Postgraduate degree in Computer Science

Course work

'OPERATIONAL RESOURCE MANAGEMENT (ERP) SOFTWARE SYSTEMS'.

Work of

Kormazos Evangelos

(A.M.:ΜΠΛΛ21036)

Supervisors:
Konstantinos Metaxiotis
Konstantinos Liangouras

Athens, November 2022

Introduction

The purpose of this work is to create our own ERP, for a Secondary Education Tutoring Center, this is done in the context of the course "Enterprise Resource Management Software Systems (Erp)" for the Master's degree in Computer Science at the University of Piraeus. At first, we will mention some information about ERPs and in the following chapters we will analyze the requirements of our ERP and the detailed design of the System. At the end of this paper we hope that the reader and ourselves will have a better understanding of the concept of ERPs and the difficulties we encounter in their construction.

Contents

Chapter 1:About Enterprise Resource Management Systems

1.1 What are Enterprise Resource Planning Systems (ERP)?

ERP stands for Enterprise Resource Planning. It is a type of software system used by organizations to manage their business processes and operations. ERP software integrates various functions of a company, such as accounting, procurement, inventory management, production planning, human resources and customer relationship management, into a single system.

The main purpose of ERP is to provide a centralized view of business processes and data across all departments and functions in real time. It helps to streamline workflows, optimize resources, improve communication and make informed decisions based on accurate and up-to-date information. ERP systems are commonly used by medium and large sized organisations and can be customised to meet specific business needs. It is a type of software system used by organisations to manage their business processes and operations. ERP software integrates various functions of a company, such as accounting, procurement, inventory management, production planning, human resources and customer relationship management, into a single system.

The main purpose of ERP is to provide an aggregated view of business processes and data across all departments and functions in real time. It helps to streamline workflows, optimize resources, improve communication and make informed decisions based on accurate and up-to-date information. ERP systems are commonly used by medium and large sized organisations and can be customised to meet specific business needs.

1.2 Implementation of an ERP system in a Secondary School Benefits, requirements and challenges.

The implementation of an ERP system in a Secondary School can bring many benefits to the organization, such as improved management of student information, course schedules, financial data and administrative processes. However, it is important to carefully consider the requirements and needs of the university, as well as the potential challenges and risks associated with implementing an ERP system.

The implementation process should begin with a thorough analysis of the university's existing processes and systems to identify the areas where an ERP system could provide the most significant benefits. This may involve collecting input from various stakeholders such as faculty members, administrators and students. Once the requirements are identified, the tutorial institution can begin to evaluate various ERP solutions to determine which is best suited to its needs. This may include looking at factors such as cost, functionality, ease of use and the ability to integrate with existing systems.

Once an ERP system has been selected, the implementation process can begin. This usually involves a project team consisting of internal staff and external consultants who will work together to configure the system, input data and test the system to ensure it meets the university's requirements. Staff and faculty members should be trained in how to effectively use the new system and ongoing support should be provided to ensure that the system continues to function optimally. Overall, the implementation of an ERP system at a Secondary Education Tutoring Center can help streamline administrative processes, improve data management, and improve the overall efficiency and effectiveness of the organization.

1.3 What they offer to a business.

There are many advantages of using an ERP system in a business. Some of the major advantages include:

- Streamlined business processes: ERP systems automate and streamline business processes, eliminating redundancies and reducing the risk of errors. This has the effect of resulting in improved efficiency and productivity.
- Improved decision making: ERP systems provide data and real-time analytics, allowing businesses to make informed decisions about operations, finance and other critical areas.
- Improved visibility: ERP systems provide an aggregated view of business processes, allowing stakeholders to access to accurate, real-time data on operations, finance and supply chain management.
- Increased collaboration. collaboration and effective communication between different departments and teams, leading to better decision making and problem solving.
- Cost reduction: By automating and streamlining the business processes, ERP systems help reduce costs associated with manual data entry, inventory management and other tasks.
- Improved customer service: ERP systems provide accurate and up-to-date information, which helps businesses to respond quickly to customer queries and provide better customer service.
- Compliance: ERP systems help businesses to comply with various regulatory requirements and standards, such as SOX and GDPR, providing controls and procedures to ensure compliance.
- Interconnectivity: ERP systems help businesses to interconnect with other companies using the same information system
- Reduction of time: They contribute to the reduction of the execution time of the work or even its elimination.
- Consolidate the different PS of the company: The consolidation of the different A company's management accounts is a critical step in obtaining a comprehensive picture of its financial performance. MAs are a key tool for companies to monitor and analyze their financial operations, including budgeting, forecasting and decision-making.

Overall, an ERP system offers an integrated solution that helps the businesses to improve efficiency, reduce costs and make better informed decisions, ultimately leading to improved profitability and growth.

1.4 Problems and possible disadvantages of implementing an ERP system.

While there are many advantages to implementing an ERP system, there are also some potential disadvantages that organisations need to consider:

- Cost: Costs include software licenses, hardware, consultancy and training.
- Long payback period (RoI):The long payback period (RoI), or Return on Investment, is another potential drawback of ERP systems. Implementing an ERP system can be a significant investment, both in time and money, and it may take some time before the organization realizes the return on this investment.
- Complexity: ERP systems can be complex and time-consuming to implement, requiring significant planning and coordination between different departments and functions.
- Adjustment: Customizing an ERP system to meet specific business needs can be difficult and costly, requiring significant development efforts and ongoing maintenance.
- Change management: Implementing an ERP system may require significant changes to business processes, which can cause disruption and challenges for employees.
- Data security: ERP systems store sensitive business data, which can be vulnerable to cyber-attacks and other security threats.
- Lock in supplier: Transitioning to a new ERP system can be difficult and costly, as businesses may be locked into contracts with the ERP vendor.
- Adoption by users: ERP systems require employees to learn new processes and software, which can be difficult and affect user adoption.
- Loss of flexibility: Changing an ERP system is time-consuming and costly, making it quite difficult to change them.
- Time-consuming product installation and development: Implementing an ERP system can be a complex process that requires significant time and resources to customize and integrate the software with existing business processes and data.
- Long-term and accurate application: The ERP system should be designed to provide reliable and accurate information to the organization in the long term.

It is important to note that these disadvantages do not apply to all ERP systems and many of them can be mitigated with careful planning, effective change management and ongoing support and training. It is important for organizations to evaluate the potential benefits and risks of implementing an ERP system before making a decision.

1.5 The most well-known ERP.

There are many ERP systems available in the global market and their popularity can vary depending on the industry, region and specific business needs. However, some of the most popular ERP systems used by organizations worldwide are:

- SAP: SAP is one of the world's largest ERP software providers, offering a wide range of solutions for various industries.
- Oracle ERP: Oracle offers a suite of ERP solutions that cater to businesses of all sizes, from small and medium enterprises to large corporations.
- Microsoft Dynamics 365: Microsoft Dynamics 365 is a cloud-based ERP solution that integrates various business functions and offers scalability and flexibility.
- Infor ERP: Infor offers a range of ERP solutions designed to meet the needs of various industries such as manufacturing, healthcare and hospitality.
- NetSuite ERP: NetSuite ERP is a cloud-based solution that integrates various business functions, including financial management, inventory management and order management.

