**ASSIGNMENT 2 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 4 HND Diploma in Computing** | | |
| **Unit number and title** | Unit 7: Software Development Life Cycle | | |
| **Submission date** | 6-4-2022 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | PHAN HONG HANH | **Student ID** | BH00354 |
| **Class** | SE06201 | **Assessor name** | DINH VAN DONG |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | HANH |

**Grading grid**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P5 | P6 | P7 | M3 | M4 | M5 | M6 | D3 | D4 |
|  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

Introduction

In the software development process, one of the most crucial steps is gaining trust and support from the customers. At this stage, we have successfully persuaded the Net Education Institute (NEI) to select our project for development. This trust is not only a significant milestone but also an opportunity for us to execute a meaningful project that adds value to NEI.

In this report, we will present the process of analyzing and designing software for NEI's project. Each task is designed to provide a detailed and comprehensive insight into the necessary steps to understand and meet the specific requirements of this project.

We hope that this report will not only serve as a useful tool for the software development process but also as an opportunity for us to share our knowledge and experience in building and deploying effective software solutions.

1. (P5) Undertake a software investigation to meet a business need.
2. Identify the stakeholders, theirs roles and interests in the case study.

In today's business environment, effective software adoption is a key factor in providing solutions to complex business needs. In this context, the Net Education Institute (NEI) has recognized the importance of having a modern and flexible software system to support their operations. To ensure that they were moving in the right direction, a comprehensive software survey was conducted.



* Identify the stakeholders, theirs roles and interests(các bên liên quan)

**NEI Management:**

*Role:* NEI management plays a crucial role in defining the direction of development and business objectives of the institute.

Benefits: NEI management needs an efficient software system to manage and oversee academic operations and resource management effectively, thereby improving performance and service quality of the institute.

**Faculty Members:**

*Role:* Faculty members are directly involved in using the software system to manage classes, teaching materials, and student information.

Benefits: The software system needs to provide diverse teaching and management tools, helping faculty members optimize the teaching and learning process and enhance interaction with students.

**Students:**

*Role:* Students use the software system to access information about courses, class schedules, and academic results.

Benefits: The software system needs to offer a user-friendly, easy-to-use, and flexible interface, enabling students to access information and interact with the institute conveniently.

Information Technology (IT) Department:

*Role:* The IT department is responsible for building, deploying, and maintaining the institute's software system.

Benefits: The software system needs to be developed and maintained efficiently, ensuring stability, security, and scalability, thus helping the IT department optimize system management and maintenance effectively.

**Academy Management Board:**

*Role:* The academy management board participates in decision-making processes regarding the selection and implementation of the software system.

Benefits: They require a robust and flexible software system to support strategic management and decision-making for the institute, thereby enhancing its competitiveness and development.

**Requirement definition of the project ( FRs and NFRs)**

***What are the functional requirements?(FRs)***

Functional Requirements (FR) specify the specific functions that a software system needs to perform to meet user needs. Specifically, functional requirements outline what the system should do, including activities, features, and services it provides to users. Functional requirements are typically expressed as specific requirements that shape the expected behavior of the system.

**The role of these stakeholders**

NEI Board of Directors As NEI's senior management group, they have the role of deciding on the institute's strategy and development direction. Interested in optimizing administrative processes and providing the best learning experience for students.

Faculty and Staff Play a key role in implementing NEI's academic and administrative processes. Desire to have effective management tools to optimize teaching processes and support students.

Students Are the main group that this project system is aimed at. Expect a convenient online learning experience,from registering for courses to accessing academic resources.

Software Development Team Consists of analysts, developers and testers, playing a key role in system

development and deployment. Payattention to understanding customer requirements and desires to provide a perfect software solution.

****

**List out FRs in NEI project**

1. **User Authentication:** The system provides a login function, allowing users to enter login information (username and password) to access the system.
2. **Course Management:** Users have the ability to create, edit, and delete courses in the system. They can also add, edit, and delete information about schedules, instructors, and related documents.
3. **Grade Management:** The system allows instructors to enter grades for assignments, quizzes, and exams. It calculates total grades and displays students' final grades.
4. **Communication:** The system provides communication tools such as email or discussion forums between instructors and students.
5. **Report Generation:** The system is capable of generating reports on student progress, attendance records, and course evaluations for management and instructors.

***What are functional requirements(NFRs)?***

Non-Functional Requirements (NFR), also known as "Quality Attributes" or "Quality of Service (QoS) Requirements," refer to criteria that define aspects of a system's functionality that are not directly related to the behavior or features of the system. Instead, they define characteristics or attributes that impact the system's operation, usability, security, performance, and other non-functional aspects.

Non-functional requirements describe how the system should operate rather than what the system should do. They focus on qualities such as performance, reliability, scalability, usability, security, and maintainability. These requirements are crucial for ensuring that the system meets overall quality standards and user expectations beyond just functional capabilities.



List out NFRs in NEI project

1. **Performance:** The system needs to handle concurrent access from multiple users without significant degradation in response time. It should load course content pages within an average of 3 seconds.
2. **Reliability:** The system should maintain a continuous uptime of at least 99.9% during regular operating hours. It should recover from incidents within 5 minutes without data loss.
3. **Scalability:** The system should be able to scale to accommodate a 20% increase in user count or course load without performance degradation. It should support up to 10,000 concurrent users during peak usage periods.
4. **Usability:** The user interface should be user-friendly and intuitive, requiring minimal training for instructors and students to effectively use the system. It should adhere to accessibility standards to ensure ease of use for users with disabilities.
5. **Security:** The system should implement robust authentication and authorization mechanisms to restrict access to sensitive information. Data transmission and storage should be encrypted to prevent unauthorized access or data breaches.
6. **Maintainability:** The system should be well-documented with clear and concise documentation for administrators, developers, and end-users. It should adhere to coding standards and best practices to facilitate future maintenance and updates. Regular system backups should be performed to prevent data loss in case of system failure.

***Relationships between the FRs and NFRs***

In the Neo project, the relationship between Functional Requirements (FR) and Non-Functional Requirements (NFR) is applied closely to ensure the development of an efficient and reliable software system.

The FRs of the Neo project reflect the main features and functionalities that the system needs to provide, such as course management, user management, document creation and management, as well as interactive user interface. These requirements help identify what the system should do to meet user needs and project objectives.

To ensure that these features operate efficiently and provide a good user experience, NFRs are also applied in the Neo project. NFRs such as performance, reliability, security, and scalability are defined to ensure that the system not only operates as expected but also meets quality and security standards.

An FR of the Neo project may require the system to provide a course registration function. To ensure acceptable performance, a corresponding NFR may require the system to process registration requests within a short time and ensure that the website does not crash due to overload.

The relationship between FR and NFR in the Neo project is an integral part of the software development process, playing a crucial role in ensuring that the system not only meets functional requirements but also ensures performance, reliability, and security simultaneously.

***Top of Form***

**The difference between functional and non-functional requirements**

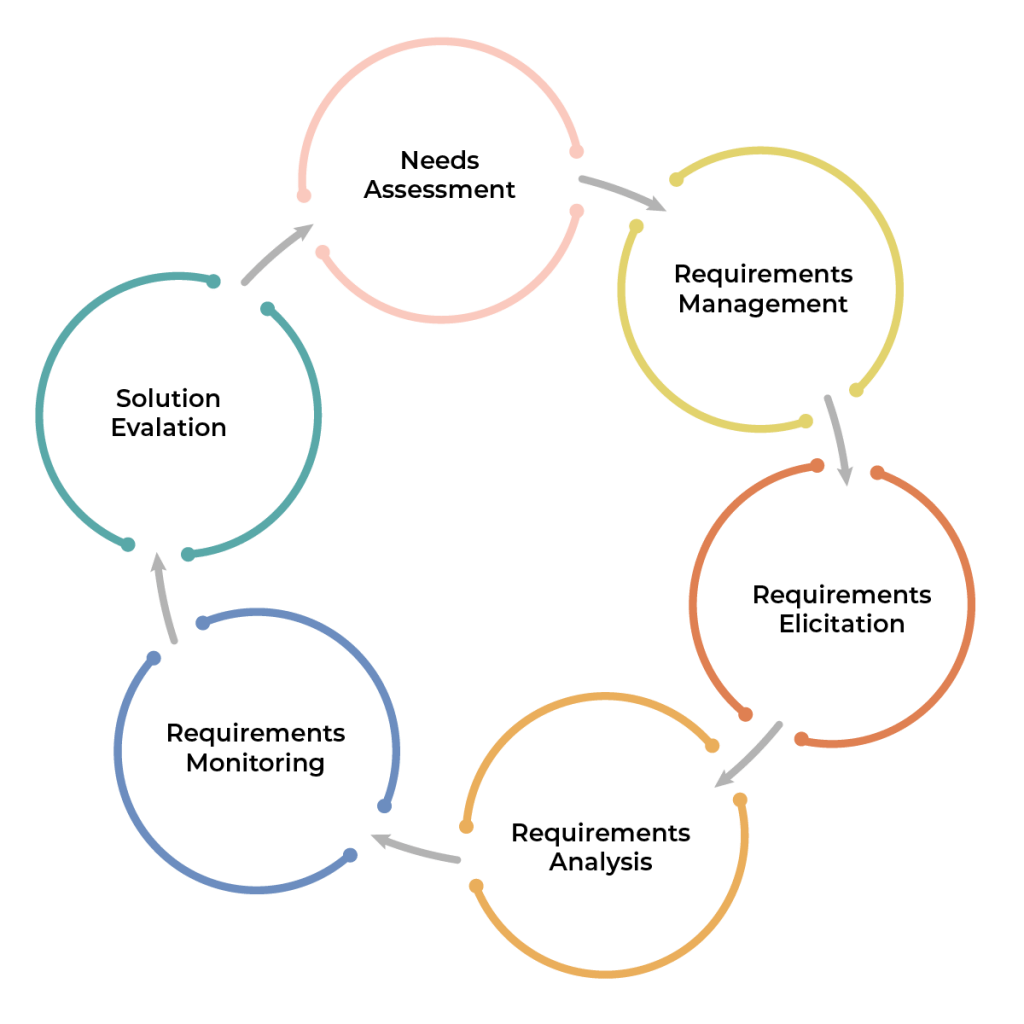
|  |  |  |
| --- | --- | --- |
| **Id** | **FRS** | **NFRS** |
| **1** | Manage student personal information, including name, address, contact information and related academic information | Scalable to handle increasing amounts of data and number of users without affecting system performance |
| 2 | Students need to be able to register for courses according to their study schedule and manage registered courses. | Ensure high performance in system processing and response, especially in subject registration and student information access. |
| **3** | Support the tuition payment process and create invoices related to tuitionfees and other expenses | Comply with security standards to ensure that students' personal information is closely protected |
| **4** | Manage and access scores, study progress and related information for students and teachers. |

*b.Discuss software quality attributes that are applicable to NRIs*

There are two terms here, "Requirements" and "Technical Requirements". Requirements are precisely defined as conditions or capabilities that a user needs to solve a problem or achieve a goal. In other words, requirements are the conditions or capabilities that a system must meet or possess in order to satisfy contracts, standards, specifications, and other formal documents.

