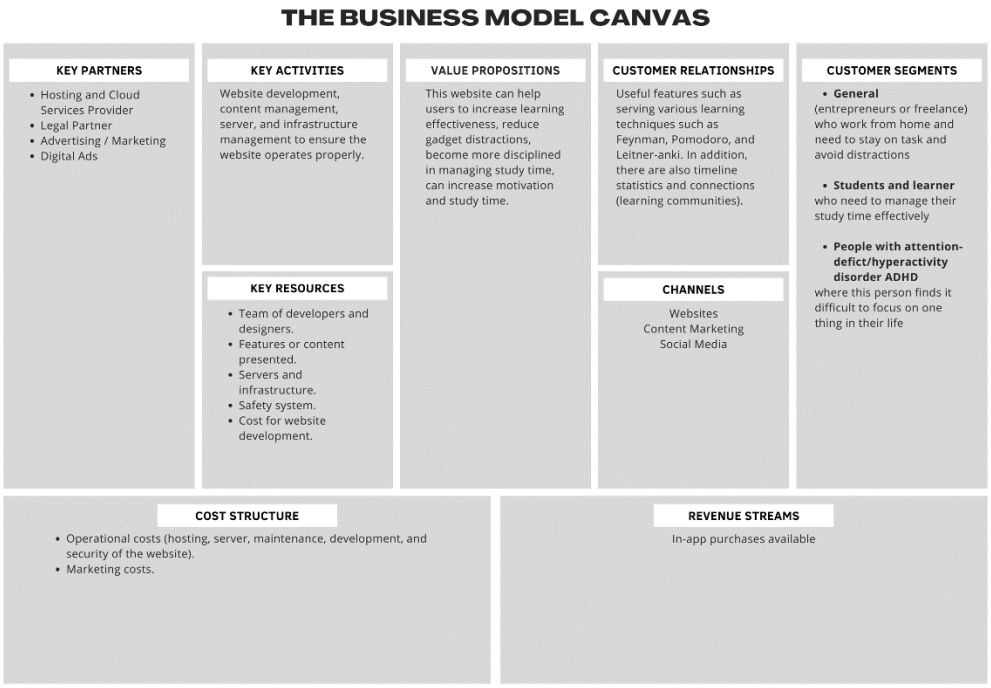
Nerd Project – Software Engineering

Member:

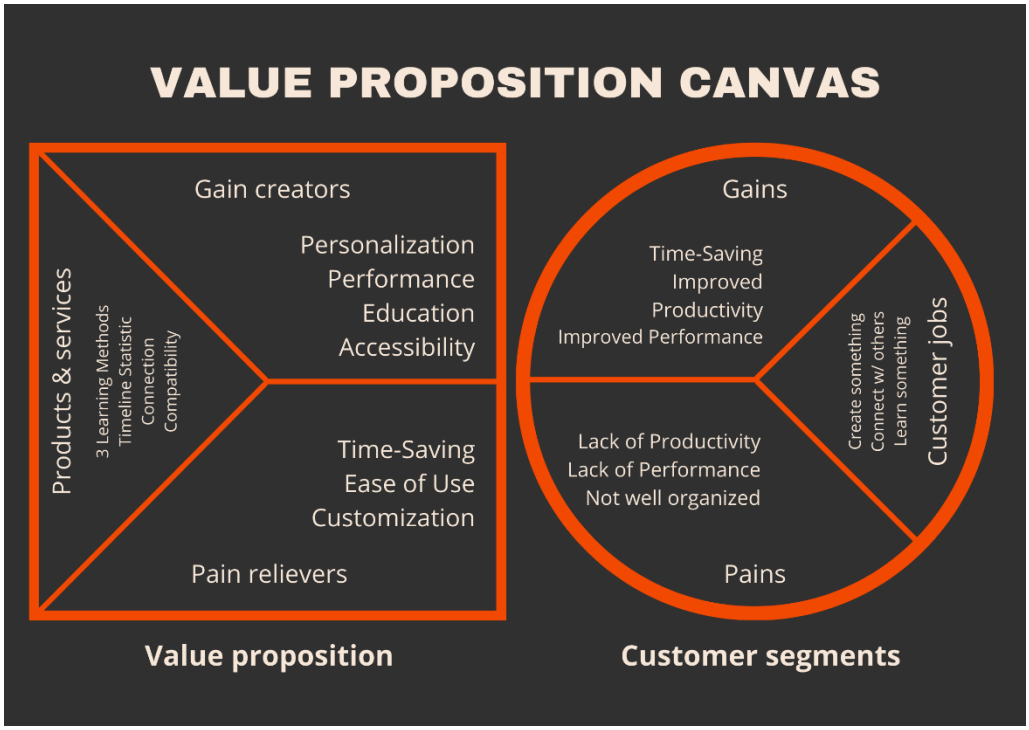
1. Chantikka Riffka Ramadhani Citra Karisma – 2502023650
2. Dora Kalifa Dharmawan – 2540135473
3. I Gusti Ayu Ngurah Stita Maharani – 2540134262
4. Pristian Budi Dharmawan – 2501983105