These are just a few of the most well-known ERP systems used by organizations worldwide. The choice of ERP system ultimately depends on a company's specific requirements and budget.

While many of the above companies are active in the Greek market, we may be interested in finding a Greek ERP company that understands the specific needs and challenges of businesses operating in Greece. After our research we can recommend some Greek ERP companies:

- Softone Technologies:SoftOne Technologies S.A. was founded in 2002 and is active in the development of ERP systems, CRM, cloud services, enterprise mobility applications and e-invoicing solutions, having received numerous awards and distinctions for its innovation and market leadership.
- Entersoft : Entersoft is an innovative IT company specializing in software production and business services.

- Unisoft: UNISOFT has enterprise resource planning systems, designed to meet the business needs of all small and medium enterprises.
- Danaos Corporation: Danaos offers a range of ERP solutions designed to meet the needs of a company in the Shipping Sector.

1.6 Different types of ERP

There are several types of ERP systems available and they can be broadly categorized based on their functionality and development models. The following are some of the most common types of ERP systems:

- On-premise ERP: On-premise ERP systems are installed and hosted locally on the company's own servers and infrastructure. They are typically customized to meet specific business requirements and are managed and maintained by the company's IT team.
- Cloud-based ERP: Cloud-based ERP systems are hosted in the cloud and accessed via the internet. They offer flexibility and scalability, as businesses can easily add or remove users and features as needed. Cloud-based ERP systems are typically subscription-based and require minimal hardware and maintenance costs.
- Open source ERP: Open source ERP systems provide access to the source code, allowing developers to adapt and modify the software to meet specific business needs. They are usually free to use and offer a cost-effective solution for small and medium-sized businesses.
- Industry-specific ERP: Industry-specific ERP systems are designed to meet the unique needs of specific industries, such as manufacturing, healthcare or retail. They offer industry-specific features and functionality and are tailored to meet industry regulatory and compliance requirements.
- ERP for small businesses: ERP systems for small businesses are designed to meet the needs of small businesses and start-ups. They offer key features and functionality, such as accounting, inventory management and customer relationship management, at an affordable cost.
- Hybrid ERP: Hybrid ERP systems combine the functionality of multiple ERP systems, such as on-premise and cloud-based, to provide a more comprehensive solution that meets the unique needs of a business. They can offer the advantages of both systems, such as the customization of on-premise and the flexibility of cloud-based ERP.

These are just a few examples of the types of ERP systems available and businesses can choose the type that best suits their needs and budget.

Chapter 2: Claims analysis

Requirements analysis for an ERP (Enterprise Resource Planning) system requires understanding the scope, requirements gathering, categorization, structuring, prioritization and evaluation. It refers to the process of systematically collecting, documenting and analyzing an organization's needs and expectations with respect to the implementation of an ERP system.

Requirements analysis usually involves several key steps, such as:

- Identify the scope of the project: This involves defining the specific business processes and functions that the ERP system will be designed to support.
- Requirements gathering: this involves conducting interviews and workshops with stakeholders across the organization to determine their requirements for the ERP system. This may include requirements related to functionality, performance, usability and data management.
- Prioritisation of requirements: once requirements have been identified, they need to be prioritised based on their relative importance to the organisation.
- Documentation of requirements: all requirements shall be documented in detail, including their functional and non-functional characteristics.
- Validation of requirements: requirements must be validated with stakeholders to ensure that they accurately reflect their needs and expectations.
- Management of changes in requirements: As the ERP implementation progresses, changes to requirements may be required. These changes must be managed through a formal change control process to ensure that they are properly evaluated and approved.

Requirements analysis is a critical part of the ERP implementation process, as it helps ensure that the system meets the needs of the organization and its stakeholders. A thorough requirements analysis can help avoid potential problems and delays during the

duration of the implementation and ultimately lead to a successful ERP implementation that will bring tangible benefits to the organization.

The requirements fall into two categories:

• Functional Requirements : They describe exactly what the MA must do, specifying the necessary functions. For example, user accreditation and authorization, business logic, use cases and scenarios, administrative functions.
•Non Functional Requirements : They describe constraints that must be satisfied by the services and functions offered by the system. They are the ones that largely determine the success of the project. Some examples of non-functional requirements include:

- Performance: Number of users, data volume or response time.
- Reliability: Specifies the level of reliability and availability required for the system, including availability requirements and backup and data recovery procedures.
- Usability: Determines the ease of use, accessibility and user experience of the system, including the design of the user interface and accessibility features.
- Security: defines the security requirements of the system, including authentication, access control, data encryption and compliance with security standards.
- Scalability: Determines the ability of the system to handle increased demand, such as increasing numbers of users, transactions or data volumes.
- Maintainability: Determines the ease of maintaining and upgrading the system, including system documentation, error handling, and troubleshooting.
- Interoperability: Identifies the ability of the system to interoperate with other systems or technologies, including data exchange formats, APIs and data integration.

Non-functional requirements are necessary to ensure that the system not only meets its functional requirements but also performs well in terms of other important characteristics. They help to ensure that the system is reliable, efficient, secure and easy to use, which can ultimately lead to greater user satisfaction and improved business results.

Ultimately, however, some requirements may lie "somewhere in between" the above types but this varies depending on the software under development, its operating environment and the stakeholders - its users

2.1.1 Purpose

Our goal is to implement an ERP system for a Secondary Education Tutoring Center that will be intended for students, parents and teachers.

2.1.2 Scope of application

Our system will be called "Academia" because we want it to refer to Plato's Academy (The Academy was founded in Athens around 387 BC by Plato, where there were various scientific fields with philosophy, mathematics, natural and political sciences as objects of study).

Each student will have a personal tab with their details and grades, will have access to the system that will show the hours they have class and the preparation they have to do for each. For teachers there will be a tab where they will fill in the material taught, the lessons for the next session and optional supplementary material to study. In short, there will be a virtual classroom where students and teachers collaborate while the necessary transactions can be made.

With this app, students will be able to know their classes, their times, their absences and their debts anytime and anywhere. Teachers will be able to more easily organize their classes, have their materials organized and the data of past classes to be able to evaluate students and their coursework.

2.1.3 Utility

An ERP system in a tutorial can be very useful for streamlining various administrative and academic processes. Here are some ways in which an ERP can help a business:

- Centralized data management: with an ERP system, all important data related to students, faculty, staff, finances and other administrative and academic processes can be stored in a central database. This makes it easy to access and manage the data efficiently.