Requirements engineering is defined as the process of identifying, documenting, and maintaining requirements. This subject covers all techniques, methods and procedures involved in identifying and managing user needs related to the system being studied.

In general, Requirements Engineering is a set of activities involved in defining and communicating the goals of a system or software and the context in which it will be used.

****So, Requirements Management acts as a bridge between the real-world needs of users, customers, and other constituents affected by the software or system and the capabilities and opportunities provided. provided by software-intensive technologies.

Works during technical requirements

There are several activities that we face when working with requirements. In engineering technical requirements submission, there are four main activities namely,

Ask for suggestions - this is the process of gathering, reviewing, documenting and understanding stakeholders and needs and forcing users to face the season. Users need information domains, existing information systems, regulations, standards, etc. Based on this information, we make requirements. We then move on to requirements and commercial analysis.

Requirements and volume analysis - analysis is the process of adjusting the needs and releasing the force of the user on the basis of the information gathered and collected. Then we move on to operational documentation.

Documentation requirements - once we have the required specifications, we move on to the documentation section. We document user needs and constraints clearly and concisely.

Validate requirements - finally, in the validation activity we insert that the season requirements should be complete, concise and clear.

Once we complete these production jobs, we iterate on them until we get an agreed document requirement that is the final specification.

What is requirements gathering?

Requirements gathering, as the name suggests, is a process of researching, understanding and documenting the exact requirements that a project needs from start to finish.

As part of the elicitation process, it is important that we ask the right questions. When I hear someone say "Customers don't know what they want," I tend to cringe. I think customers know what they want. They don't know how to express it to us. Our job is to ask the right questions so we can help them solve what they want. Sounds simple, right??

What are the steps when stimulating?

Here are the steps to request the size:

Identify information sources and requirements. This also includes identifying stakeholders.

Now, establish the project scope and define the boundary boundary system.

This operation is performed at the start of the activation process. Also, they don't need to have an order tool either.

Now, we state or select the appropriate techniques used for each information source to extract the requirements.

Finally, we are ready to prepare our documents.

**STEP 1**

Requested source

There are various sources from which we can gather our requirements. Some of them include:

Related parties

Existing system

Documents available

Competitors and other similar systems

Interface with the system

Laws and standards

Company policy

*What are stakeholders?*

Users are one of the most important stakeholders, but they are not the only stakeholders. For example, if we are building a nightclub, simply looking at potential customers won't do. We will have to include other people like staff, waiters, DJs, security guards and more in terms of how they work. Accordingly, we will collect requests from both users and employees. But then, we forget to consider the surrounding areas. Neighbors may not be the ones using the club, but they are affected by it. Therefore, their opinions and requests must also be considered.

Therefore, we can define stakeholders as individuals or organizations that benefit or lose from the success or failure of a system. Therefore, identifying project stakeholders is fundamental to achieving requirements success.

*Who are the stakeholders?*

Customers - those who pay for the development of the system. They are the ones who have the final word on what the product will be like. For an internal product, they are the product manager. Additionally, for consumer markets, consumers can act as the marketing department.

Users - users of current and future products/systems are also important stakeholders for an organization. They are true experts of the current system as well as the competition. They are the best indicators of improvements in existing systems. Their needs are what the organization must give high priority to and their ideas and suggestions must not be neglected. We also have to choose our users carefully.

Domain Experts - They are experts who know what is involved in the job. They are people who are familiar with the problems that the software or system must solve. Additionally, they know the environment in which the product will be used.

Inspectors - They are experts in government rules and regulations and the safety required by the project.

lawyers - They are experts in law and legal matters and the standards to keep in mind when developing products/systems.

Systems experts - systems experts, are the people who interact with the system to build it. They are very familiar with the system interfaces.

...

**STEP 2**

Set project scope

The following steps can be taken to establish the scope of the project:

Find out why the project was started

Assets identify the key goals to be achieved through the project

Drawing up a statement of work for the project will help you divide the work logically among team members.

List items that need to be handed over at the end of the project

Choose milestones that need to be achieved

Identify the main obstacles and limitations the team may encounter during project development

Create a list of items that are excluded from the list of scope items

Have stakeholders sign the scope document as it provides confirmation that they have been informed about the project and its contents.

**STEP 3**

Request Collection Quest

*Planning:*

Why must this specific requirement be implemented and what benefits will it bring? - Objectives of the project

Who will be responsible for creating it? - Expert for elicitation efforts

When is the best time to do it? - Scheduling estimated sources

How will it be done? - Strategy and procedures

And the risks

Transparent:

Confirm project viability. Find out if the project is really worth it

Understand issues and problems from a stakeholder's perspective

Extract the essence of the requirements stated by the stakeholders

Find better ways to get things done for users

Innovation is the key to victory

Next:

Analyze the results to properly understand the information collected

Negotiate a consistent set of requirements that are acceptable to stakeholders. Set priorities also

Document the results in the requirements specification

Requirements gathering is an incremental process. You must repeat this step as much as possible.

Now, choose an appropriate set of techniques for each requirement source. Determine the technique based on the source, system developed, etc. Remember that not all techniques can be used in all situations.

**STEP 4**

Documentation of requirements

The final step in the process is to finalize all requirements in document form. This document mainly contains user notes and requests. And these requirements will be incomplete, inconsistent, and disorganized. But this is just the starting point. Documents may be edited from time to time and things may be added or changed.

Several techniques are used to collect requests

Interviews - They are exploring ideas. They work mainly when the data is qualitative. Interviews can guide interviewees and thus encourage contact between developers and users. Furthermore, it is a time-consuming process.

Questionnaires - they answer specific questions. They are useful in providing quantitative and qualitative data. Additionally, it has a wider reach. But it must be designed carefully because the response rate is low and they are not what you need.

Brainstorming - The purpose of this technique is to generate new ideas and find solutions to problems. Usually people like domain experts and subject matter experts are included in this technique.

prototyping - This technique is used mainly when looking for unknown or missing requirements. Regular demos are done with customers so they can better understand what the product looks like.

Study existing documents - they help when we want to learn about procedures, regulations and standards. They only work in case of quantitative data. There is no need for user time although daily work will be from documented procedures.

Analysis of existing documents - Through this technique, information is collected by analyzing existing and available documents, reports and other documents. This is a very useful technique for projects that involve migration.

Use Case - This technique often involves a combination of text and graphics to enhance understanding of the requirements. Use cases are used to describe more of the 'what' part of the project and focus less on the 'how' part.

What are the benefits of Requirements Gathering?

There are several benefits of requirements gathering. These include:

Collecting requirements helps establish an accurate scope of work and budget. With the help of this, you can provide your clients with realistic budgets and release dates.

Collecting accurate requirements ensures reduced confusion during development. It also helps avoid many meetings and wasted time.

Effective requirements gathering helps develop a product that fits the customer's business and adds value to the business.

Accurate requirements gathering helps reveal requirements that remain hidden because they are so obvious.

Effective requirements gathering allows you to develop relevant functionalities and choose the best technology.

What is the problem with Collecting Requirements?

There are many problems people encounter during the requirements gathering phase. These include:

Sometimes it may be that stakeholders do not know exactly what they want and expect. So it becomes quite difficult to state the requirements precisely.

Stakeholders explain the requirements in their own words. Therefore, understanding them becomes a bit difficult.

Different stakeholders may have different and sometimes conflicting requirements.

System requirements can be influenced by organizational and political factors.

Requirements may change during the analysis phase. It is very likely that new stakeholders will emerge, which will completely change the business environment.

**6 Tips for Collecting Perfect Requirements**

Keep a Stock of "Good Questions" I believe that successful inquiry-eliciting interviews start with preparation. Many analysts think they can sit with a user and figure out what they want. That's not the situation. Analysts need to research the problem domain and think about the questions they need to ask. The fundamental difference between professional analysts and novice analysts lies in the ability to recognize situations and apply appropriate tools (i.e. questions) appropriate to the situation. Experienced analysts tend to ask the same types of questions - they know they will get the best results. When conducting an interview, watch out for instances where a particular question or particular phrasing of a question works well in giving you the information you need. When that happens, write it down. Add to the list as you become more experienced. Having these questions on hand makes preparing for interviews quicker. These questions, or versions of them, will serve you well for almost any project. Put them in your question “toolbox.”

*What “pain points” are we trying to solve?* This is a great question to solve a real business problem. We often go into projects assuming we all understand why we're doing it. Besure that. Let the user describe the pain he hopes will be alleviated by this project. I asked a user this time and they replied that they didn't know what pain this project would alleviate. Not a good scenario. An alternative to this question is to ask what user needs this project will fill.