Requirement Engineering

A. Business Model
- Business Model Canvas



- Value Proposition Canvas



B. Stakeholders

Stakeholder	
List of Stakeholders	**Role Description**
Customer	Determine the project scope based on the demand. The scope can be approved by both parties by signing contracts with us.
Media	Meanwhile, the media have indirect engagement with our project since this project is aimed at educating, media can affect PM decisions and our software reputation
End-User	Our end-user can be divided into three main characters: student, lecturer, and general. They can affect our project based on what they demand such as design or functionality
Project Manager	PM is the main actor in our project development since they will take critical decisions for the team. PM in our project will work to lead and organize every primary stakeholder.
Developer	The developer in our project will be developing the websites and every functionality inside of them. They will develop and check the main function and its features before the launch date.
UI/UX Designer	UI/UX will take a critical impact on the main design of our product such as how users will interact with the product and how users feel "safe" using our product by looking at its appearance
Business Analyst	BA will be analyzing our customer idea to be reported to PM, this could lead to a critical decision that PM will take in it. BA will give the developer and UI/UX several checklists to fulfill the needs of the customer. Also, BA will make predictions to understand the budget and time to create our project.

C. Functionality

LIST OF FUNCTIONALITY	DESCRIPTION	RELATED STAKEHOLDERS
Transaction Handling	Transaction handling is part of an application that is responsible for coordinating transactions across one or more resources.	Customers, Vendors/Suppliers, Financial institutions, Business partners
Certification Requirements	Certification Requirements are The requirements specified in this agreement for quality, compatibility, and/or performance of a Software Title and to the extent not inconsistent with the foregoing standards.	Certification bodies, Professionals, Employers, Customers/Clients.
Reporting Requirements	Reporting requirements is a document that outlines the necessary elements of a project or system.	Regulators, Executives, and management, Investors, Employees, Customer
Administrative Function	Administrative function is a function normally associated with the routine operation of government including tax, assessment and collection, personal services, purchasing, records management services, data processing, warehousing, equipment repair, and printing.	Executive leadership, Management and staff, Customer/Client, Suppliers, and partners.
Authorization Levels	Authorization levels refer to the different levels of access and permission granted to users within a system or organization. In many systems, there are different levels of authorization, and the levels typically include Guest or Anonym User, Registered User, Contributor, Editor, Administrator, or superuser.	IT Department, Executive leadership, Management, Customers/Clients.
Audit Tracking	Audit Tracking or audit logging is the process of recording and tracking all user activity within a system or app. The purpose of audit tracking is to provide a detailed and comprehensive record of all user actions for securities and compliance purposes.	IT Department, Executive leadership, Compliance, Internal and external auditors, Customers/Clients.
External Interface (API, GUI, CLI, Hardware Interface)	External interfaces are critical components of many software systems, as they allow systems to communicate with other systems and external entities, enabling the exchange of data and information. Proper design and implementation of external interfaces are crucial for ensuring a software system's security, reliability, and functionality.	Industry association, Competitors.

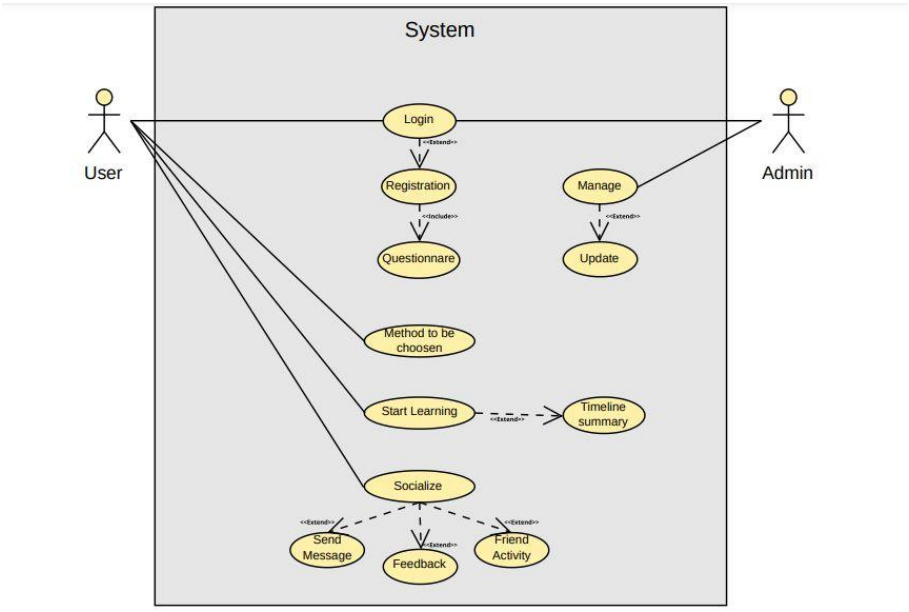
Historical Data Management	Historical data management refers to the process of collecting, storing, and maintaining historical data for future use. Historical data is any data that is generated by a system or organization and has a historical record or timeline associated with it. Examples of historical data include transaction records, financial data, sensor data, and other types of time-stamped data.	Data Management, Data Analyst, Business, and Operation.
Legal and Regulatory Requirements	Legal and Regulatory Requirements refer to the rules and standards established by government bodies or other regulatory agencies that organizations must comply with to operate legally and ethically.	Legal and compliance, Regulatory agencies, Customers/Clients, Investors, and shareholders.