- Improved communication: An ERP system can include communication tools that allow for easy communication between parents, students, teachers and staff. This can improve overall communication and collaboration within the school community.

- Saving time and costs: Automating administrative tasks and reducing paperwork can save time and money for the school. An ERP system can automate processes such as admissions, attendance tracking, grading and billing, which can reduce staff workload and save time.
- Analyses and reports: An ERP system can produce reports and analyses that provide information on various aspects of the tutorial's performance.

2.1.4 Market research for existing ones in the sector.

Some popular and highly rated options on the market include:

- PowerSchool: this is a widely used student information system that includes features such as attendance tracking, grade management and communication tools for parents and teachers.
- Blackbaud: This is an integrated school management software that includes tools for managing student information, fundraising and financial management.
- SchoolTime: This is a cloud-based ERP system designed specifically for K-12 schools. It includes features such as student information management, admissions management, and fee management.
- MyClassCampus: This is another popular ERP software for schools, which includes features such as attendance tracking, timetable management and communication tools for parents and teachers.
- Fedena: This is an open source school management software that includes features such as student information management, financial management and messaging tools for communicating with parents and staff.

In Greece, only Greek universities (e.g. EKPA, Metsovio etc. During the period of Covid-19 many applications were developed in secondary schools and foreign language schools in order to carry out the courses online. Students during this period used the **Electronic School Class** (Greek e-class https://eclass.sch.gr).

2.2 General description

This section of the ERP should describe the general factors affecting the product and its requirements. This section does not state specific requirements, it makes these requirements easier to understand

2.2.1 Product perspective

The product after its implementation will not need to be connected to another platform.

2.2.2 Product functions

The user will log on via the internet to the system address where they will be asked for their confirmation codes. Depending on his/her status, the user will be able to perform the actions he/she wishes. The actions of each user will depend on their status. Students will be able to settle their debts, view the day's classes and their grades. Teachers will be able to access previous years and past notes and each student's tab. In general, after the system is implemented, the tutorial will be able to store, manage and visualize its data. Achieve better customer service and improve the efficiency of its employees.

2.2.3 User characteristics

The users of our application will be divided into three categories: students, teachers and employees. Students will be able to see his/her grades, their classes or a message sent by the secretariat-manager. Teachers will be able to fill in the syllabus taught, lessons for the next session and optional supplementary material to be studied. Other users will have access to all the functions of their discipline.

2.2.4 General restrictions

The operations of the platform should respect the personal data of each user in accordance with the relevant legislation.

2.2.5 Assumptions and dependencies

For the operation of the system it will be necessary for the user to be connected to the internet as well as a device with which he can access it. Such a device need not have many operating system requirements.

2.3 Special requirements

This will be the largest and most important part of the requirements analysis. The section will provide the requirements used to guide the design, implementation and testing of the project software.

Any requirement in this section should be:

- Correct
- Traceable
- Undisputed
- Verifiable (i.e. testable)
- Hierarchical (in terms of importance and/or stability)
- Full
- Consistent
- Uniquely recognisable

Care should be taken to carefully organise the requirements presented in this section so that they are easily accessible and understandable.

2.3.1 External interface requirements.

2.3.1.1 user interfaces

The possible user interfaces we were going to need are the following:

- Login and authentication interface:This interface allows users to authenticate themselves to the system, ensuring that only authorized users have access to the application.
- Dashboard interface:This interface provides users with an overview of the key information and features available in the application, such as announcements, calendars, notifications and alerts.
- Student profile interface:This interface allows students to view their grades, attendance records, timetables and other personal information.
- Teacher profile interface This interface allows teachers to manage their classes, lesson plans, assignments and assessments, as well as communicate with students and parents.
- Employee profile interface:This interface allows administrators to manage school data, such as adding or removing users, updating schedules, managing resources and creating reports.

2.3.1.2 hardware interfaces

The possible user interfaces we were going to need are the following:

- Network interface: this interface connects the application servers to the local area network (LAN) or wide area network (WAN) of the tutorial, allowing users to access the application from anywhere on the network.
- Storage interface: This interface connects application servers to the storage devices, such as hard disk drives (HDDs) or solid state drives (SSDs), that store the application data and files.
- Backup interface: this interface connects application servers to backup devices, such as tape drives or cloud storage, that are used to back up and restore application data and files.
- Printing interface: this interface connects the application servers to the printers and scanners used to print or scan documents, such as student files or exam papers.
- Audio/Visual interface:This interface connects application servers to audio/visual equipment, such as projectors or smart boards, used in classrooms or conference rooms for presentations and lectures.
- Security interface:This interface connects the application servers to the security devices, such as surveillance cameras or access control systems, used to ensure the safety and security of school premises.

2.3.1.3 software interfaces

Software Interfaces for this application could include:

- Web browser interface to allow users to access the school application through a web browser on a desktop or mobile device, using standard web technologies such as HTML, CSS and JavaScript.
- Mobile application interface to allow users to access the school application through a mobile application installed on their smartphone or tablet, using platform-specific technologies such as Swift for iOS or Java for Android.
- Database interface to allow the school application to interact with a database management system (DBMS) that stores and manages

the application data, using standard database technologies such as SQL or NoSQL.

- Web services interface to allow the school application to interact with external web services or APIs, using standard web services technologies such as SOAP or REST.
- Application programming interface (API) allows third-party developers to interact with the data or functionality of the application using standard programming languages such as Java, Python or PHP.

2.3.1.4 Communications Interfaces - Communication Interfaces

For our system we will need the following Communications Interfaces:

- Email: to allow users to send and receive emails from within the app, such as sending notifications to parents or students about events or schedule changes.
- Video conferring : Video conferencing - to allow users to participate in virtual meetings, such as parent-teacher conferences or online courses, using video conferencing technologies such as Zoom or Google Meet.
- Social media : Social media - to allow users to communicate and share information through social networking platforms such as Facebook or Twitter, using the social media integration of the school application.

2.3.2 Operational requirements

Some possible functional requirements for our system are:

- The app should be accessible from various devices, such as desktops, laptops, tablets and smartphones, to ensure that students, teachers and parents can use it from anywhere.
- The application should have a user-friendly interface that is easy to navigate and intuitive to use, so that users can quickly and efficiently access the information and tools they need.
- The application should be reliable and scalable to handle the growing number of users and the increasing volume of data as the school expands and evolves over time. It should also have backup and disaster recovery procedures to ensure that data is not lost in the event of hardware or software failures.