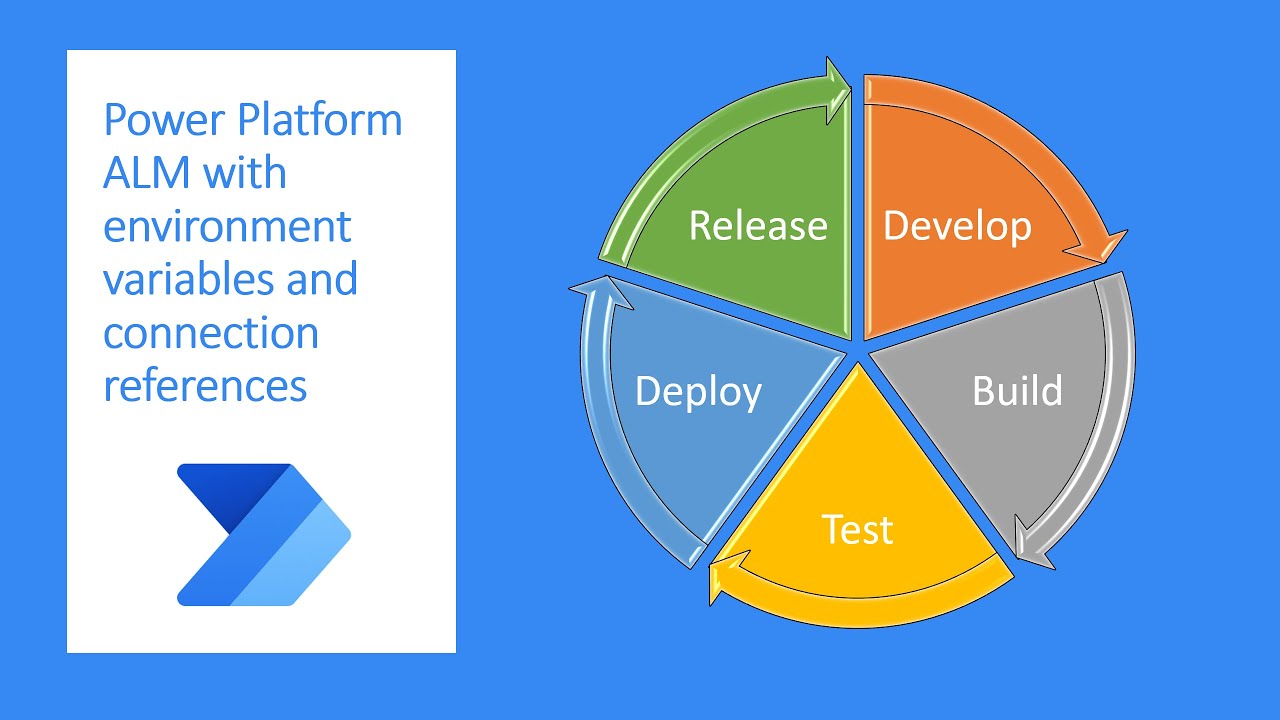
*What will happen if we don't do this project?* This type of question can help you understand the importance of the project. If users don't feel it's important, maybe we should rethink why we're spending precious resources at this time on this effort.

*What does success look like to you?*

This helps you understand the stakeholders' vision for this project. What is the most important outcome of this project for you? Consider creating a checklist of success factors and arranging them in order of importance.

*Who will benefit the most from this project?*

This will help identify key stakeholders and users. This can provide a starting point for identifying actors for high-level use cases or user stories.

End every interview by asking if there's anything else that needs to be covered. This gives the interviewee the opportunity to express other thoughts or opinions that are important to them. This almost always uncovers some valuable new item.

Request a tour of :

Visit Request Platform The ALM Platform is a flexible request management tool that can be tailored to the specific needs of your organization, projects, and processes. The platform helps you manage requirements throughout the entire development lifecycle, from gathering and analyzing requirements to tracking changes and keeping all stakeholders up to date. Not only can Visure integrate with popular Microsoft programs, but it also provides compliance templates for various international safety standards. In other words, it covers all your bases and then some.

With the ALM Request a Visit platform, you can:

Collect requirements from multiple sources and stakeholders using different

methods (e.g. interviews, workshops, existing documents)

Capture requests from Microsoft Word or Excel and other legacy tools as well as IBM DOORS via ReqIF data exchange

Analyze requirements through Visure Quality Analyzer to ensure requirements are clear, complete, consistent, traceable and actionable

Write top quality requirements in your desired format like EARS or ITEMS.

Order and prioritize requirements to ensure the right features are deployed first

Create reports and presentations to keep all stakeholders informed about the status of requirements

Track changes to requirements and automatically generate reports showing who changed what and when

Configure the tool to fit the specific needs of your organization, projects, and processes.

**Conclude**

Requirements engineering is a process of understanding and documenting the needs of a business or organization to create system requirements. The purpose of requirements gathering is to gather information about those needs from stakeholders, who are often people within the business or organization. The steps involved in requirements gathering may vary depending on the project, but typically include identifying stakeholders, defining the scope of the project, and collecting data through interviews, surveys, or other means. In this article, we've shared 6 effective requirements gathering tips that will help you get started on your next project.

P6. Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation.

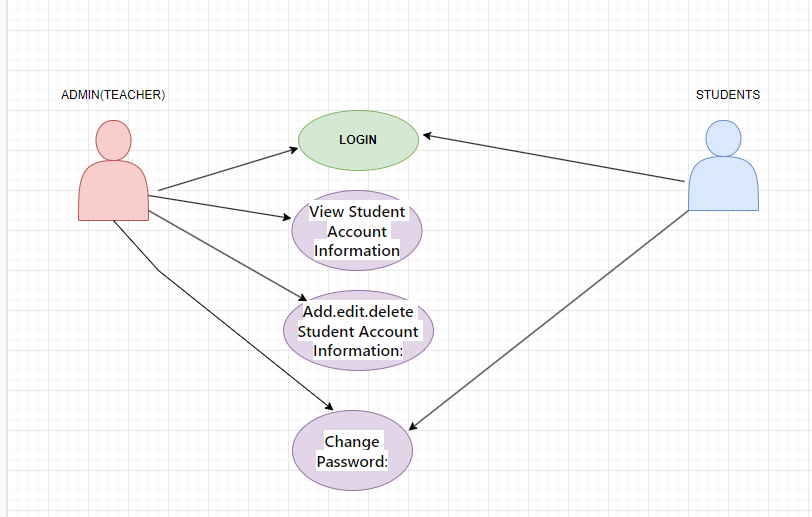
In this section, we will use structural and behavioral modeling techniques to analyze the requirements identified in Task 1. We will create supporting documents such as use case diagrams , Use Case specification, context diagram, data flow diagram and ERD data model for the entire student management system for NEI.

* 1. **Use Case diagram for the entire system.**

Use Case Diagram is a modeling tool that describes the interactions between users (actors) and

system and identify the main functions the system provides. In the student construction project

management system for NEI, the Use Case diagram will help us better understand the main features and interaction between user and system.



Fg:Use Case diagram for the entire system

In this diagram:

1. **Actors**:
   * **Admin**: Represents an administrative user who has privileges to manage student accounts.
   * **Student**: Represents the student users who have accounts and interact with the system.
2. **Use Cases**:
   * **Login**: Both Admin and Student must log in to access the system.
   * **View Student Account Information**: Admin can view detailed information about student accounts.
   * **Edit Student Account Information**: Admin can edit or update student account details.
   * **Add New Student Account**: Admin can add a new student account to the system.
   * **Delete Student Account**: Admin can delete a student account from the system.
   * **Change Password**: Both Admin and Student can change their passwords for security purposes.
3. **Associations**:
   * **Admin-Login**: Admin logs into the system to perform administrative tasks.
   * **Student-Login**: Student logs into the system to access their account information.
   * **Admin-View/Edit/Delete Account**: Admin can perform various actions related to managing student accounts.
   * **Student-Change Password**: Student can change their account password.
4. **Includes** (optional):
   * **Change Password**: Both Admin and Student use the "Change Password" use case, which is included in the main use case diagram.

Use Case Specification for 2 Use Cases.

Below is a detailed description for 3 Use Cases in NEI's student management system:

2.1**. Use Case: Sign Up**

Description

A user (who can be a student, faculty member, or staff member) wants to register a new account on the NEI student management system.

Steps

(1). The user accesses the account registration page on the user interface.

(2). Users fill in the necessary information such as name, email, address, and password in the corresponding fields on the registration form.

(3). The user clicks the "Register" button to submit registration information.

(4). The system confirms the information and creates a new account in the database.

(5). The system displays a message confirming successful registration and redirects the user to the login page.

Constraint

Passwords must have a minimum length and require a certain complexity.

Emails must be unique and properly formatted.

All information fields must be filled out.

2.2. **Use Case: Login**

Description

The user wants to log in to the NEI student management system with the registered account.

Steps

(1). The user accesses the login page on the frontend.

(2). The user enters his email and password in the corresponding fields on the login form.

(3). The user clicks the "Login" button to authenticate login information.

(4). The system checks information and authenticates users.

(5). If the login information is correct, the system allows access and redirects the user to the system's main page. Otherwise, display an error message and ask the user to try again.

Constraint

Users must provide correct login information.

The user's account must be authenticated and exist in the database.

**2.3. Use Case: Account Management**

Description

The system administrator or staff member wants to manage user accounts.

Steps

(1). Administrator or employee accesses the resource management function account on the system interface.

(2). The system displays a list of existing accounts including information such as name, email, permissions, and status.

(3). Admins or staff members can take actions such as viewing details, editing, disabling, or deleting accounts.

(4). After completing the changes, the system updates the account data in the database and displays a confirmation message.

Constraint

Only administrators or employees have permission to access and manage accounts.

Any changes to the account must be properly recorded and updated in the database.

1. Context Diagram.

In this section, I will create a context diagram to provide a clear understanding of the system's position within its broader environment. The context diagram will outline the interactions between our system and external stakeholders, helping us visualize information flows and interactions between the system and its surroundings.

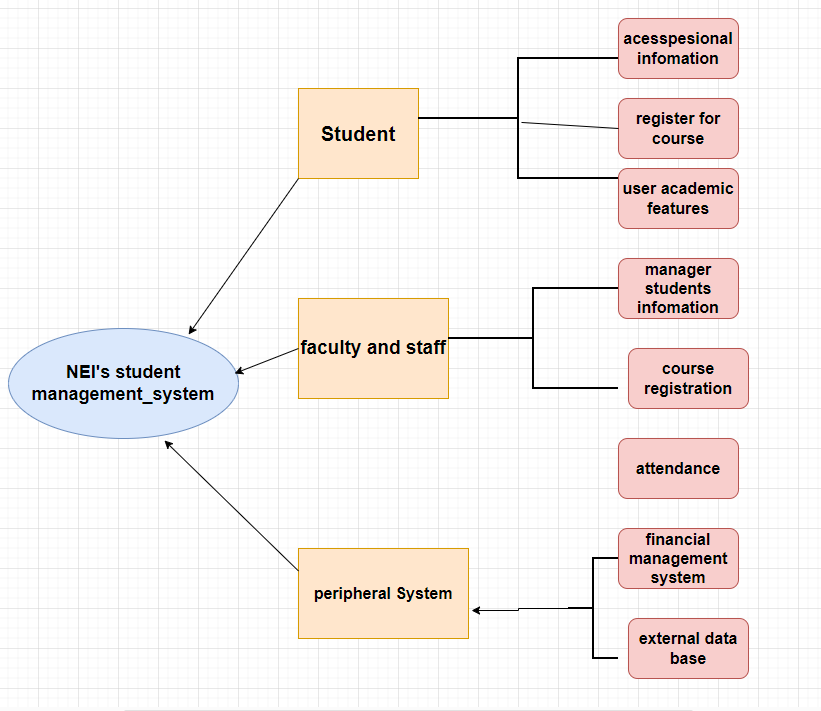
The Context Diagram for the entire system includes:

- NEI's Student Management System: This is the system under development, encompassing functionalities for student information management, course registration, attendance tracking, and access to academic resources.