D. Non-Functionality

LIST OF NON-FUNCTIONALITY	DESCRIPTION
Usability	User comfort and interest when accessing the system that has been created (UI UX)
Portability	Ease when users will access the system, create user-friendly media, and regularly update system updates
Reliability	The durability of a system in working in certain situations (determining the limits of how the system can work to avoid overload / improve hosting services)
Performance	Limitations on the speed of the system when operating. The need for response, timing, and throughput of the system. Pay attention to the delay, late response of the system
Safety	High security requires a system so that user privacy is not disturbed. Such as verification authentication when you want to reset your password to make it more secure (number of password characters or pin).
Security	Security to ensure the safety of the system (access limit, database that admin can only change, data that can be backed up automatically)

Software Model

A. Use Case Diagram & Expanded Description



Role: Admin

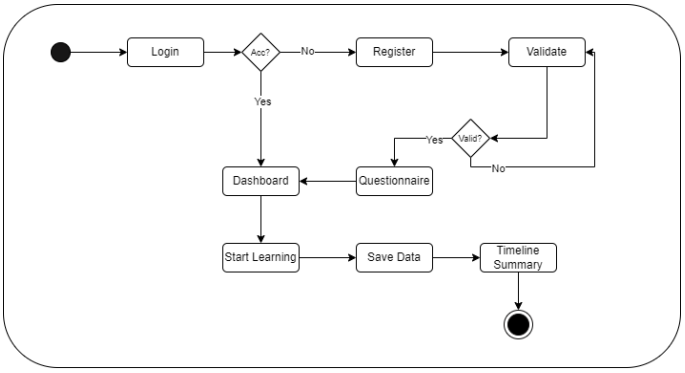
- Login: Use case which allows admin to login into the system by providing valid credentials. The login system includes "Registration" and "Questionnaire".
    - Registration: New users create personal accounts or profiles within an application to uniquely identify the user within the system and provide appropriate access to the features and functionality of the application.
    - Questionnaire: Questions that will be answered by users to determine suitable learning methods.
- Manage: Use cases that allow admins to manage existing data. In this system, it is included in updating data by the admin.
    - Update: The act of updating information, data, or settings related to system administration or application management. Admins have special access and authority to modify and update various aspects of their applications or system. Some examples of updates that admins can do include:
        - Settings Update: Admins can update app or system settings, such as changing display preferences, adjusting business rules, or setting user access permissions.
        - Data Update: The admin can update data in the system, for example adding, editing, or deleting data entries. This could involve managing users, updating profile information, or updating data related to business entities such as products or services.
        - Security Update: The admin is responsible for ensuring system security. They can make security-related updates, such as changing passwords, enabling two-factor authentication, or implementing additional security policies.
        - Software Updates: Admins can be responsible for updating software used in the system, including updating applications, operating systems, or related hardware components.

- Update Policies and Procedures: The admin can update policies and procedures related to the use and management of the system. This may include changes to rules, usage procedures, or privacy policies.

Role: User

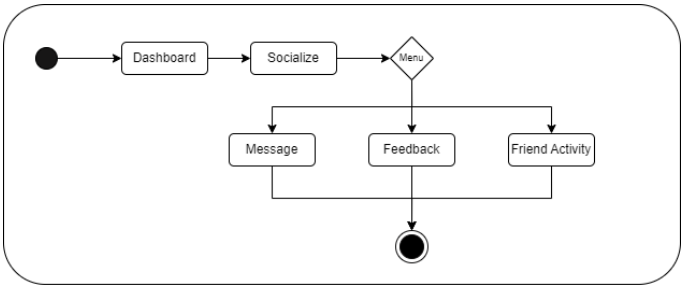
- Login: The process of entering the website using a registered account.
  - Registration: The process of creating a new account for users who do not yet have an account on the website.
  - The questionnaire: Questions that will be answered by users to determine suitable learning methods.
- Method to be chosen: The method that the user will choose for the learning process.
- Start learning and timeline summary: After the login process and determining the learning method the user can start the learning process on the website. In the timeline summary section, the system will display the progress of learning progress so that users can see the progress of learning and continue the material that needs to be studied again.
- Socialize: Includes socializing with other users/friends. There are several sections in socializing, among others:
  - Send Message: Send messages to other users or admins.
  - Feedback: Assessment of applications used by users
  - Friend Activity: Can view friend's activity such as "online", "busy", etc.