2.3.3 Non-Operational Requirements

Often these requirements must be achieved at the system level rather than at the plant level. Some of these may be:

- The application should be able to handle a large number of concurrent users and process requests quickly, with minimal latency or downtime, to ensure a flexible and seamless user experience.
- Security - the application should have strong security measures to prevent unauthorised access, data breaches or other security threats, such as the use of encryption, firewalls, access control and authentication mechanisms.
- Scalability - the application should be able to grow or shrink as the school's needs change, such as managing increasing numbers of users or data volumes, without compromising performance or availability.
- Reliability - the application should be highly reliable and available, with minimal downtime or errors, to ensure that users can access and use the application whenever they need it.
- Usability - the application should have a user-friendly interface and intuitive navigation, with clear and concise instructions to ensure that users can easily and efficiently access the information and tools they need.
- Accessibility - the application should be accessible to users with disabilities, such as providing support for screen readers, keyboard navigation or colour contrast to ensure equal access for all users.
- Maintainability - the application should be easy to maintain and update, with clear documentation, modular design and low coupling to ensure that future changes or improvements can be made without disrupting existing functionality or causing errors.

Chapter 3: Analytical Design

The resource management subsystem of the business management system can be designed using UML (Unified Modeling Language) diagrams to represent the different parts of the system. A detailed design of each part of the system using UML diagrams follows:

- Class Diagram-Class Diagram: this diagram defines the classes, their attributes and methods that will be used in the resource management system. It also shows the relationships between the different classes and their dependencies.
- Use Case Diagram: this diagram is used to show the functional requirements of the system. It depicts the use cases that the system must support and the entities that interact with the system. The actors in this subsystem may include employees, managers and administrators who use the system to manage resources.
- State Machine Diagram-State Machine Diagram: this diagram describes the various states and events that can occur during the operation of the resource management system. It can be used to model the behaviour of the different objects in the system.
- Activity Diagram-Activity Diagram: this diagram shows the workflow or sequence of actions involved in managing resources. It can be used to model business processes, the activities of different actors and how they interact with the system.
- Sequence Diagram-Sequence Diagram: this diagram depicts the interactions between objects or elements of the system over time. It shows how objects communicate with each other to perform a particular task in the system.
- Communication Diagram-Collaboration or Communication Diagram: this diagram depicts the communication between different objects or components of the system. It shows the messages exchanged between the

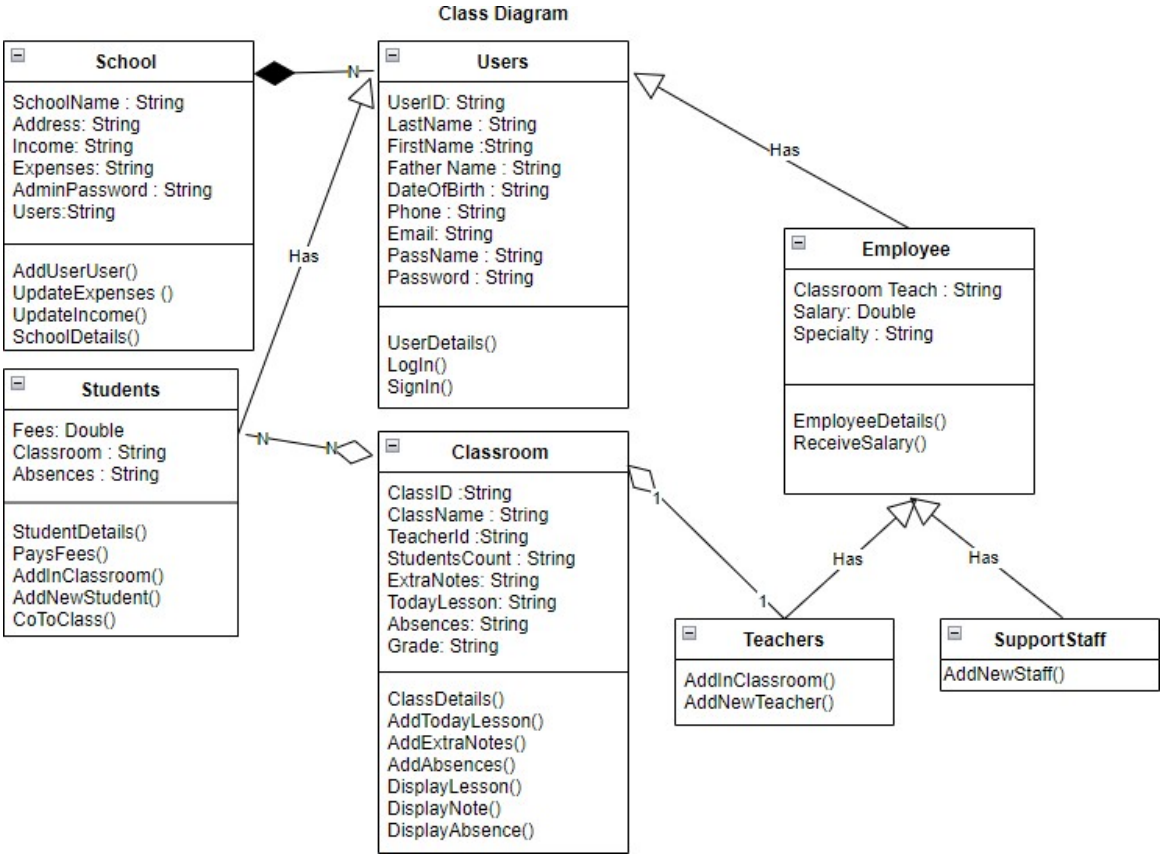
objects and how they interact to perform a specific task in the system

- Object Diagram-Object Diagram: this diagram is used to illustrate instances of classes and their relationships. It shows how the various objects are created and managed in the resource management system.
- Component Diagram-Component Diagram: this diagram shows the various components of the resource management system and their dependencies. It also shows the interfaces of the components and how they interact with each other.
- Deployment Diagram-Deployment Diagram: this diagram shows the physical architecture of the system, including the hardware and software components, and how they are deployed. It can be used to model the distribution and communication between the different components.

Overall, these UML diagrams can be used to design and develop the resource management subsystem of the business management system, ensuring that all aspects of the system are considered and accurately represented.