- **Students**: Representing the primary users of the system, they will engage with the system to access personal information, register for courses, and utilize academic features.

- **Faculty and Staff**: Representing the group responsible for managing and providing academic services, they will interact with the system to manage student information, facilitate course registration, and oversee attendance records.

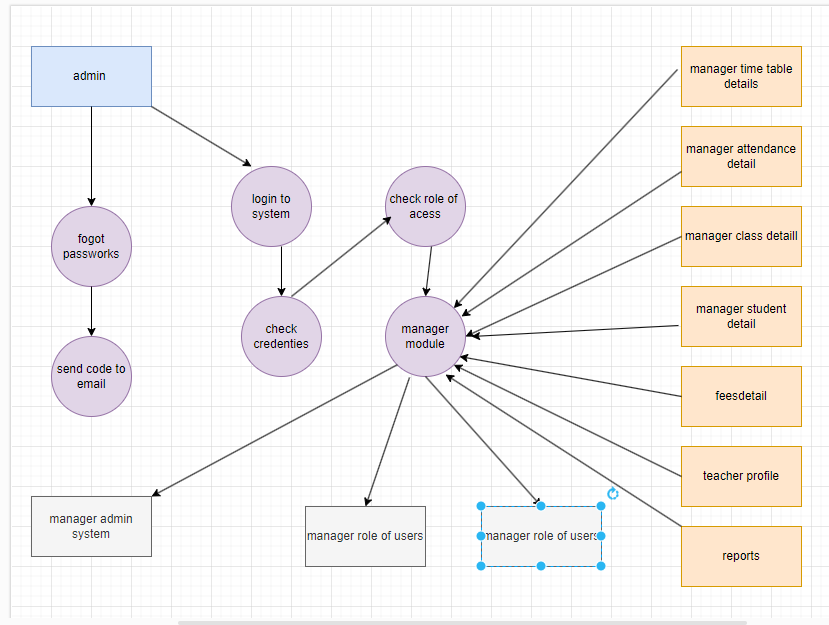
- **Peripheral systems**: Encompasses external components with which our system can interact, such as a financial management system or an external database.



Contex diagram

1. **Data Flow Diagram – Level 0 for the whole system**

A Level 0 Data Flow Diagram (DFD) provides a broad overview of the entire system, showing the main processes and data flows between them at a high level. Here's a simplified Level 0 DFD for NEI's Student Acout Management System



Data flow digram

5. ERD for the whole system.

1. Entity-Relationship Diagram (ERD) for the entire system.

To design an effective ERD for NEI's system, we first need to identify the main entities and their relationships:

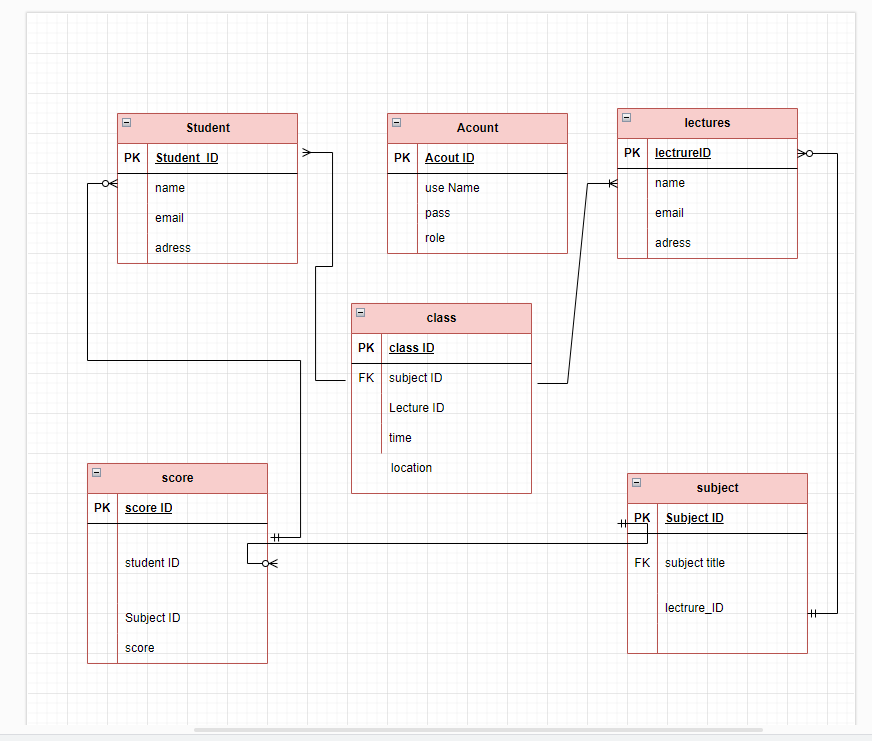
Main Entities:

1. Account
2. Student
3. Instructor
4. Subject
5. Class
6. Score

Relationships between Entities:

1. Account:
   * No direct relationship with any other table. Stores information about user accounts, including username, password, and access permissions. As it solely manages account information, it doesn't require relationships with other tables.
2. Student:
   * No direct relationship with any other table. Stores information about students, such as student ID, name, email, address, and date of birth. Each record in this table represents a single student, hence no relationships are needed with other tables.
3. Instructor:
   * No direct relationship with any other table. Stores information about instructors, including teacher code, name, email, and address. Each record in this table represents a single instructor, and thus requires no relationships with other tables.
4. Subject:
   * One-to-many relationship with the "Instructor" table via the field "lecturerID". Each subject is taught by a single instructor, but an instructor can teach many subjects.
5. Class:
   * One-to-many relationship with the "Subject" table via the field "subjectID". Each class is related to a specific subject.
6. Score:
   * One-to-many relationship with the "Student" table through the field "StudentID". Each record in this table represents a student's score in a specific subject.

The ERD will visually represent these entities and their relationships, providing a clear understanding of how data is structured and interconnected within the NEI's system.



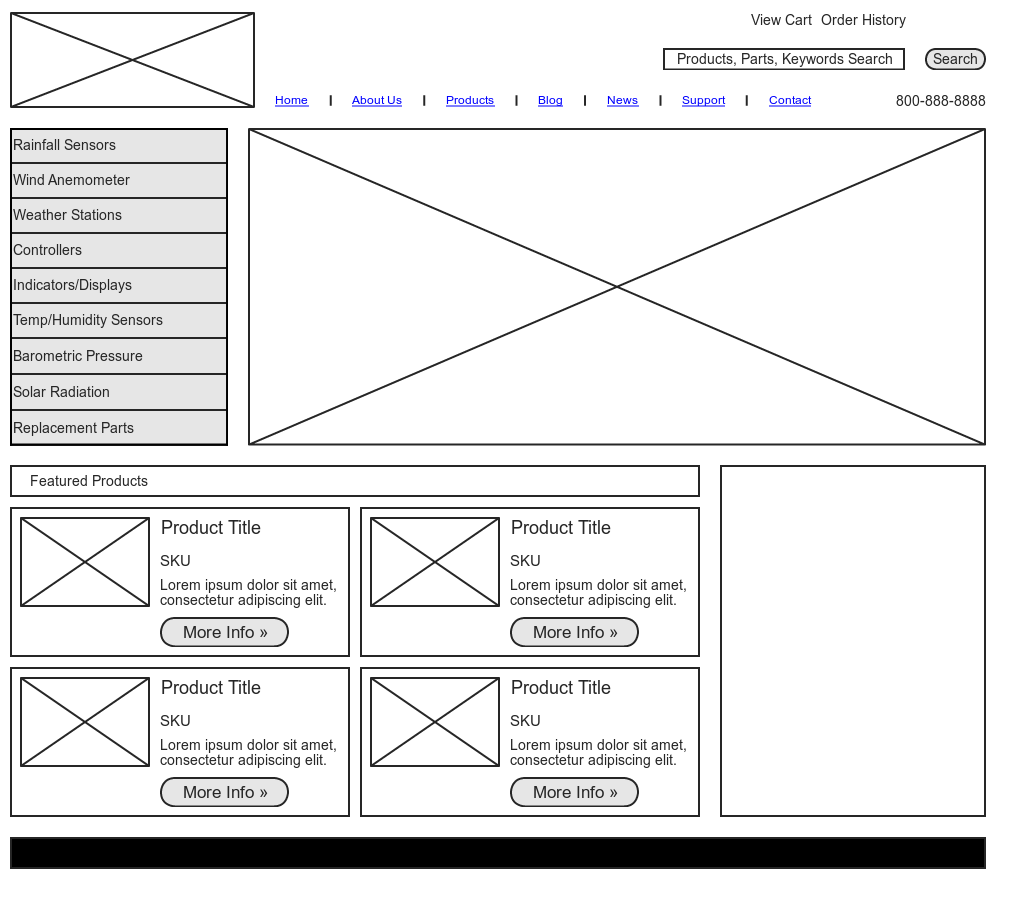
Erd diagram

P7. Discuss how to proceed with the design and development phaseof anNEIproject.

1. **and Wireframes**:
   * **Mock-ups**: Mock-ups are static images or diagrams designed to visually describe how the user interface will look after completion. The main purpose of a mock-up is to create a clear and comprehensive image of the interface, including the positioning of elements, page structure, colors, and graphics. Mock-ups are typically created using graphic design tools or UI design software such as Adobe Photoshop, Sketch, or Figma.

****

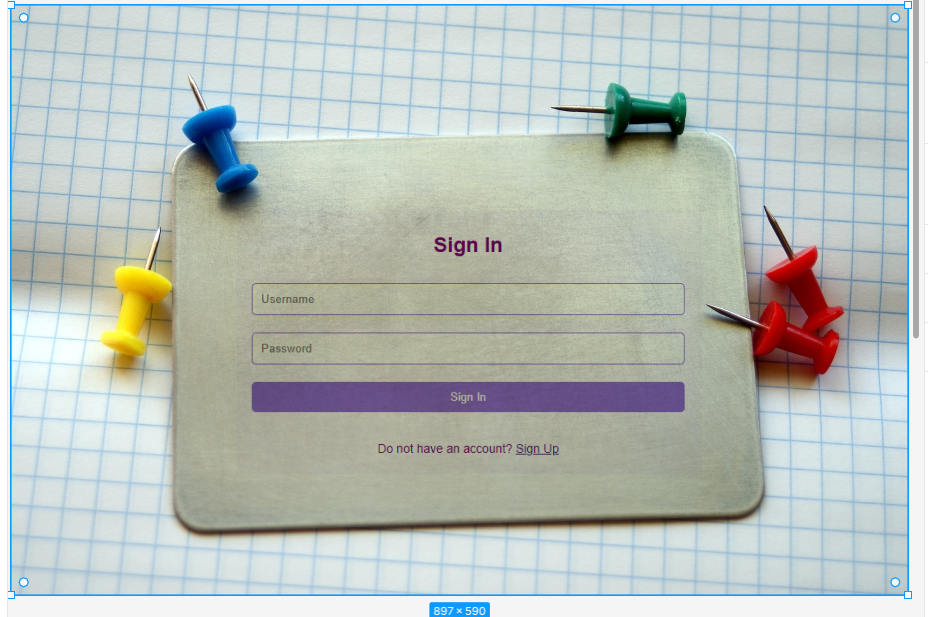
* + **Wireframes**: Wireframes are simple documents or images used to represent the overall structure and layout of the user interface. The main goal of a wireframe is to focus on determining basic elements such as frames, squares, and the placement of elements on the page, without concern for details about color or graphics. Wireframes are often quickly created using graphic design tools or design software like Adobe XD, Balsamiq, or Axure.