B. Activity Diagram



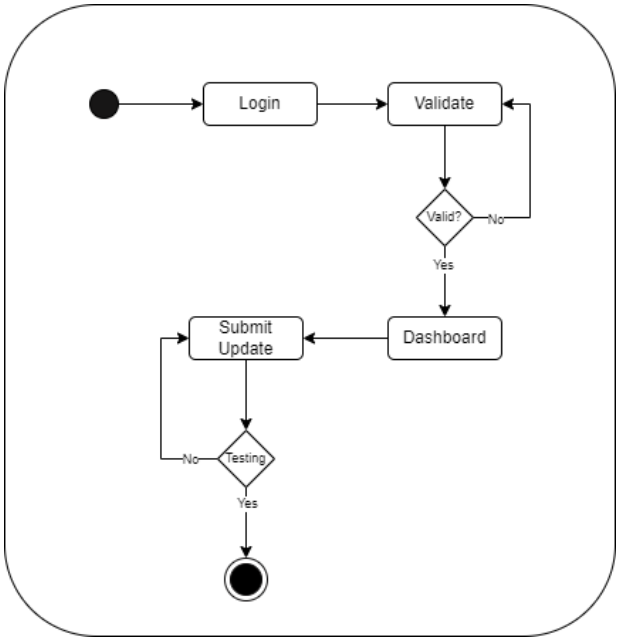
When they open this website for the first time, users log in. If it does not work acc, it will redirect to -> register. After registering, the data will be validated; if the data is valid, it will be forwarded to the questionnaire, and if it is not valid, it will be redirected back to validate. If the user has completed the questionnaire, the user will be directed directly to the dashboard page.

If the user already has an account, when logging in, the user will be immediately directed to the dashboard and start learning. If so, the user can save the learning data, which will be displayed in the summary timeline.



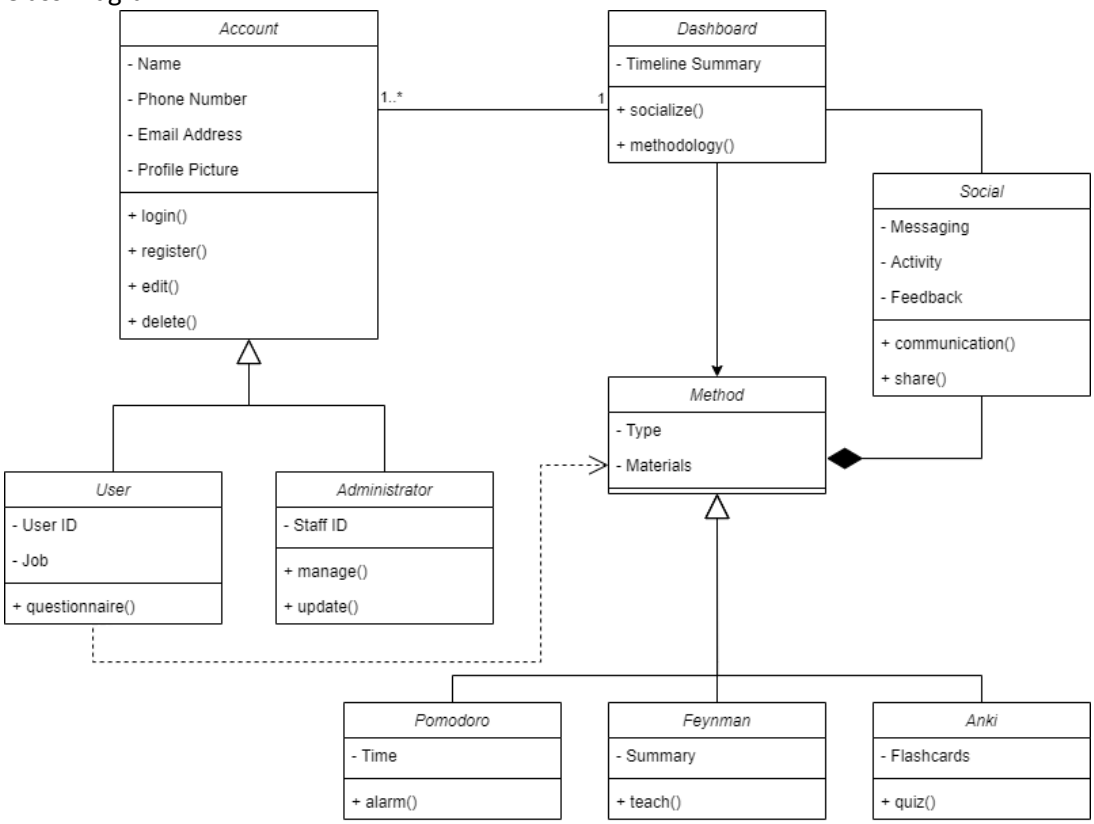
This dashboard leads to socializing. Socialization has three menu features: message, feedback, and friend activity:

- Message: In this feature, the user can interact with his friends by exchanging messages
- Feedback: Users can provide input about this website
- Friend activity: The user can see what activities his study friends are doing



Login ->validate if the data is correct, it will immediately be directed to the dashboard. If the data is incorrect at the time of validation, it will be redirected to validate again. Dashboard ->submit an update. If testing is wrong, then it will return to submit an update. If it is correct, it will save.

