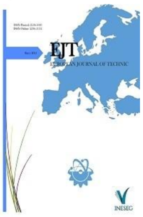
European Journal of Technique

journal homepage: https://dergipark.org.tr/en/pub/ejt

Vol.13, No.2, 2023



Research Article

Artificial Intelligence-Based Tools in Software Development Processes: Application of ChatGPT

Zeynep Özpolat[1*], Özal Yıldırım[1] and Murat Karabatak[1]
[1*]Firat University, Department of Software Engineering, 23200, Elazig, Turkey. (zynpaltinterim@gmail.com).

ARTICLE INFO

Received: Jul., 20. 2023
Revised: Aug., 12. 2023
Accepted: Aug, 24. 2023

Keywords:
Artificial Intelligence
Software Development
ChatGPT-4
AI tools

Corresponding author: *Zeynep Özpolat*

ISSN: *2536-5010* | e-ISSN: *2536-5134*

DOI: https://doi.org/10.36222/ejt.1330631

ABSTRACT

Software development processes are constantly evolving and transforming rapidly with rapid changes in technology. Recent innovations in Artificial Intelligence (AI) have led to significant changes in software development practices. AI tools can greatly improve traditional software development processes by giving developers the ability to build projects smarter, faster and more effectively. These tools can be used for a variety of tasks such as code generation, test automation, bug analysis, and performance improvements. ChatGPT, which is an artificial intelligence-based language model and has made a deep impact in almost every field, can help software developers write code faster and in a more natural language. In this study, the contributions of ChatGPT to a software development project were investigated. In the study, basic information about the use of ChatGPT in the software development process is presented. Implementations were carried out on a software project to evaluate some of ChatGPT's capabilities in the context of software development. For this purpose, a software development process was designed based on the answers given by ChatGPT. Various questions about software development processes were formulated and the answers produced by the GPT were evaluated. The results obtained showed that ChatGPT performed excellently in the software development process. Based on these findings, it has been observed that AI-based models such as ChatGPT can be effectively used as auxiliary tools in software development processes that accelerate traditional workflows. In addition, AI-based tools can save time and effort while improving software quality by automating testing processes.

1. INTRODUCTION

Software development processes are generally referred to as the steps followed during the design of a software project. These steps commence with identifying the necessary analyses for the desired software. Subsequently, the process continues with stages such as design, coding, testing, and deployment [1]. During project execution, collaborative efforts among stakeholders lead to the creation of requirements. Following the identification of requirements, the transition is made to the coding phase for the designated structures in the design stage. Finally, a testing phase is implemented to measure the efficiency of the resulting product [2]. Software projects must be completed within the defined timeframe, within the financial resources allocated to the project, and with the best possible quality that meets expectations. Throughout these steps, effective communication among the teams and stakeholders within the project stands as one of the most critical factors contributing to the success of projects. Human resources are prominent in each phase of traditional software development processes. These processes involve a wide array of integrated development environments (IDEs), test automation tools, project management tools (such as Jira, Trello, Asana, etc.), and database management tools, which significantly support software developers [3]. Depending on project requirements and the team's preferences, appropriate tools are selected to manage the software development processes efficiently. Despite the facilitation provided by these software tools, there is a need for tools capable of making rapid and effective decisions and enabling automated process management in this domain. Therefore, Artificial Intelligence (AI)-based tools have gained significant usage in software development processes, just as they have in various other fields, as of late

In its simplest form, AI can be defined as computer software capable of imitating the human mind and a set of tools integrated into this software [4]. With AI systems used in software development processes, tasks such as decision-making, prediction, identification, and pattern recognition become more manageable. AI, which provides a different perspective to traditional software development coding, utilizes popular areas such as deep learning and machine learning in the process. Noteworthy applications of AI in social life include the generation of fake videos, creation of non-existent human faces, and manipulative voice applications. Recently, ChatGPT [5] has been released as a tool that has made a significant impact in this field. Generative Pre-trained

Transformer - 4, also known as ChatGPT-4, is an AI application. Designed as a multi-language processing platform, ChatGPT is an application that can be beneficial in various processes within the software field.

