project aims to improve the workflow of the companyAAA, by providing a management tool web application that is suitable for the organization and helps increase work efficiency.

The web application consists of three parts: the front end, the back end, and the database. The front-end website serves as a graphical user interface for users and clients to create their own projects, tasks, discussions, and notes. Users and clients can also submit files to support the tasks on which they are working. The backend server is in charge of responding to web application requests, storing data in databases, and uploading files to cloud storage buckets. All information in the web application is stored in the cloud storage and real-time database of Firebase. This web-based application was developed using three programming languages: HTML, CSS, and JavaScript. The markup language HTML (Hypertext Markup Language) is used for creating and structuring web content. CSS (Cascading Style Sheets) is a stylesheet language used to describe the appearance and layout of HTML documents, whereas JavaScript is used to add dynamic behavior to web applications.

6.2 Future work

As can be seen, this project could benefit the staff of this company. It provides employees with a clearer picture of all tasks, enables them to distribute duties they have been given, and adds more channels that will improve collaboration with others. This online application needs to be created so that consumers or startup businesses that apply for funding with companyAAA can use it, which will make the project more comprehensive. Furthermore, in order to avoid sensitive data being stolen or destroyed, developers need to understand more about data security.

Developers should implement several crucial methods to strengthen a web application's security. To avoid widespread vulnerabilities like SQL injection and cross-site scripting (XSS), they should place input validation and sanitization first. SQL injection attacks must also be avoided by using prepared statements or parameterized queries rather than simply concatenating user input into database queries. To protect user credentials, strong authentication procedures should be used, such as salt-enhanced password hashing. By using output encoding or HTML escaping techniques and imposing security headers like Content Security Policy (CSP), developers should also concentrate on preventing XSS attacks. It is necessary to use secure session management techniques, such as creating distinctive session IDs, employing secure cookies, and establishing suitable timeouts. Role-based access control (RBAC) aids in ensuring adequate access control and permission. Implementing anti-CSRF tokens and confirming request origins are necessary for defense against cross-site request forgery (CSRF). By transferring encrypted data through HTTPS, transmission security can be increased. Developers should take important measures such as routinely upgrading and patching software components, performing security tests and assessments, creating strong error handling, and logging systems, configuring servers securely, and informing users about security best practices. Developers can improve the overall security posture of web apps by continually putting a strong emphasis on data security and remaining up to date on the most recent threats and defenses.

Bibliography

[1] *Information Architecture Basics | Usability.gov*. (n.d.). Information Architecture Basics | Usability.gov. https://www.usability.gov/what-and-why/information architecture.html

[2] *Information architecture (IA) คืออะไร? อยากเป็น UX Designer ต้องรู้ - Designil*. (2021, July 24). Designil. https://www.designil.com/information-architecture/

[3] *A Beginner's Guide To Information Architecture in UX [2023]*. (2021, April 13). CareerFoundry.https://careerfoundry.com/en/blog/ux-design/a-beginners-guide-to-information-architecture/

[4] *What Is Information Architecture & Why Does It Matter?* (2022, September 13). What Is Information Architecture & Why Does It Matter? https://blog.hubspot.com/website/information-architecture

[5] Gaudiose, K. (2019, September 5). *SEO UX Play: Information Architecture & Linking Hierarchy*. Search Engine Journal. https://www.searchenginejournal.com/information-architecture-linking-hierarchy/322297/

[6] UI/UX Studio, T. (2022, September 19). *Hierarchical Structures (In Information Architecture)*. Think 360 UI/UX Studio. https://think360studio.com/blog/hierarchical-structures-in-information-architecture

[7] *What Are Wireframes? | Wireframing Academy | Balsamiq*. (n.d.). What Are Wireframes? | Wireframing Academy | Balsamiq. https://balsamiq.com/learn/articles/what-are-wireframes/

[8] *What is User Experience (UX) Design?* (n.d.). The Interaction Design Foundation. https://www.interaction-design.org/literature/topics/ux-design

[9] *The Definition of User Experience (UX)*. (n.d.). Nielsen Norman Group. https://www.nngroup.com/articles/definition-user-experience/