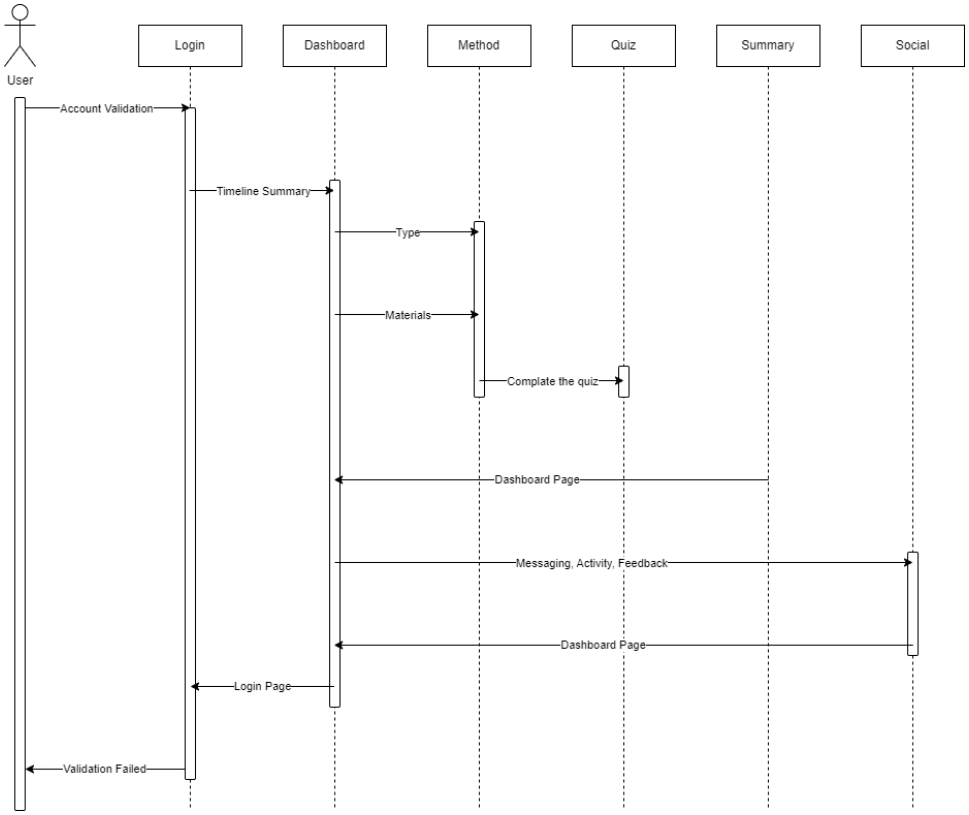
C. Class Diagram



We define several class diagrams which represent our system flow. The figure above, we could describe as follows:

1. Account
   The account class is used to define the type of the account. If the account matches our staff credentials, then it will be redirected to the administrator homepage. If it's not it will be redirected to the user dashboard instead.
   a. User
      This class is used for describes the user's account and describe what kind of method he will use.
   b. Administrator
      Meanwhile, this class is used to determine what kind of operations that could be done by the administrator.
2. Dashboard
   This class describes every account summary. Users will see the timeline summary of their learning progress. The dashboard has a direct association with the method that every user uses.
3. Social
   This class is used to define what account will appear in their social life which they can see their friends' statuses, messaging with their friends, and others.
4. Method
   The method is an abstract class that contains the type of the method and the materials when you learn. However, after or when you write all of the materials inside of your method, it will redirect you into further step in these three method:
   a. Pomodoro
   b. Feynman
   c. Anki