**Were frame**

1. **Use Figma to design the interface of your NEI project.**

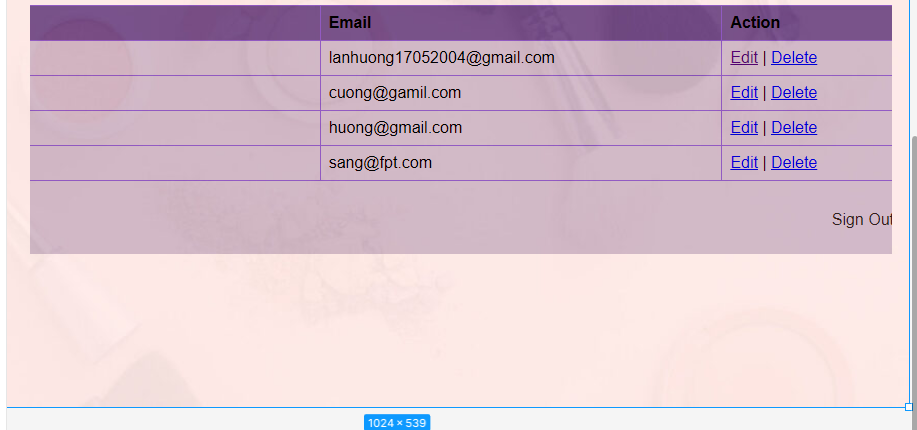
Utilize Figma for crafting the interface of your NEI project. The interfaces sculpted via Figma are meticulously tailored to meet user demands comprehensively. With a focus on simplicity and user-friendliness, these interfaces encompass vital functionalities such as registration, login, student management, and personal information viewing. Moreover, they are meticulously crafted to adapt seamlessly across various devices and screen dimensions, ensuring a uniform experience across all platforms. Adhering to interface design principles and crafting intuitive, easy-to-navigate interfaces, they facilitate users in engaging with the system in the most efficient and intuitive manner possible. Presented below are the 5 interface designs I've meticulously curated using Figmar.



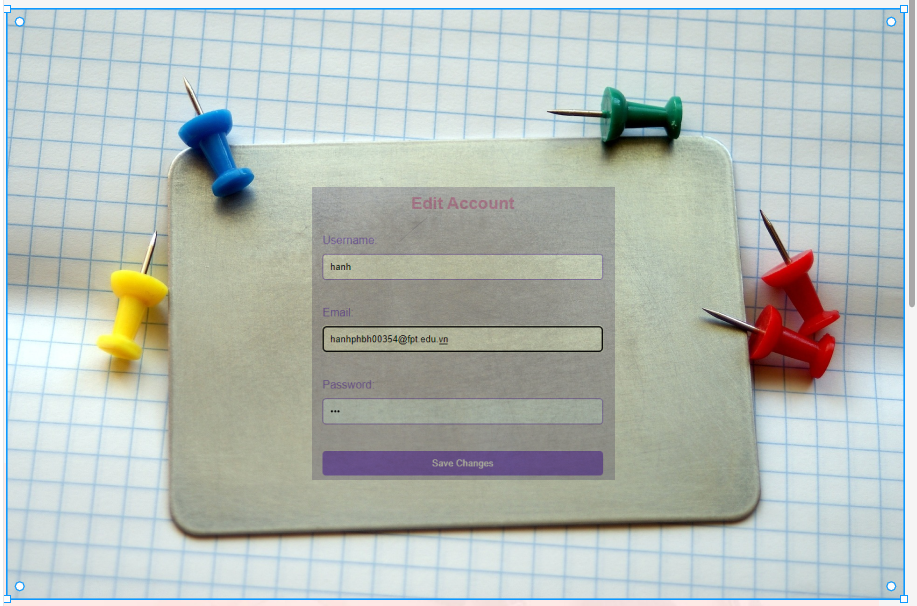
*Use Figma to design the interface form signin*

**

*Use Figma to design the interface form signup*

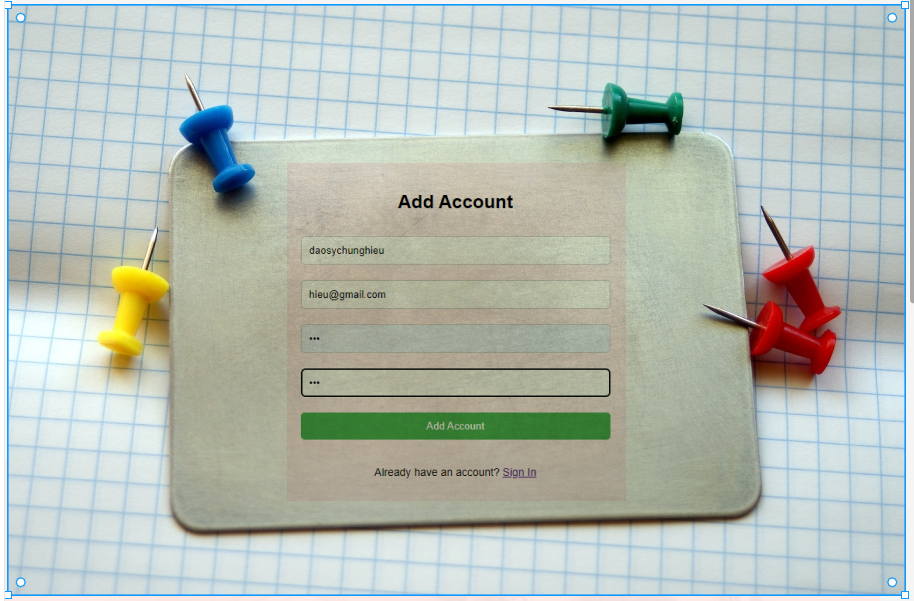


Use Figma to design the interface form Manage User Account



Use Figma to design the interface form Edit Account

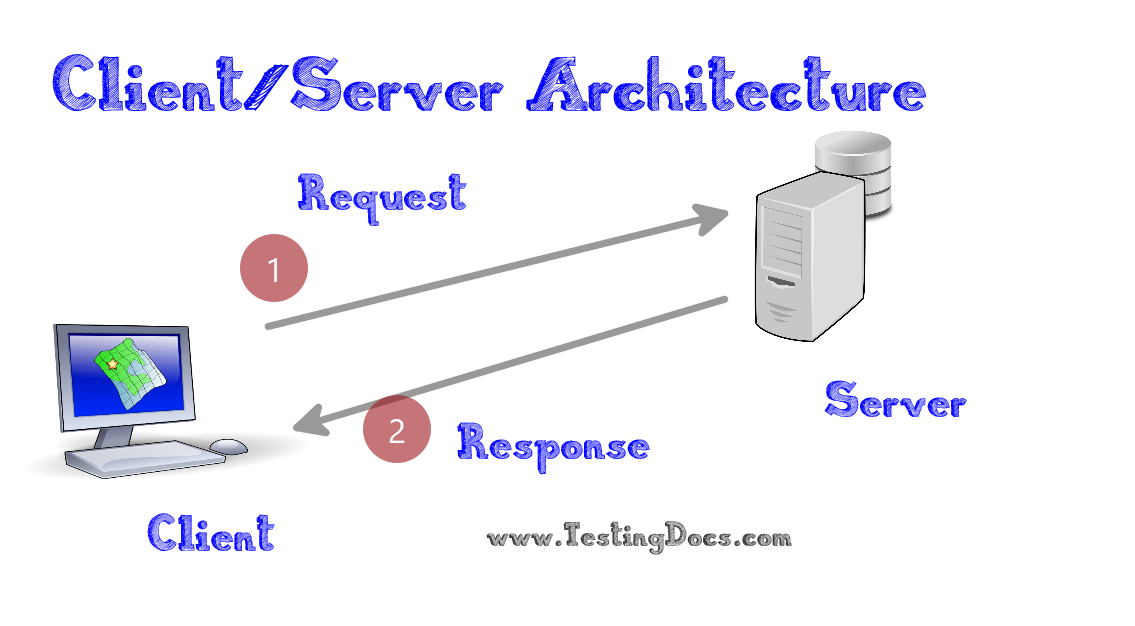
Use Figma to design the interface form edit Account



Use Figma to design the interface form edit Account

**3. Choose the appropriate architecture.**

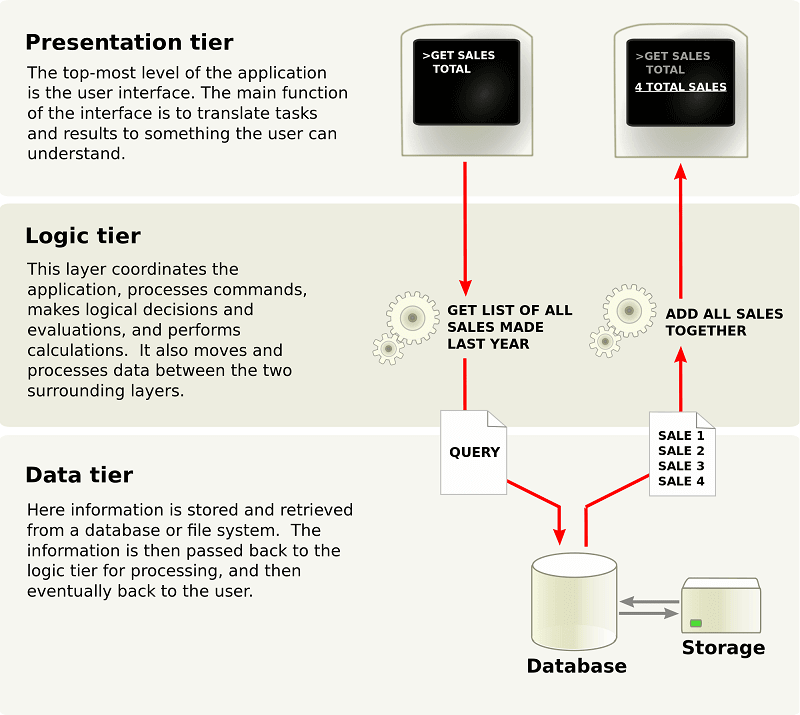
3.1. Popular architectures. Client-Server architecture In this architecture, the system is divided into two main parts: client and server. The client isresponsiblefor sending requests and receiving results from the server. This architecture is suitable for traditional webapplications, where the client loads web pages and communicates with the server to retrievedataorperformoperations.