3.1 Class Diagram- Class Diagram

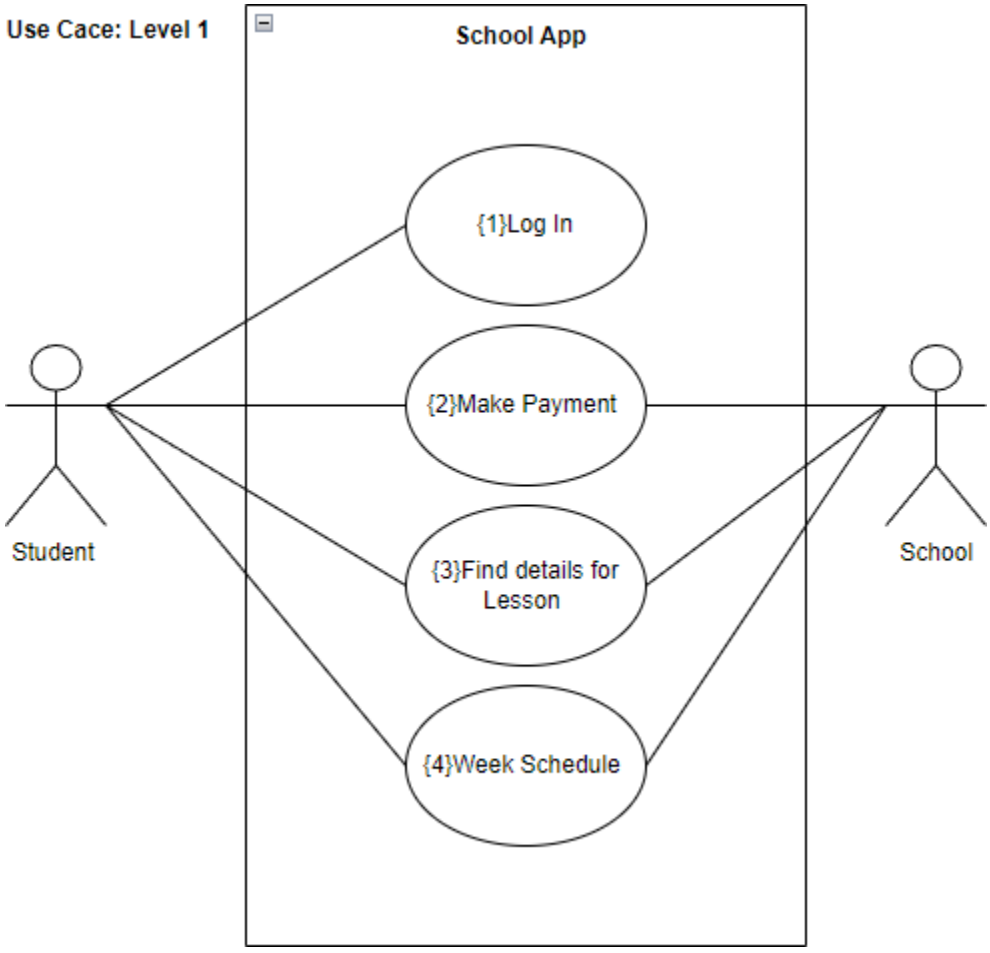
UML class diagrams are used to visualize the structure of a system, showing the classes of the system and the relationships between them. A class diagram typically consists of classes, interfaces, associations, and inheritance relationships among other elements. They are useful for understanding the design of a system, communicating that design to others, and assisting in the implementation of the system



3.2 Use Case Diagram - Use Case Diagram

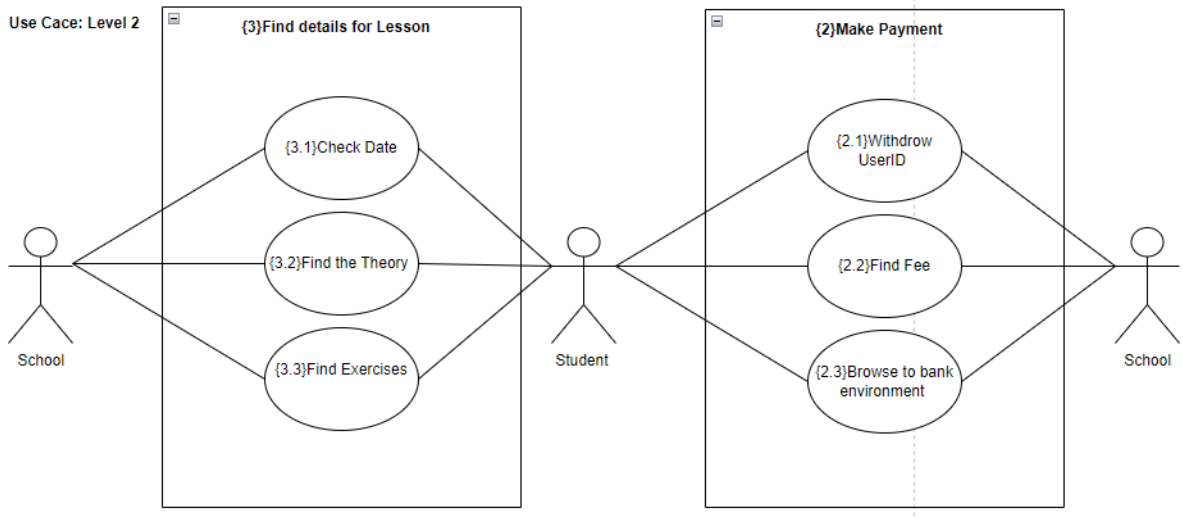
Level 1: Use case diagram

At level 1, we can identify the main actors and use cases (in our case only students) of the Resource Management System of the Business Management System. The diagram will show the main interactions between the actors and the system.