There are several research studies investigating how ChatGPT can assist in software development and engineering. Ahmad et al. [6] presented research on how architecture-centric software engineering (ACSE) professionals can benefit from ChatGPT. They focused on designing the software process with ChatGPT and the factors to consider in the ACSE & ChatGPT partnership. The authors discussed the advantages and disadvantages of ChatGPT-supported ACSE and provided solutions for implementing human-bot collaborative work. Nascimento et al. [4] compared the performance of software engineers and AI-based systems, including ChatGPT. Through various evaluation criteria, they analyzed the tasks assigned to ChatGPT and inferred that it outperformed engineers on certain software engineering tasks, especially those of easy to medium-level complexity. However, in some studies, human engineers were successful, while in others, AI prevailed. Katar et al. [7] employed ChatGPT in their research paper and presented results, highlighting both the positive and negative aspects of using ChatGPT as an auxiliary tool. The authors concluded that while ChatGPT is not yet efficient enough to write a research paper entirely on its own, it can be a valuable tool to assist researchers in the writing process. Akbar et al. [8] conducted a motivational, demotivational, and ethical evaluation of software engineers' usage of ChatGPT. They listed the ethical challenges posed by ChatGPT, such as risks of plagiarism, confidentiality breaches, data security issues, and the generation of malicious data. The authors conducted a comprehensive survey among software engineers and performed Cross-Impact Matrix Multiplication Applied to Classification (MICMAC) analysis to create a cluster-based decision model. Fraiwan and Khasawneh [9] provided an overview of emerging language models and chatbots. They emphasized that AI-based tools have the potential to offer different perspectives in areas such as education, software engineering, healthcare, and marketing. However, they also noted that these systems come with challenges, including issues of plagiarism and lack of transparency. The authors conducted a study investigating the positive and negative effects of language models actively used in fields like education, health, software engineering, and marketing, and discussed how to take precautions against potential negative consequences and misuse scenarios.

This study presents both traditional software development processes and the use of AI tools in these processes. Traditional software development processes are explained in general terms and the effects of contemporary AI tools on these processes are discussed. The contributions of this article to the literature can be listed as follows:

- ➢ As far as the authors know, the existing studies in the literature are generally completed with the support of a software development specialist. This study has been tried to be completed entirely with the support of AI.
- ➢ The answers to these questions are included in the article without making any changes to the questions asked to ChatGPT-4 for an ERP project.
- ➢ This article is of great importance as it sheds light on the effectiveness of using ChatGPT in software project development.

The remaining parts of the study are as follows: Section 2 gives general information about software development processes. In Section 3, the material and method are mentioned. Questions directed to the AI system are given in this section. Section 4 includes the results of the experimental study created in line with the answers from ChatGPT-4. In Section 5, discussions about the study and the general results of working with Section 6 are mentioned.

2. SOFTWARE DEVELOPMENT PROCESSES

Software is a tool used by a software team to define the way devices behave. The sequence of cyclical stages used to describe a software process is defined as the software life cycle [1]. Each software follows a predefined sequence of steps according to the Software Development Life Cycle (SDLC) [10]. The Software Development Life Cycle is defined as a guideline and logical process used by system developers to develop systems [11]. A basic diagram of the software life cycle is given in Fig.1.

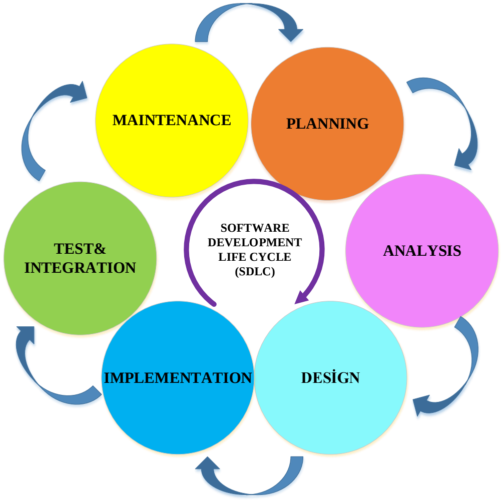


Figure 1. Software Development Life Cycle.

The steps of a basic software lifecycle and the operations performed in each of these steps can be listed as follows [7]:

- **Planning:** This is the first stage of the cycle where the requirements and needs of the users are taken into consideration. Here, the objectives are clarified by determining the systemic and technical needs required for the work to be created in parallel with the user requirements. The work given by the user for a purpose turns into a project at this stage.
- **Analysis:** It is a stage in which the estimated time for the desired outputs within the project and the risky situations that may be encountered in the project during this period are revealed. Possible situations are detailed using UML diagrams.
- **Design:** In line with the planning and analysis made in the previous stages, the drawing, i.e. the design of the project begins. The basic structure of the software to be used in the project, its algorithm, interface, software components and all kinds of technical details of the project are determined before proceeding to the next stage, implementation. All the theoretical and practical decisions regarding the project are made and the next phase begins.
- **Development:** Development is defined as the part where the decisions made in the design phase are now poured into a code environment. It is the stage where the project, which has all the theoretical infrastructure ready, is put into practice. Programming languages and software

development tools determined in accordance with the project are used.