*Choose the appro ariatetecture*

**N-tier architecture**

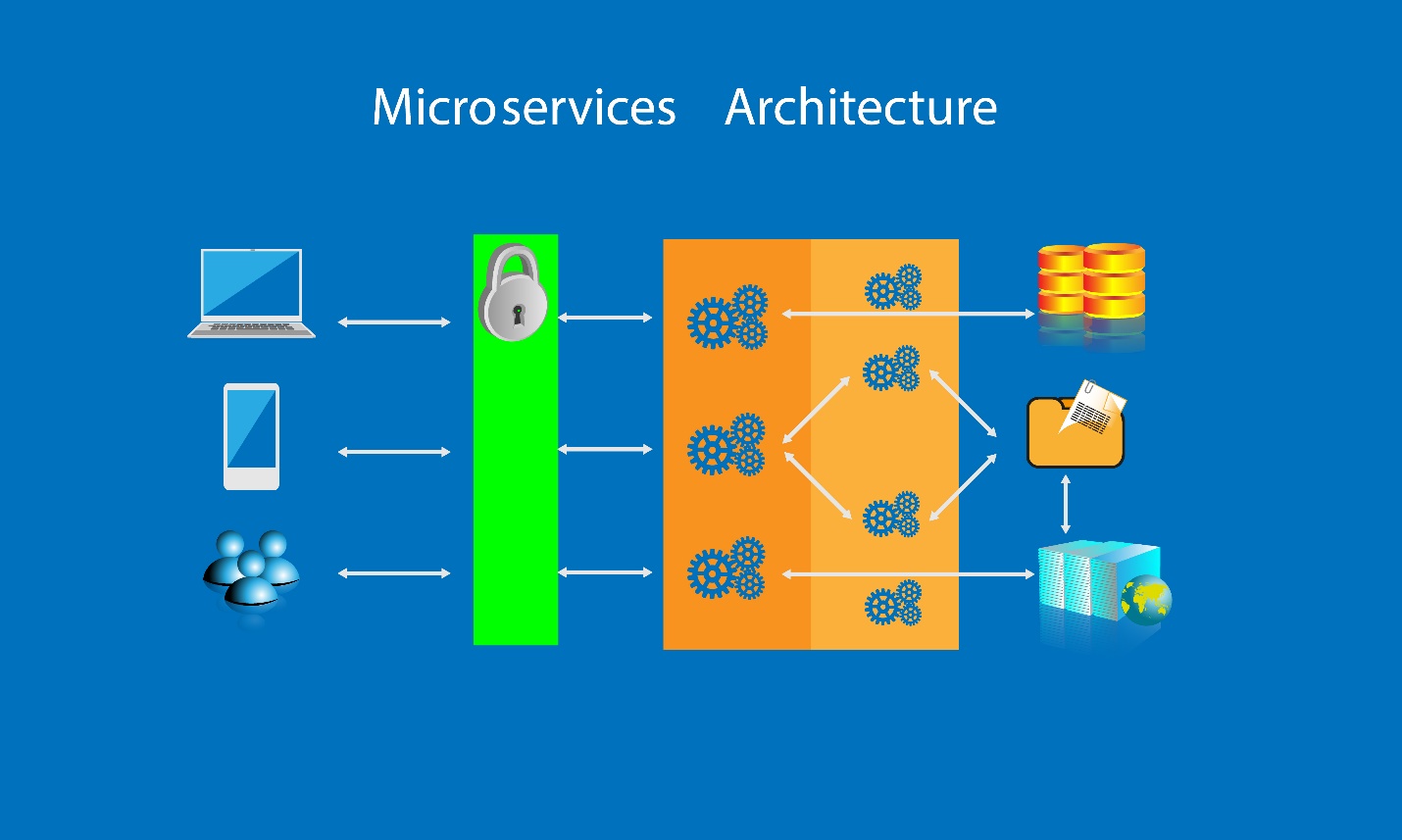
This architecture divides the system into many different layers, usually including the user interfacelayer,application logic layer and data layer. Each layer has a specific task and interacts with other layersthroughpredefined communications. This architecture is often used in complex enterprise applications.



*Ntier architecture*

***Microservices Architecture***

In this architecture, the system is divided into independent microservices, each responsibleforaspecificfeature. These services communicate with each other through lightweight protocols such as HTTPormessagequeues. Microservices architecture helps increase system modularity, flexibility, and scalability



*Micro sevices architecture*

**3.2. Determine the appropriate architecture for the project. Based on the overall analysis of the NEI** project and specific requirements, we decidedtochoosetheMicroservices architecture as the most suitable for the project. The main reason is because of thecomplexityand diversity of the project, as well as the need for flexibility, scalability and high performance. Microservicesarchitecture allows dividing the system into small, independent services, facilitating development, deploymentand management. At the same time, this architecture provides flexible scalability and ease of management,helping to optimize system performance and stability. Applying Microservices architecturewill bringmanybenefits to the NEI project, from development to system management and expansion.

**4. Technical solutions deployed.**

4.1. Specific technical solutions.

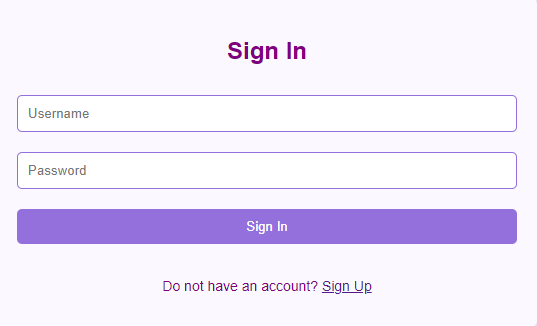
To implement the NEI project, we propose to use the PHP programming language combinedwiththeMySQL database. PHP is a powerful, flexible and popular programming language for webapplicationdevelopment. It supports many libraries and frameworks that make application development andmaintenancemore convenient. The combination of PHP and MySQL creates a stable and powerful development environment,helping to build and manage databases effectively. *Using PHP* and MySQL also helps reducesystemdevelopment and maintenance costs, while also facilitating future integration and expansion.



Php and my sql

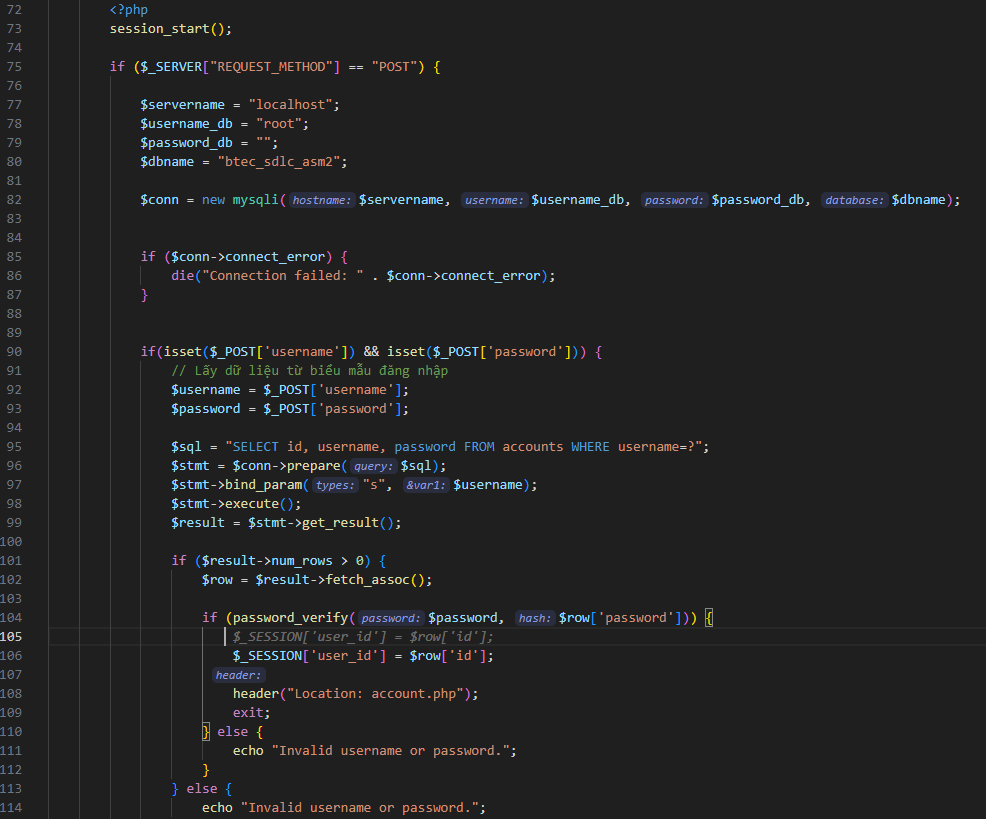
*4.2.* **Implement specific use cases. Use case signin**

The user will enter his sigin information (email and password) and the systemwill thencheckthisinformation against the database to authenticate the user. Using PHP, we will make queries totheMySQLdatabase to check the credentials and allow access to the system if the credentials are valid.

**

*Figure 22 Use case sigin in*

Use PHP to check the login information from the user and compare it with the dataintheMySQLdatabase. Because PHP provides authentication and session management functions, it helps identifyusersandsecurely check credentials. MySQL stores student account information and supports security mechanismssuchas password encryption.



In the PHP code provided above, we've implemented a straightforward login mechanism that leverages a MySQL database and the password\_verify() function for password validation. To maintain flexibility, we retrieve the login credentials from the data submission form using the variables $\_POST['username'] and $\_POST['password']. Employing parameterized SQL statements helps safeguard against SQL Injection attacks.

The password\_verify() function plays a crucial role in verifying whether the entered password matches the hashed password stored in the database. This ensures that passwords are never stored in clear text, thereby enhancing user authentication flexibility.