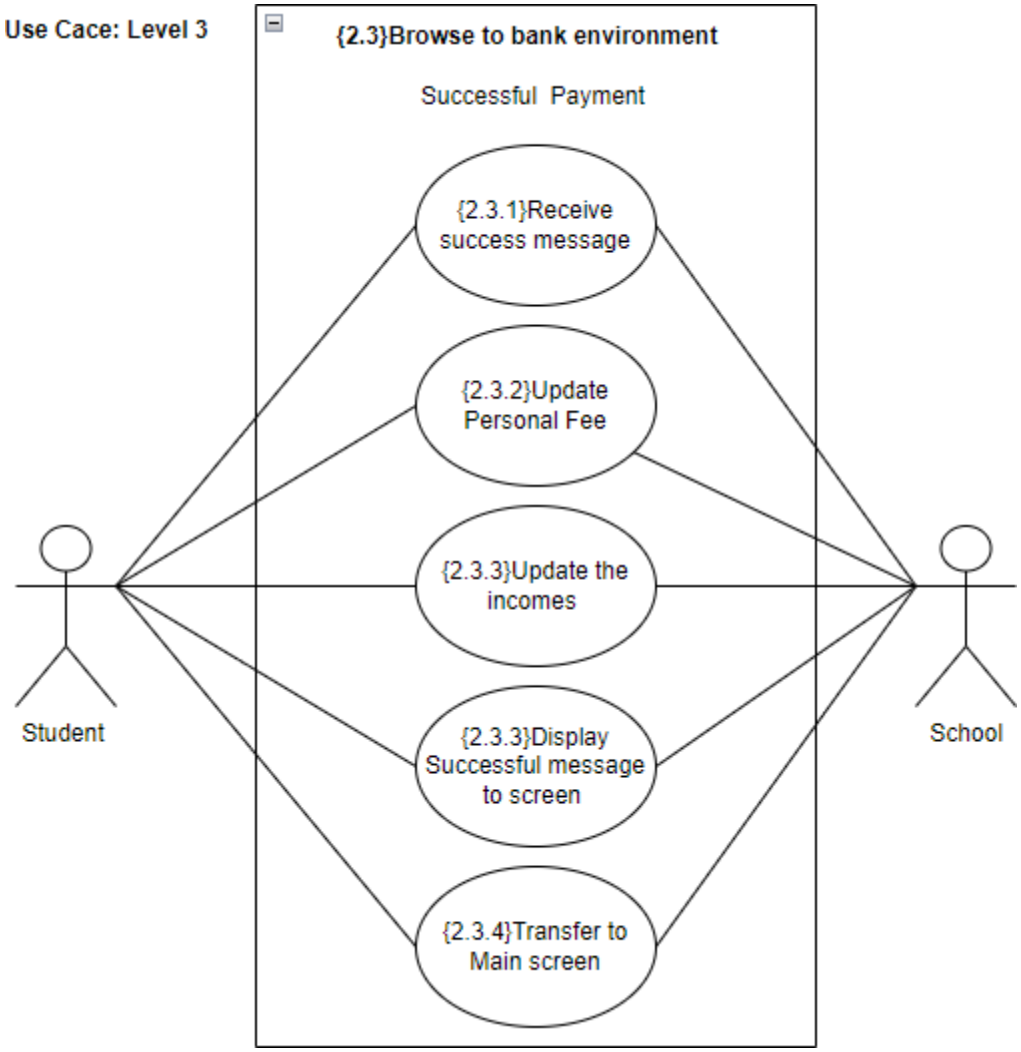
Level 2: Use case diagram

At level 2, we can refine the use cases identified in the level 1 diagram by breaking them down into smaller, more specific use cases. This will help us to better understand the requirements of the system and how the different use cases are linked.



Level 3: Use case diagram

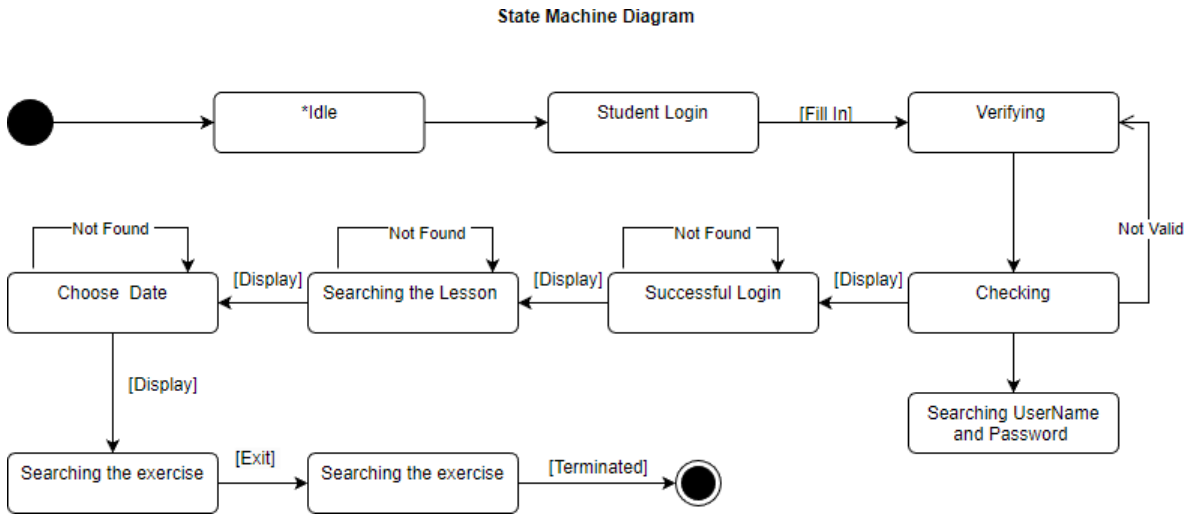
At level 3, we can further refine the use cases identified in the level 2 diagram by adding more details, such as conditions, meta-conditions and exceptions. This will help us to define the behaviour of the system more accurately. For our example, we will assume that the user has successfully completed a transaction and returns to the application.



Overall, UML case diagrams can help us to design the Resource Management System of the Business Management System in a systematic and structured way, ensuring that all requirements are identified and addressed.

3.3 State Machine Diagram-State Machine Diagram

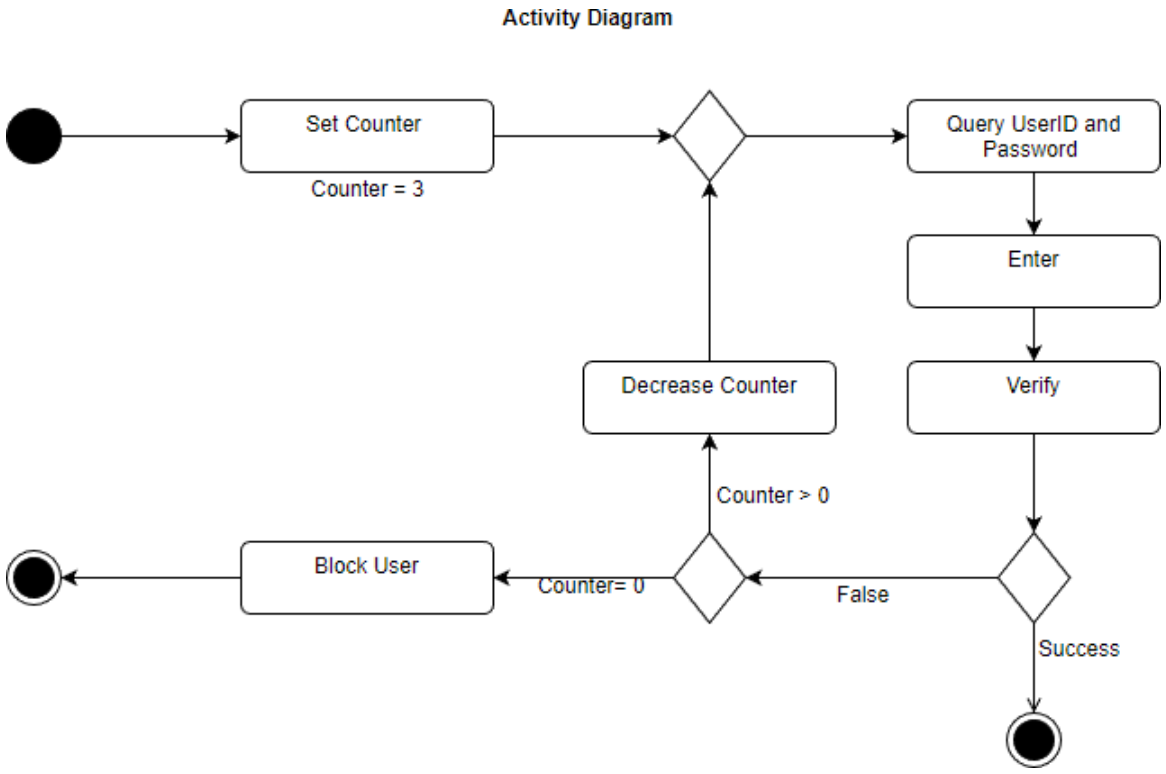
UML's State Machine Diagrams are used to model the behaviour of a system, showing the various states the system can be in and the transitions between these states. It usually consists of states, transitions, events and actions, among other elements. State machine diagrams are useful for understanding the behavior of a system, testing the behavior of the system, and helping to implement the logic of the system. Below we will have the diagram where the user wants to see some exercises to read.