[10] *What is User Interface (UI) Design?* (n.d.). The Interaction Design Foundation. https://www.interaction-design.org/literature/topics/ui-design

[11] *What is a Graphical User Interface? Definition and FAQs | HEAVY.AI*. (n.d.). What Is a Graphical User Interface? Definition and FAQs | HEAVY.AI. https://www.heavy.ai/technical-glossary/graphical-user-interface

[12] *What Is GUI? Graphical User Interfaces, Explained*. (2022, August 25). What Is GUI? Graphical User Interfaces, Explained. https://blog.hubspot.com/website/what-is-gui

[13] *Introduction to HTML*. (n.d.). Introduction to HTML. https://www.w3schools.com/html/html_intro.asp

[14] *HTML5: The Language That Runs the Internet - Spiceworks*. (n.d.). Spiceworks. https://www.spiceworks.com/tech/tech-general/articles/what-is-html-five/

[15] *HTML5 - Overview*. (n.d.). HTML5 - Overview. https://www.tutorialspoint.com/html5/html5_overview.htm

[16] *HTML5 Introduction (Start Your Web Development Journey)*. (n.d.). Tutorialstonight. https://www.tutorialstonight.com/html/html-introduction

[17] *CSS Introduction*. (n.d.). CSS Introduction. https://www.w3schools.com/css/css_intro.asp

[18] *JavaScript Tutorial*. (n.d.). JavaScript Tutorial. https://www.w3schools.com/js/

[19] *JavaScript | MDN*. (n.d.). JavaScript | MDN. https://developer.mozilla.org/en-US/docs/Web/JavaScript

[20] *JavaScript basics - Learn web development | MDN*. (n.d.). JavaScript Basics - Learn Web Development | MDN. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

[21] *JavaScript — Dynamic client-side scripting - Learn web development | MDN*. (n.d.). JavaScript — Dynamic Client-side Scripting - Learn Web Development | MDN. https://developer.mozilla.org/en-US/docs/Learn/JavaScript

[22] *Web APIs*. (n.d.). Web APIs. https://www.w3schools.com/js/js_api_intro.asp

[23] *Client-side web APIs - Learn web development | MDN*. (n.d.). Client-side Web APIs - Learn Web Development | MDN. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs

[24] *Firebase Realtime Database*. (n.d.). Firebase. https://firebase.google.com/docs/database

[25] *Visual Paradigm Online*. (n.d.). Visual Paradigm Online. https://online.visual-paradigm.com/drive/#diagramlist:proj=0&dashboard

[26] *dbdiagram.io - Database Relationship Diagrams Design Tool*. (n.d.). dbdiagram.io - Database Relationship Diagrams Design Tool. https://dbdiagram.io/d/6462618fdca9fb07c41e63a6

[27] Talalaev, A. (2021, February 27). *11 Ways To Improve Your Web Application Security - Patchstack*. Patchstack. https://patchstack.com/articles/web-application-security/

[28] Nair, S. V. (2021, January 29). *How to Improve Web Application Security?* How to Improve Web Application Security? https://beaglesecurity.com/blog/article/how-to-improve-web-application-security.html

[29] Lohvynenko, A. (2023, January 21). *How to Secure Web Applications From Vulnerabilities in 2023*. MobiDev. https://mobidev.biz/blog/best-practices-to-secure-web-applications-from-vulnerabilities

[30] A. (2022, November 23). *26 Ways to Improve Your Web Application Security*. Code Signing Store. https://codesigningstore.com/26-ways-to-improve-your-web-application-security

[31] Perdikis, N. (2021, January 17). *Preliminary Due Diligence - Devensoft*. Devensoft. Retrieved October 20, 2022, from https://www.devensoft.com/blog/preliminary-due-diligence/

[32] Rayne. (2011, October 20). *Valuations 101: Scorecard Valuation Methodology - Gust*. Scorecard Valuation Methodology. Retrieved October 20, 2022, from https://gust.com/blog/valuations-101-scorecard-valuation-methodology/

[33] Berkus. (2009, November 4). *The Berkus Method – Valuing the Early Stage Investment. | BERKONOMICS*. BERKONOMICS | Dave Berkus' Business Insights. Retrieved October 20, 2022, from https://berkonomics.com/?p=131/