- **Testing:** The start of the trial process of the codes, programs or software-based tools prepared for the project takes place in the testing phase. At this stage, the project's compatibility with real life is tested by applying techniques such as code correctness, detection of software errors, performance tests, usability evaluation. In addition to the control of software processes, providing software training to the people or units that will benefit from this project is also included in the testing phase.
- **Maintenance:** Where tasks such as eliminating errors and making improvements are carried out as long as the software or project is used the stage.

During the maintenance process, certain requirements that were not initially included in the software may arise in response to the users' needs. In this case, the cycle goes back to the beginning and is reconstructed from the planning stage. The reason why the software development process is presented as a life cycle can be summarized concisely. In this life cycle, which comprises fundamental stages, various traditional models are employed to organize and streamline the processes. Some of the traditional software development models include the 'Waterfall', 'V Model', 'Prototype', and 'Spiral'. The 'Waterfall' model is the first of the process development models and is the oldest SDLC approach. It follows a linear and sequential approach, often referred to as the linear-sequential life cycle model in some sources [12]. It includes 6 steps: planning, design, implementation, testing, deployment, and maintenance [13]. It is easy to comprehend and yields favorable results, particularly for small projects with well-defined requirements. The 'V Model' is often described as an extension of the waterfall model. This is because it has a similar sequential and linear flow. But differently, the process steps are bent upwards after the coding phase to form the V-shape [14]. The planning, design, implementation, and testing steps take place before coding, which saves time [13]. While early test preparation offers the advantage of detecting bugs at an early stage, excessive focus on the product itself rather than the testing phase can lead to potential issues [15]. The 'Prototype' model is an approach that facilitates the development of an initial prototype to comprehend customer requirements. Based on customer feedback, the prototype is continuously improved, and a process is followed to create the final product. The steps in this model encompass business modeling, data modeling, process modeling, implementation, and testing [14]. The 'Spiral' model was introduced in 1988 to solve the limitations of the waterfall model. It is an iterative software model, combining the features of prototyping and the waterfall model. The spiral model consists of four phases starting with planning, objectives, risk analysis, and development. The model organizes all activity inputs in a spiral, and all phases are repeated continuously. Spiral development commences on a smaller scale and expands based on the number of iterations. Since not all requirements are specified at once, additional studies may be needed [16].

2.1. Artificial Intelligence-Based Software Development Tools

Software development is the foundation of the information age society. It is one of the most important elements of the world of information processing, as well as being a complex structure that is constantly changing. In the modern digital age, the development of software solutions is a crucial issue to keep pace with the rapidly evolving technology [17]. Software development processes play an active role in product quality and process efficiency. Issues such as requirements specification and error detection can be considered as basic initial activities. Supporting, changing, periodically evaluating, and increasing the potential of these activities requires continuous optimization. Software development processes are costly and time-consuming. At this point, AI-supported technologies work as a model that increases the effectiveness, speed, efficiency, and potential of software development processes [18].

AI has a long history in computer science, but more recently it has been making practical achievements, mostly in machine learning. It also develops software that can learn in an automated, human-free environment using sophisticated algorithms and large data sets. This environment of success not only provides a good understanding of decisions in critical decision-making processes but also increases the acceptance of AI-enabled developments [19]. The use of AI applications in development processes is usually aimed at automating software cycle phases. It can be applied to every phase of the software development cycle, including analysis, design, code, testing, and maintenance. When the most popular AI models in recent years are examined, it is seen that they are classified under the headings of natural language processing, computer vision, autonomous driving, and health [17]. In order to draw a framework for the integration of AI technologies into software development processes, models can be emphasized. Many AI models such as Automatic Programming, Artificial Neural Networks, Machine Learning, Knowledge-Based Systems, and Natural Language Processing can be applied in software engineering stages [20].

The phases where AI models contribute the most are requirements analysis and testing. Requirements analysis is the first stage of the software process. Researchers often incorporate AI into this first step of the cycle. The Natural Language Processing model, a semi-automated approach, provides useful results here, shortening the time needed for the requirements step and improving its quality [21]. Automated Programming is involved in the automated process of generating code, reusing code, and refactoring code and offers an intermediary association. The possibility of automated programming assistants to provide pre-made models for specific situations and to demonstrate best practices is very important here [22]. Machine Learning is used to link information from structural and fault-based testing to functional aspects of the program. This linked information links results to different test technique implementations and offers ease of testing [23].