* "Idle" refers to a state of inactivity or non-use. It is often used to describe a device, system, or person who is not currently engaged in an activity or task.

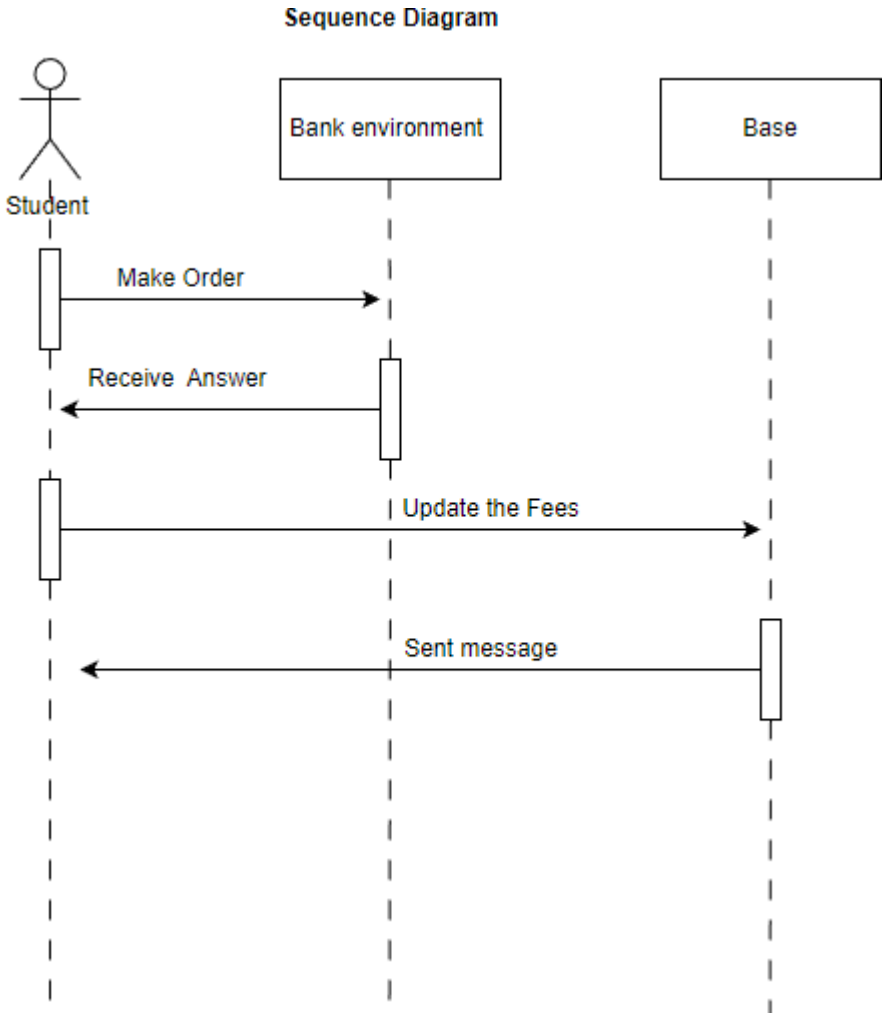
3.4 Activity Diagram-Activity Diagram

UML activity diagrams are used to model the flow of activities or processes in a system or business process. An activity diagram typically consists of actions, decision points and flows, among other elements, and can be used to model complex processes with many branches and decision points. Activity diagrams are useful for understanding the flow of a process, identifying inefficiencies or bottlenecks, and assisting in the planning and implementation of the process. Below we will have the diagram where the user wants to Log in to our application.



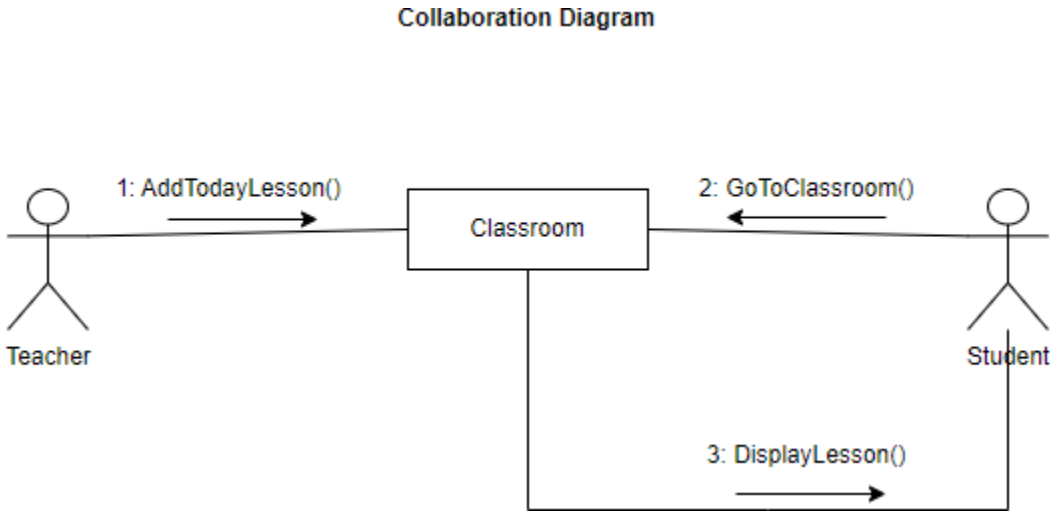
3.5 Sequence Diagram-Sequence Diagram

UML Sequence Diagrams are used to illustrate the interactions between objects in a system or application, showing the messages that are exchanged between them over time. A sequence diagram typically consists of lifelines, messages, and trigger bars, among other elements, and can be used to model complex scenarios with multiple objects and interactions. Sequence diagrams are useful for understanding the behavior of a system, identifying potential errors or inefficiencies in the interactions between objects, and helping to implement and control the system. Below we have the diagram where the user successfully performs a transaction and returns to the application.