D. Sequence Diagram

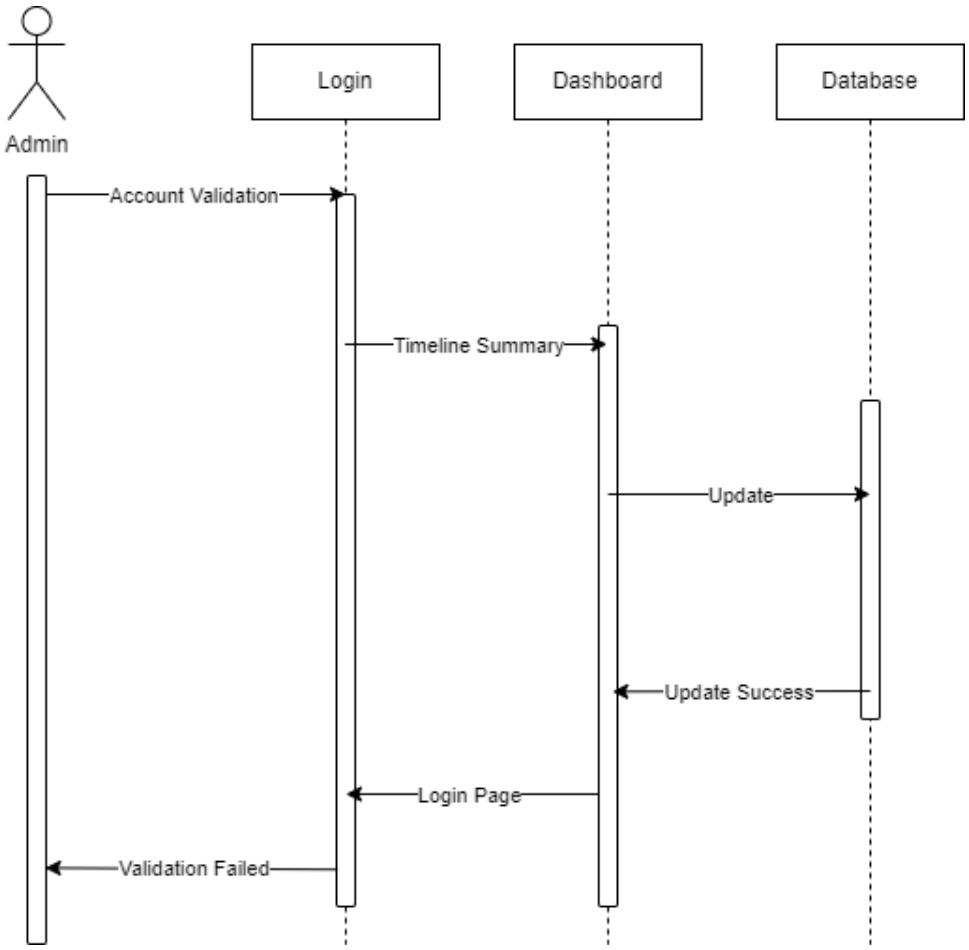


The interaction steps in this sequence diagram are as follows:

The user starts the app.

1. The system displays the main page:
   - The system responds to user requests by displaying the main page of the application.
2. The user selects the "Sign In" feature:
   - The user clicks the "Login" button to access the login feature into the system.
3. The system displays the login form:
   - The system displays a login form that prompts the user to enter credentials (for example, username and password).
4. The system validates credentials:
   - The system checks for matches between the credentials entered by the user and the data stored in the system.
5. The system authenticates the user:
   - If the credentials entered are valid, the system authenticates the user and grants access to the relevant features. Otherwise, it will return to the login page.
6. The system displays the home page:
   - The system responds to successful authentication by displaying a home page containing additional information and functionality.
7. The user performs an action:
   - Users interact with the system by performing various actions, such as writing messages, performing searches, or selecting certain menu items.
8. The system responds:
   - The system displays responses to user actions, such as displaying search results or saving entered data.

9. User exits the application:
    - The user selects the "Exit" feature to exit the application.



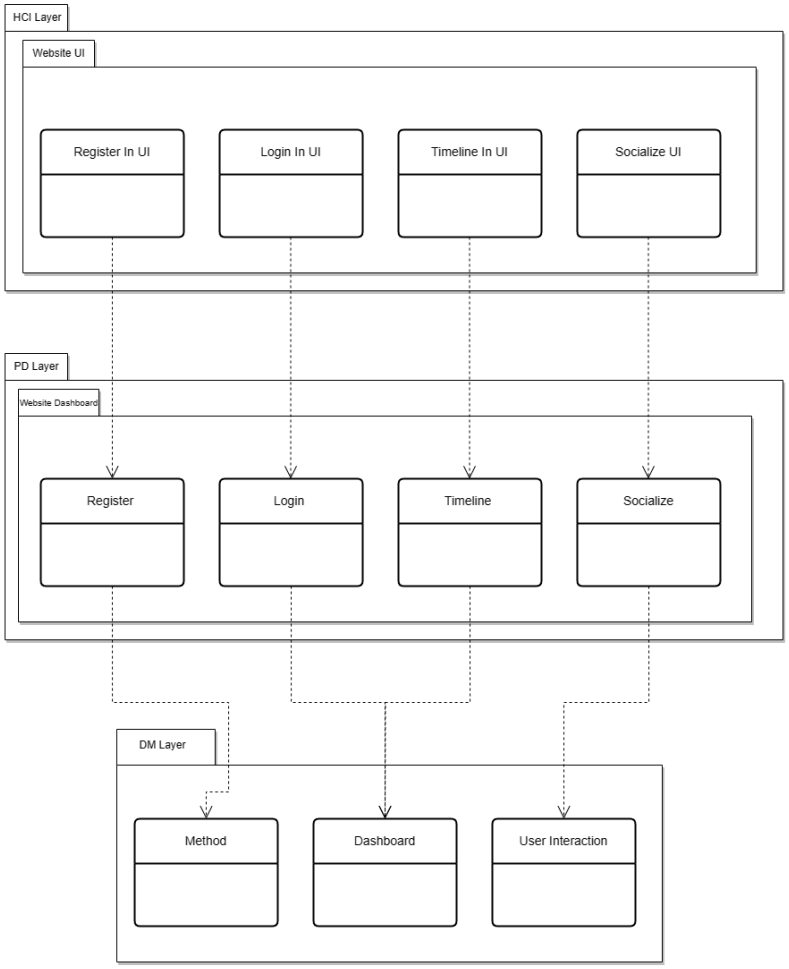
The interaction steps in this sequence diagram are as follows:
The Admin starts the app.