In summary, it can be said that the goal of AI is to achieve intelligence. The software development cycle, on the other hand, aims to create a system that is valid, verified, cost-effective, and without any maintenance or user acceptance issues. Despite the dilemmas that these two fields have, it is also very difficult to achieve overlap. More studies are needed to increase the success and identify the needs correctly. At this point, some studies have raised questions such as "If the cycle succeeds in becoming fully intelligent, won't both domains become a single domain?", "if AI becomes the traditional software development life cycle itself, will this mean the end of the cycle?" [20]. There are some AI – support programs currently used by software developers such as TensorFlow, PyTorch, Keras, Scikit-learn, Apache MXNet, Microsoft

Cognitive Toolkit, IBM Watson Studio, H2O.ai, Google Cloud AutoML, Amazon CodeGuru, Bugspots [24]. Each of these and similar programs aims to automate tasks, reduce errors and maximize efficiency.

3. MATERIAL AND METHODS

In this study, software development processes were implemented step by step on a project using an AI-based tool.

Using the recently popular Chat-GPT model, software development processes were managed in line with the questions identified. The block diagram of the work carried out within the scope of this article is presented in Fig.2. As can be seen in Figure 2, the questions containing the parts of the software development processes are given as Q1, Q2,...,Q13. Prepared questions were directed to ChatGPT-4. Answers produced by ChatGPT have been evaluated by an expert.

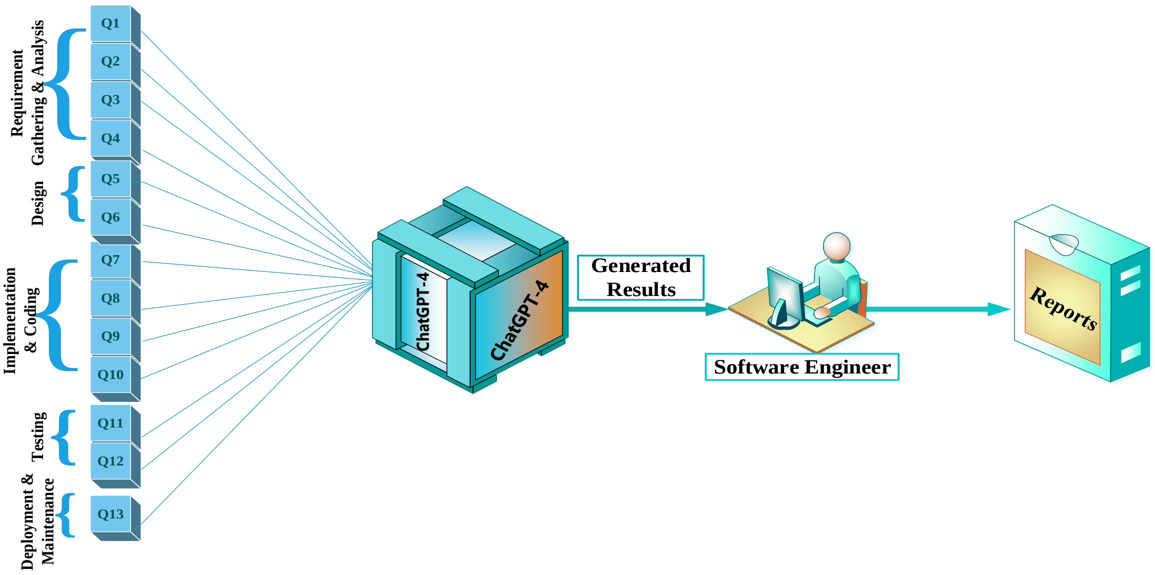


Figure 2. Block diagram of the material and method.

In this study, appropriate questions were formulated for each step of the software development process using the ChatGPT-4 tool, and the corresponding answers were obtained. During the question preparation phase, a suitable set of inquiries was carefully crafted, taking into account the software development processes. The intended software system to be constructed using ChatGPT was determined as an enterprise resource planning (ERP) project. The ERP system is a software solution designed to address both specific and general needs across all units within a company [25]. Traditionally employed by large companies, the demand for ERP systems has expanded in recent years to include medium and small-sized companies. Consequently, the software development field has experienced accelerated growth to meet the increasing market demands. To swiftly address these requirements, the development of AI-supported solutions has become imperative, complementing the efforts of human personnel.