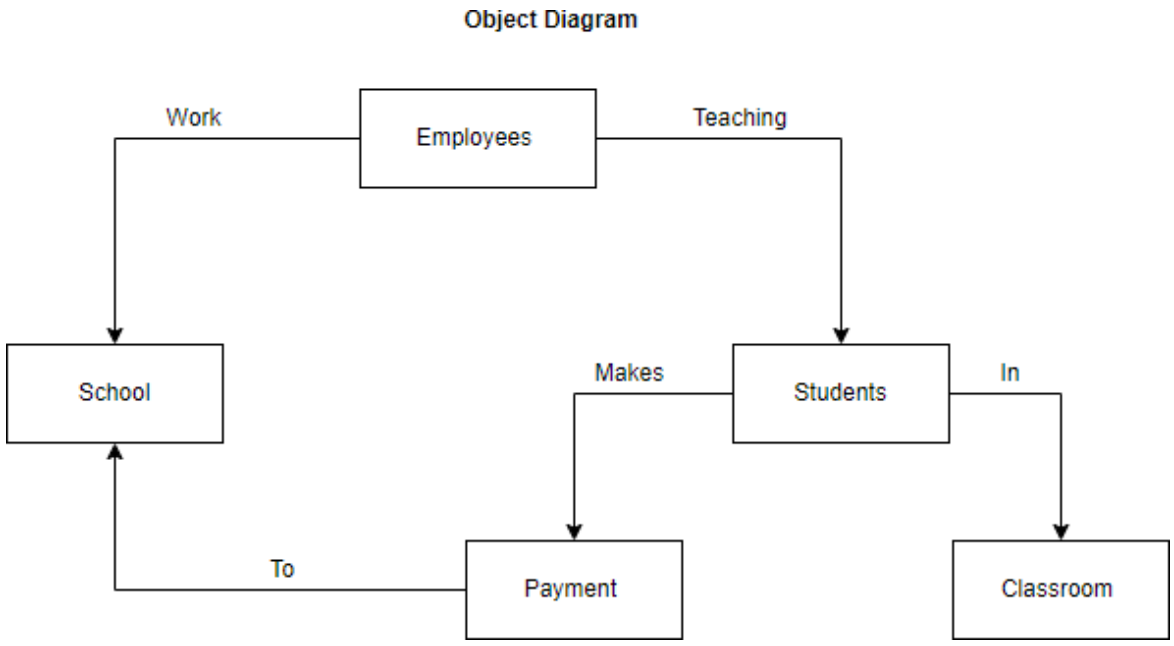
3.6 Communication Diagram-Collaboration Diagram

Collaboration Diagrams in UML are used to show the interactions between objects in a system or application, emphasizing the relationships between objects rather than the order in which messages are sent. A collaboration diagram typically consists of objects, links, and messages, among other elements, and can be used to model complex scenarios with many objects and interactions. Collaboration diagrams are useful for understanding the relationships between objects in a system, identifying potential errors or inadequacies in those relationships, and helping to design and implement the system. For our example, we will see how a student-teacher communication takes place through our system.



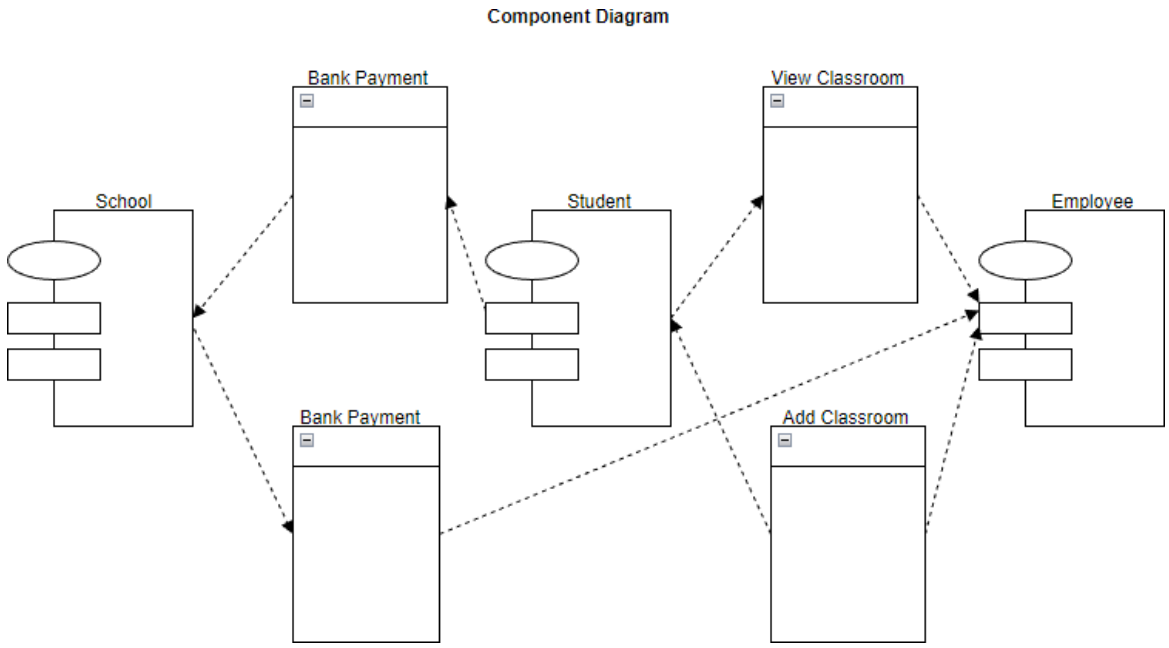
3.7 Object Diagram-Object Diagram

UML Object Diagrams are used to present a snapshot of the objects of a system and their relationships at a particular point in time. An object diagram typically consists of objects, classes, and links between other elements, and can be used to model the structure of a system or to aid in debugging. Object diagrams are useful for understanding the relationships between objects in a system, for verifying the correct design of the system, and for helping to test and debug the system.



3.8 component diagram

Component diagrams in UML are used to represent the components of a system and their relationships between them, as well as their interactions with external systems or components. A component diagram typically consists of components, interfaces, ports, and connections between other components. Component diagrams are useful for showing the high-level architecture of a system and for identifying the dependencies between components, which can help in planning and managing the development and maintenance of the system.



3.9 Deployment Diagram-Deployment Diagram

Deployment Diagrams in UML are used to model the physical deployment of components in a system or application, showing how software is installed on hardware. A deployment diagram typically consists of nodes, elements, and links between other elements, and can be used to model complex deployment scenarios with multiple hardware and software components. Development diagrams are useful for understanding the physical architecture of a system, identifying potential problems or bottlenecks in development, and assisting in the implementation and maintenance of the system.

There are three levels of architecture in a deployment diagram:

- The first architectural layer in a deployment diagram usually shows the physical nodes that make up the system, such as servers, clients, and other hardware devices.
- The second architectural layer in a deployment diagram usually shows the software components deployed on each node, such as web servers, application servers, and databases.
- The third level of architecture in a deployment diagram can show the specific instances of each software component running on each node, as well as the communication paths between nodes and components.

Deployment diagrams