1. The system displays the main page:
    - The system responds to user requests by displaying the main page of the application.
2. The user selects the "Sign In" feature:
    - The user clicks the "Login" button to access the login feature into the system.
3. The system displays the login form:
    - The system displays a login form that prompts the user to enter credentials (for example, username and password).
4. The system validates credentials:
    - The system checks for matches between the credentials entered by the user and the data stored in the system.
5. The system authenticates the user:
    - If the credentials entered are valid, the system authenticates the user and grants access to the relevant features. Otherwise, it will return to the login page.
6. The system displays the home page:
    - The system responds to successful authentication by displaying a home page containing additional information and functionality.
7. The system displays the admin panel page:

- The system responds to admin requests by displaying the admin panel page which contains the required functions and features.

8. Admin makes changes to users:
   - Admins make necessary changes to users, such as changing access rights, resetting passwords, or updating user information.
9. System updates user data:
   - The system updates user data according to changes made by the admin.
10. Admin save changes:
    - The admin saves the changes made to the user by clicking the "Save" button or similar action.
11. The system validates changes:
    - The system validates the changes made by the admin to ensure the suitability and integrity of the data.
12. The system confirms:
    - The system confirms to the admin about the success of the changes made to the user.

E. Package Diagram



In this package diagram, there are three packages: the Human-Computer Interaction (HCI) package, the PD layer package, and the DM layer package. In the HCI layer, there is a Website UI package that contains the following:

- Register In UI
- Login In UI
- Timeline In UI
- Socialize In UI

Whereas the PD layer package contains the Website Dashboard package, which consists of Register, Login, Timeline, and Socialize, which leads to the DM Layer Package, which contains:

- Method
- Dashboard
- User Interaction

F. User Interface (Prototype)

The raw file of the UI & UX Design of "Nerd." could be accessed through this link below:
Click Here or https://bit.ly/UIUXNerd
The UI & UX of the "Nerd." demonstration could be accessed through the link below:
Click Here or Prototype Demo.mp4
Front-end of Nerd could be accessed through this link below: Click Here or https://github.com/iannn07/S4-SE-Project-Nerd

Software Testing

A. Software Testing Strategy

STRATEGY	TESTING OBJECTIVE	METHOD	TESTER
Unit Test	Verifying each class and module to ensure every component is working as expected.	Black Box	Programmer
Integration Test	Verifying that every different unit of code will work together correctly.	Black Box	Programmer
Validation Test	Checking that does the software meets the requirements of the user or not.	White Box	QA, Programmer, and User
System Test	Verifying that the system meets all the requirements and works as expected in all scenarios (doing its manner).	Black Box	QA

B. Software Testing Result

DATE	MODUL	TESTING SCENARIO	EXPECTED RESULT	RESULT
14/06/2023	Authentication	A user could have an account for Nerd	The user is free to choose whether login or register. If they take to log in, they will be redirected to the dashboard. Meanwhile, if they took the register, they will be redirected to the log in page.	Finished
14/06/2023	Dashboard & Menu	After, Login, the user will be automatically redirected to their dashboard and could interact with the menus	The user is free to choose the option on the left sidebar and log out from their account.	Finished
14/06/2023	Landing Page	Without any authentication, the	The landing page works as it should for the guest users.	Finished

		landing page of Nerd could be accessed by guest user		

Project Schedule

Task	Role	Prerequisite Task	Start Date	End Date
Communication and Planning	All	Project Name and Team	26/02/2023	01/03/2023
		Software Purpose		
		list of Stakeholders		
		List of Problems		
		Outline of Solution		
		Proposed Software Process Model		
Requirement Engineering	All	Business Model Canvas	08/03/2023	08/03/2023
		Value Proposition Canvas		
		List of Stakeholders		
		List of Functionality		
		List of Non-Functionality		
Software Model	Ian & Dora	Use Case Diagram and Expanded Description	21/03/2023	27/03/2023
	Stita & Caca	Activity Diagram		
	Caca & Dora	Class Diagram	25/03/2023	
		Sequence Diagram	17/05/2023	17/05/2023
		Package Diagram		
	Ian & Stita	User Interface		02/06/2023
Software Testing	Ian	Software Testing Strategy	14/06/2023	14/06/2023
		Software Testing Result		
Software Metric	Stita	Function Point	14/06/2023	14/06/2023
Software Configuration Management	Stita	Software Configuration Management Model and Software Configuration Description	14/06/2023	14/06/2023