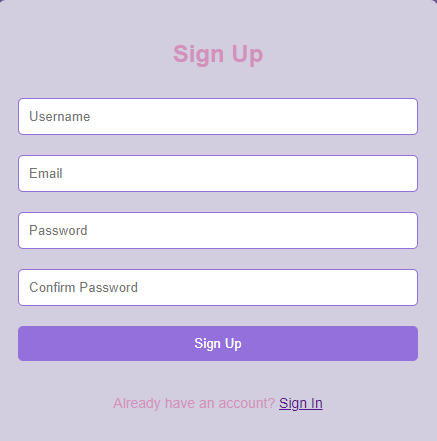
For enhanced performance, we utilize parameterized SQL statements and query prepare statements, which not only optimize database performance by thwarting SQL Injection attacks but also minimize processing time.

Moreover, the password\_verify() function allows for a streamlined comparison process, only requiring a comparison of hashed passwords rather than decrypting and comparing the entire passwords. This significantly reduces server load and boosts performance.

In terms of security, the password\_verify() function ensures that passwords remain undisclosed during storage and transmission over the network. Furthermore, employing sessions to store user\_id upon successful login, rather than storing login information in cookies, enhances security.

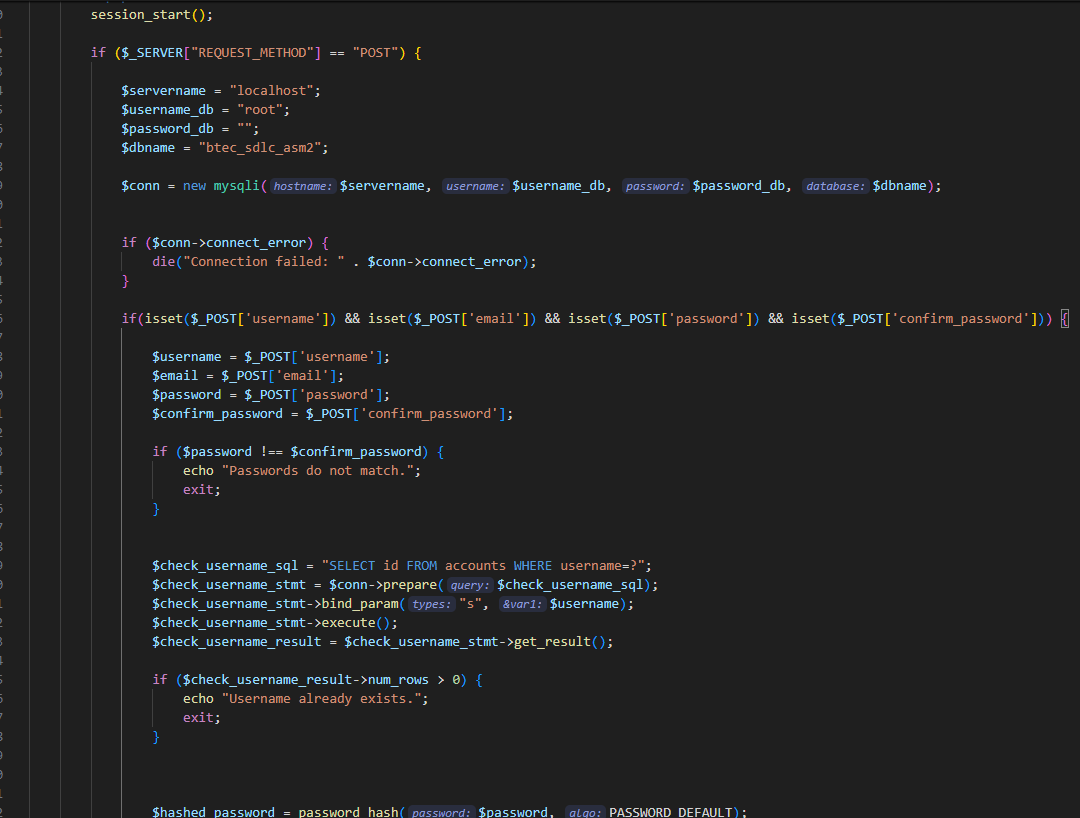
**Regarding the SignUp use case:**

users will input necessary information into the registration form, which will then be stored in the database by the system. PHP will handle the data submitted from the form and append new user information to the MySQL database.



*Source signup*

Utilize PHP to handle input from registration forms and establish new student accounts within a MySQL database. PHP stands as a prevalent programming language adept at managing form input and executing registration procedures. Meanwhile, MySQL furnishes security and scalability, ensuring the safe and efficient storage of student data.



*Soure: signin*

**The provided PHP code exemplifies a streamlined user registration process, integrating a MySQL database and the password\_hash() function for enhanced password security. Here's a refined overview of the key features:**

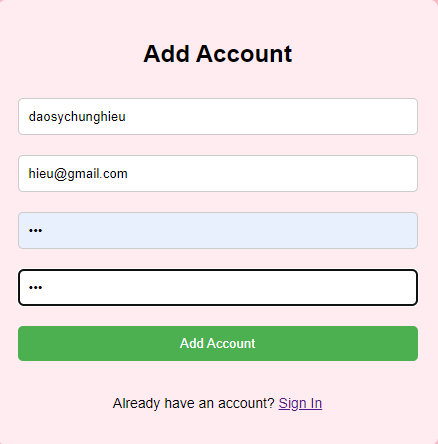
1. Password Security: Before storing user passwords in the database, they undergo hashing using the password\_hash() function. This crucial step fortifies password security by encrypting them into irreversible hash strings, bolstering defense against potential attacks.

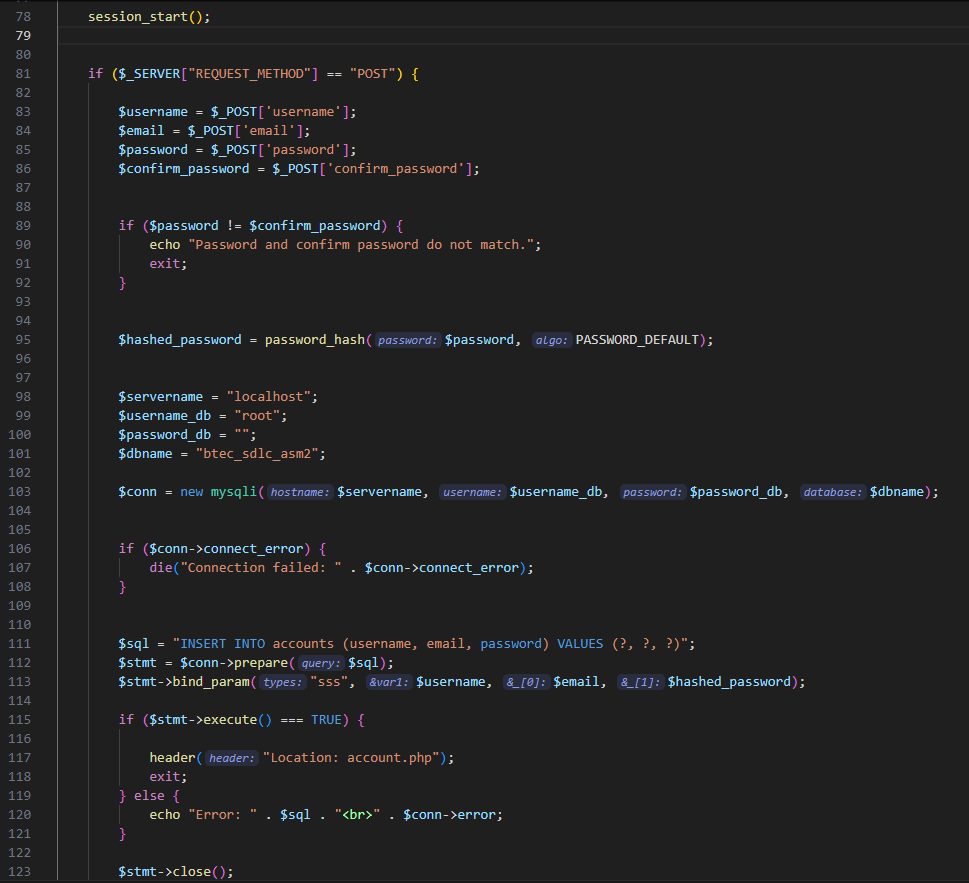
2. Prepared Statements Usage: To safeguard against SQL injection attacks, prepared statements are employed for inserting registration data into the database. By sidestepping direct insertion of user data into SQL statements without proper validation, the system maintains robust security measures.

3. Error Handling: In the event of SQL statement execution errors, clear and concise error messages are promptly provided to users. This proactive approach enhances user experience and ensures system flexibility by furnishing comprehensive error information for effective troubleshooting.

4. Redirecting after Successful Registration: Upon successful registration, users are seamlessly redirected to the login page. This intuitive redirection not only enhances system flexibility but also contributes to a user-friendly and convenient experience for all users.

Regarding the "Add Account Students" use case, administrators are empowered to incorporate new student accounts into the system through a user-friendly student addition form. Utilizing PHP, the system adeptly processes submitted form data, facilitating the seamless integration of new student entries into the MySQL database.



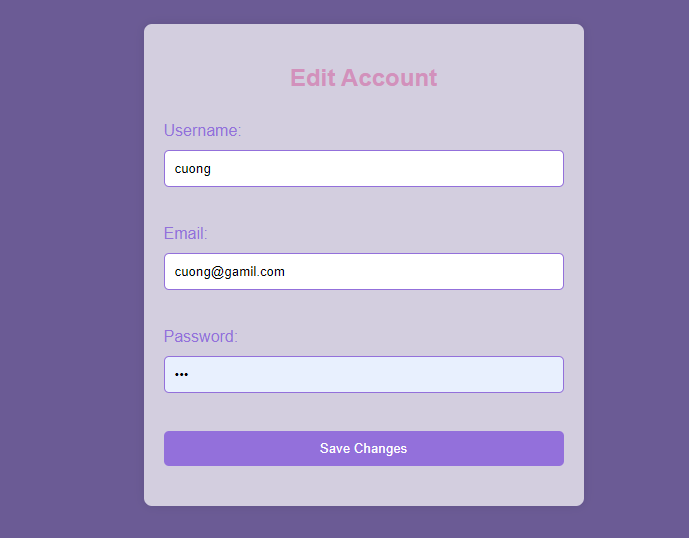


*Source code : add acount*

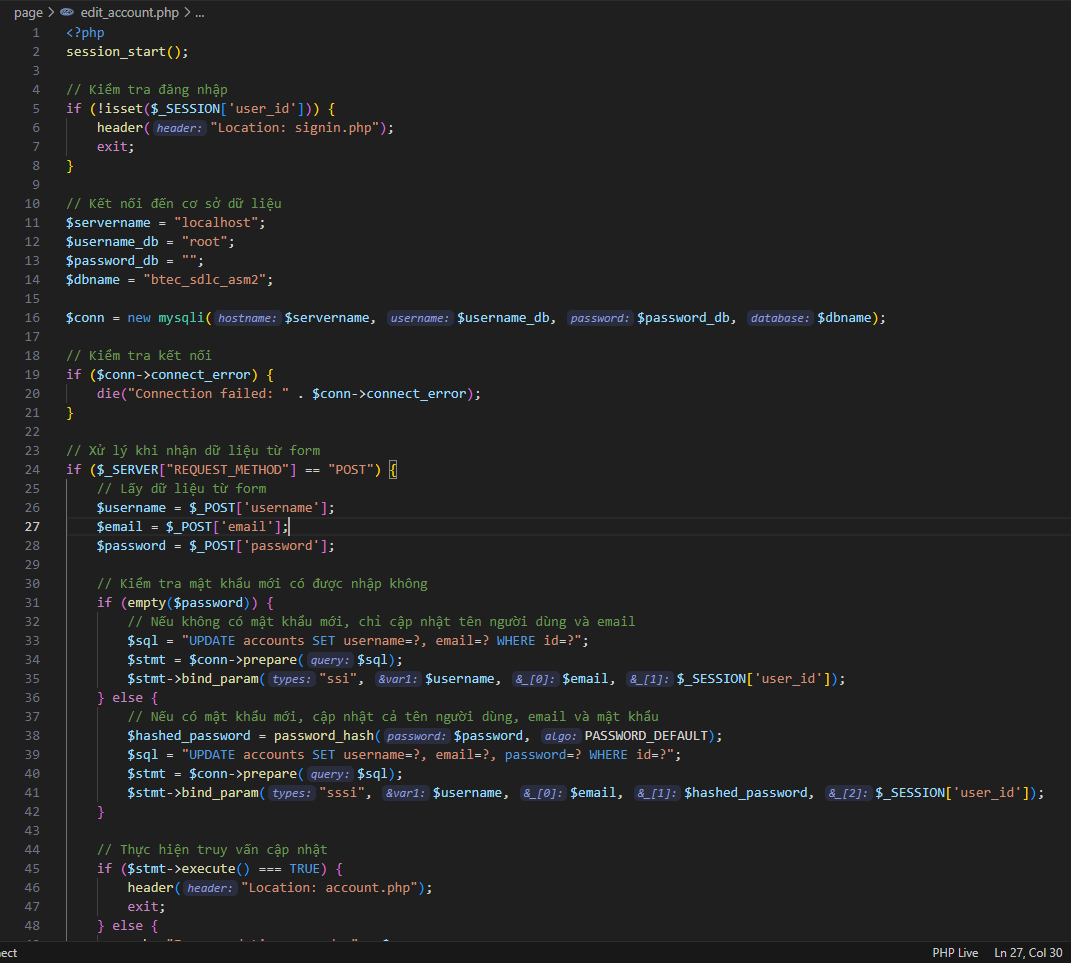
The provided PHP code implements a user registration system with enhanced security measures. Here's a refined explanation of its key features:

1. **Password Validation**: The script first ensures that the password and confirm password fields match, ensuring data accuracy and preventing input errors.
2. **Password Security**: Following the validation, the password is securely hashed using the password\_hash() function. This crucial step encrypts the password into an irreversible hash string, fortifying its security against potential attacks.
3. **Prepared Statements Usage**: Prepared statements are utilized to insert registration data into the database. This practice mitigates the risk of SQL injection attacks by avoiding direct insertion of user data into SQL statements without proper validation.
4. **Error Handling and Connection Closure**: In the event of errors during SQL statement execution or database connection, clear error messages are presented to the user. Additionally, upon successful operation, the database connection is promptly closed, and the user interface is updated, either by reloading the page or closing the window.

This PHP code exemplifies a robust and secure approach to user registration, emphasizing password security, data accuracy, and prevention of SQL injection attacks. It prioritizes user experience by providing clear error messages and ensuring proper closure of database connections after completing operations.



*Edit acount student*



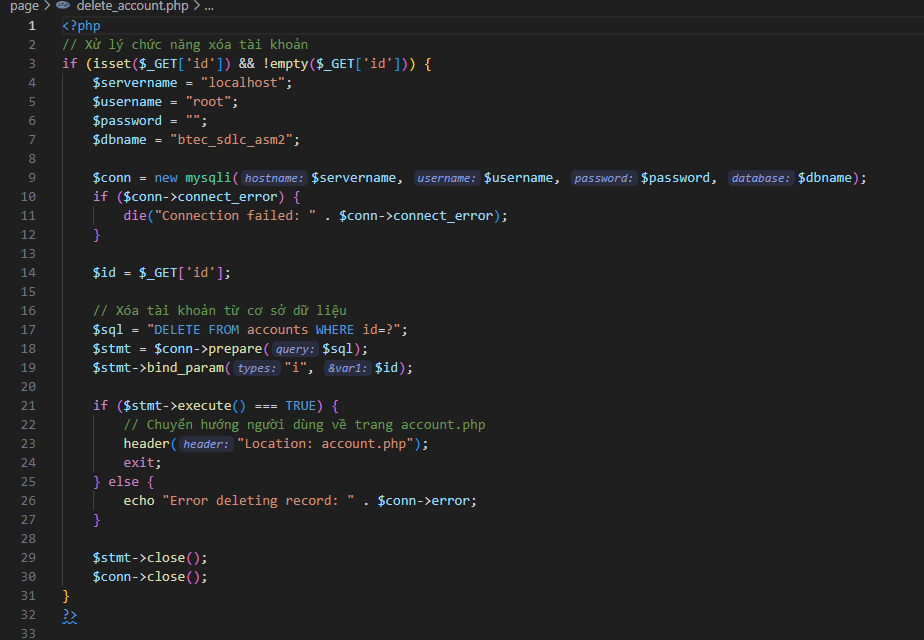
*Source code edit acount*

The provided PHP code outlines a systematic flow for users to update their account information. Here's a refined explanation for better clarity:

1. **Login Verification**: Initially, the code verifies whether the user is logged in by checking the existence of the session variable $\_SESSION['user\_id']. If the user is not logged in, they are redirected to the login page, ensuring security and access control.
2. **Database Connection**: Following login verification, the code establishes a connection to the MySQL database to facilitate querying or updating account information, ensuring seamless interaction with the database.
3. **Handling User Credentials**: Upon determining the request method, the code retrieves user information from the $\_POST variable if the method is POST. Subsequently, it proceeds with necessary checks and operations.
4. **Updating Account Information**: The code executes an SQL UPDATE statement to modify the user's account details in the database. If a new password is provided, it is securely hashed using the password\_hash() function before being updated in the database, ensuring password security.
5. **Redirection and Error Handling**: Upon successful execution of the SQL statement, the user is redirected to their account page. Conversely, if an error occurs, a specific error message is displayed, enhancing user experience and facilitating troubleshooting.
6. **Querying Current Account Information**: Before displaying the edit form, the code retrieves the user's current account information from the database. This information is then used to pre-fill the edit form, simplifying the user's account update process.

**Delete account students:**

Admin can delete student accounts from the system. Once the deletion request is submitted, weusePHP to perform a query to delete the data from the MySQL database.



*Soure code delete acount*

*Here's a refined explanation of the provided content for better clarity:*

**Establishing Database Connection**: Initially, the code initiates a connection to the MySQL database by providing the necessary login credentials such as servername, username, and password. This step ensures seamless interaction with the database.

**Handling Delete Request: Subsequently,** the code retrieves the ID of the record to be deleted from the $\_REQUEST['id'] variable. Typically, this ID is passed as a parameter through the URL, facilitating targeted deletion.

**Executing SQL DELETE Statement**: The code constructs an SQL DELETE statement to remove records from the accounts table based on the provided ID. This statement effectively deletes the specified record from the database.

**Result Handling**: Upon execution of the SQL statement, the code checks if it was successful (line $conn->query($sql) === TRUE). If successful, the code closes the database connection and redirects the user to the main home page. Conversely, in the event of an error, a specific error message is displayed to the user, ensuring transparency and facilitating troubleshooting.

This succinct explanation highlights the key steps involved in handling delete requests from the database, ensuring efficient and secure data management within the application.

**VI. Conclusion**

This document provides a comprehensive exploration of the software research process tailored to meet the specific business needs of the NEI project. It delves into various crucial aspects, including requirements analysis, software quality management, and the design and development phases. Throughout the document, we meticulously identify stakeholders and their requirements, discussing diverse techniques to fulfill these needs while emphasizing the importance of tracking requirements throughout the software lifecycle.

Furthermore, we delve into the intricacies of software quality management, delineating pertinent quality attributes and elucidating quality assurance techniques. Additionally, we outline a structured approach to navigating the design and development phases, incorporating methodologies such as mock-ups, wireframes, and interface design using Figma. We also address the selection of suitable architectures and technical solutions to ensure robust development.

Moreover, the document explores advanced software behavioral tools and techniques, including Finite State Machines (FSM) and Extended State Machine Language (ESML), elucidating their applications and distinctions. By differentiating between FSM and extended FSM, we offer valuable insights into their respective uses.

In essence, this document serves as a comprehensive guide for stakeholders involved in the NEI project, equipping them with the necessary knowledge and strategies to effectively meet business requirements, maintain software quality, and navigate the design and development phases with efficiency and proficiency.

**VII. Reference.**

GeeksforGeeks. (2020). *Traceability and its types*. [online] Available at: <https://www.geeksforgeeks.org/traceability-and-its-types/>

Iggy (2014). *40+ Free Mockup Templates to Present Your UI Designs*. [online] [Line25.com. Available at: https://line25.com/articles/40-free-mockup-templates-to-present-your-ui-designs/ [Accessed 6 Apr. 2024]](file:///C:\Users\huong\Desktop\ASM2%20PHP%20THẦY%20ĐÔNG\Line25.com.%20Available%20at:%20https:\line25.com\articles\40-free-mockup-templates-to-present-your-ui-designs\%20%5bAccessed%206%20Apr.%202024%5d)

www.visme.co. (2022). *Free Mockup Templates - Customize & Download | Visme*. [online] [Available at: https://www.visme.co/templates/mockups/ [Accessed 6 Apr. 2024]](file:///C:\Users\huong\Desktop\ASM2%20PHP%20THẦY%20ĐÔNG\Available%20at:%20https:\www.visme.co\templates\mockups\%20%5bAccessed%206%20Apr.%202024%5d)

‌

‌

‌