Software Metric

Function Point

1. Login page

No.	Information Domain Value	Count	Description	Total
1.	External Input (EIs)	6	New user registration, login with Google, log in with GitHub, log in with the created account, password reset request, and fill out questionnaires to determine learning methods.	6

2.	External Output (EOs)	2	Password reset instruction and learning method results.	2
3.	External Inquiries (EQs)	1	Help request.	1
4.	Internal Logical Files (ILFs)	2	User data and questionnaire data.	2
5.	External Interface Files (EIFs)	2	Google and GitHub API	2

2. Dashboard page

No.	Information Domain Value	Count	Description	Total
1.	External Input (EIs)	2	Choosing learning methods and log out account.	8
2.	External Output (EOs)	2	Weekly activity statistics and monthly notes progress.	4
3.	External Inquiries (EQs)	0	-	1
4.	Internal Logical Files (ILFs)	3	User data, weekly activity statistic data, and monthly notes progress data.	5
5.	External Interface Files (EIFs)	0	-	2

3. Learning page

No.	Information Domain Value	Count	Description	Total
1.	External Input (EIs)	7	create a learning section, create notes, create flashcards, set learning time, start self-learning, end self-learning, and start flashcards.	15
2.	External Output (EOs)	3	displays notes, displays flashcards that the user has created, and displays flashcards when the section starts flashcards.	7
3.	External Inquiries (EQs)	0	-	1
4.	Internal Logical Files (ILFs)	3	learning section data, flashcard data, and notes data.	8
5.	External Interface Files (EIFs)	0	-	2

4. Socialize page.

No.	Information Domain Value	Count	Description	Total
1.	External Input (EIs)	5	Send messages, search for contacts, create new groups, and send files and images.	20

2.	External Output (EOs)	3	receive messages, files, and images.	10
3.	External Inquiries (EQs)	0	-	1
4.	Internal Logical Files (ILFs)	3	Message data, contact data, and group data.	11
5.	External Interface Files (EIFs)	3	File data, image data, and profile data	5

Complexity characteristic table

Complexity scoring:
0 = No Effect
1 = Incidental
2 = Moderate
3 = Average
4 = Significant
5 = Essential

No.	Characteristics	Value
1.	Data communication complexity level	3
2.	The complexity level of distributed processing	3
3.	Performance complexity level	3
4.	Configuration complexity level	1
5.	Software usage frequency level	4
6.	Frequency level of data input	4
7.	User-friendliness level	4
8.	Data Update Frequency Level	3
9.	Data processing complexity level	3
10.	Level of Reusability of Program Code	4
11.	Level of Ease of Installation	3
12.	Level of ease of software operation (backup, recovery, etc.)	4
13.	Software level made for multiple organizations/companies/clients	2
14.	Level of complexity in following changes/flexibility	4
Total Value		45

Function Point Count Total

No.	**Infromation Domain Value**	**Count**	**Weighting Factor**			
			Simple	Average	Complex	Total
1	EIs	20	3	4	6	80
2	EOs	10	4	5	7	50
3	EQs	1	3	4	6	4
4	ILFs	11	7	10	15	110
5	EIFs	5	5	7	10	35

Count Total	279

Function Point Estimated

Count Total = 279

Complexity Factor = 45

Result:

$$FP = count\ total\ \times [0.65 + 0.01\ \times \sum F_i]$$

$$FP = 279\ \times [0.65 + 0.01\ \times\ 45] = 306.9$$

Software Configuration Management

- **Version Control System (VCS)**

  In handling software configuration management Nerd utilizes the Git version control system which is used to manage and maintain modification history such as adding, deleting, and modifying websites. This Git repository can store all files and code used in website development so that developers can work together on different parts. This makes it easier for developers to track and manage applications with different versions or commonly referred to as version naming such as v1, v2, v3, and so on.

- **Tracking Changes**

  Tracking changes is used to track every change that occurs in the website code created. This makes it easier to identify every change consistently and systematically, manage bugs and errors that may occur, and facilitate recovery if there are needs in the previous version. Thus, tracking changes allows developers to manage changes to website development more efficiently.

All Project Submission

PPT : Click Here or https://www.canva.com/design/DAFbpUZfp8A/LGd95db2VNAf0DKtPkfmpA/edit?utm_content=DAFbpUZfp8A&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Presentation Video : Click Here or Presentasi.mp4

Reports : This document is Nerd. report