This study seeks to address the question of "How can the software development process for an ERP project be realized?" utilizing ChatGPT, the most popular application of AI. Accordingly, various questions were posed to ChatGPT, and the ensuing answers are presented herein.

3.1. ChatGPT Input Data

In this article, which aims to demonstrate the use of an AI tool in a software development process, ChatGPT-4 was used in the process creation phase. For the ERP project that ChatGPT was asked to do, questions were asked at each step. In line with the answers given by ChatGPT, some suggestions given by ChatGPT were directed to ChatGPT as questions again. The question-and-answer process was carried out mutually in this way. The questions prepared within the scope of the article study and the categories that these questions correspond to in software development processes are given in Table 1.

TABLE I

Questions asked to ChatGPT-4 to create a software development process

Categories	Q	Questions
Requirement Gathering & Analysis	Q1	Can you create a comprehensive flowchart for an ERP project?
	Q2	Can you create a flowchart for the new ERP system with these modules you suggest to add?
	Q3	Can you create a software lifecycle content in accordance with this flowchart?
	Q4	What can be done during the requirement gathering and analysis phase for the ERP project we plan to do?
Design	Q5	What can be done during the design phase for the ERP project we plan to do?
	Q6	Can you draw a UML diagram covering each stage for this ERP project you have designed?
Implementation/ Coding	Q7	What kind of process can be followed in implementation/coding? Can you inform me about the coding programs to be used, database system etc.?
	Q8	It is planned to use Python programming language and Oracle as database for the designed ERP project. In this direction, can you first create the database?
	Q9	Can you create the ER diagram for the database you created?
	Q10	Can you give the Python codes for the ERP project based on the database created?
Testing	Q11	What kind of process can be followed during the testing phase for the ERP project?
	Q12	Can you give an example ERP project where the performance test is applied from these tests?
Deployment & Maintenance	Q13	Can you talk about the steps required for deployment and maintenance in an ERP software development project?

4. EXPERIMENTAL RESULTS

Before designing an ERP project, ChatGPT-4 was asked to create a flowchart. The "Show Me Diagrams" plugin from the Plugin Store was used to generate the diagram. Figure 3 shows the question Q1 given as input for ChatGPT and the diagram created for this question. As can be seen in Figure 3, ChatGPT presented a project scope consisting of 4 parts for an ERP system in response to the question asked. ChatGPT also referred to the compartments that may be needed in the subsections of each chapter.

Answer for (Q1):

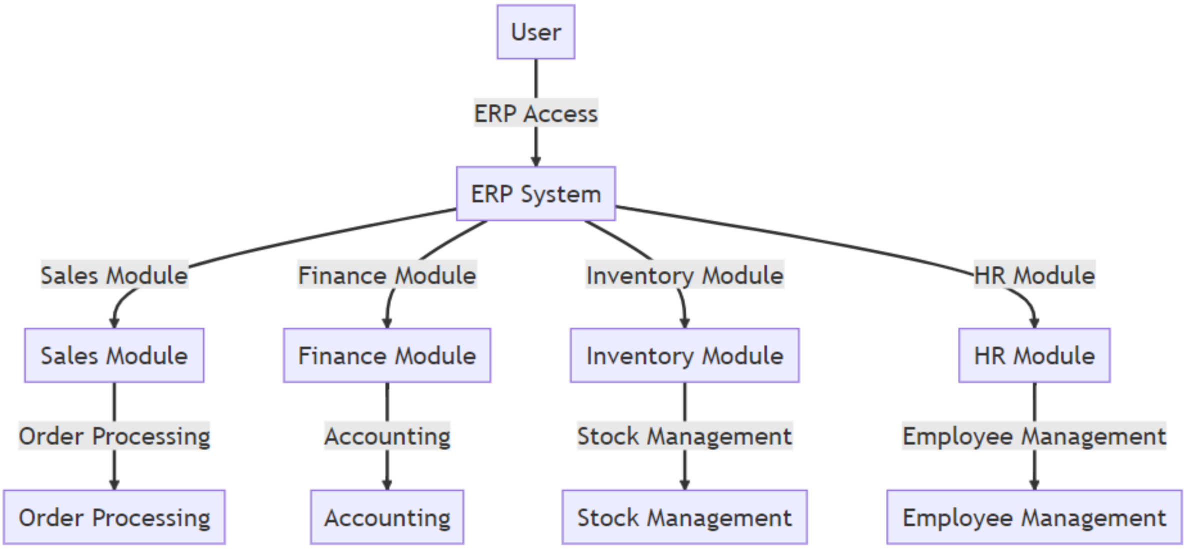
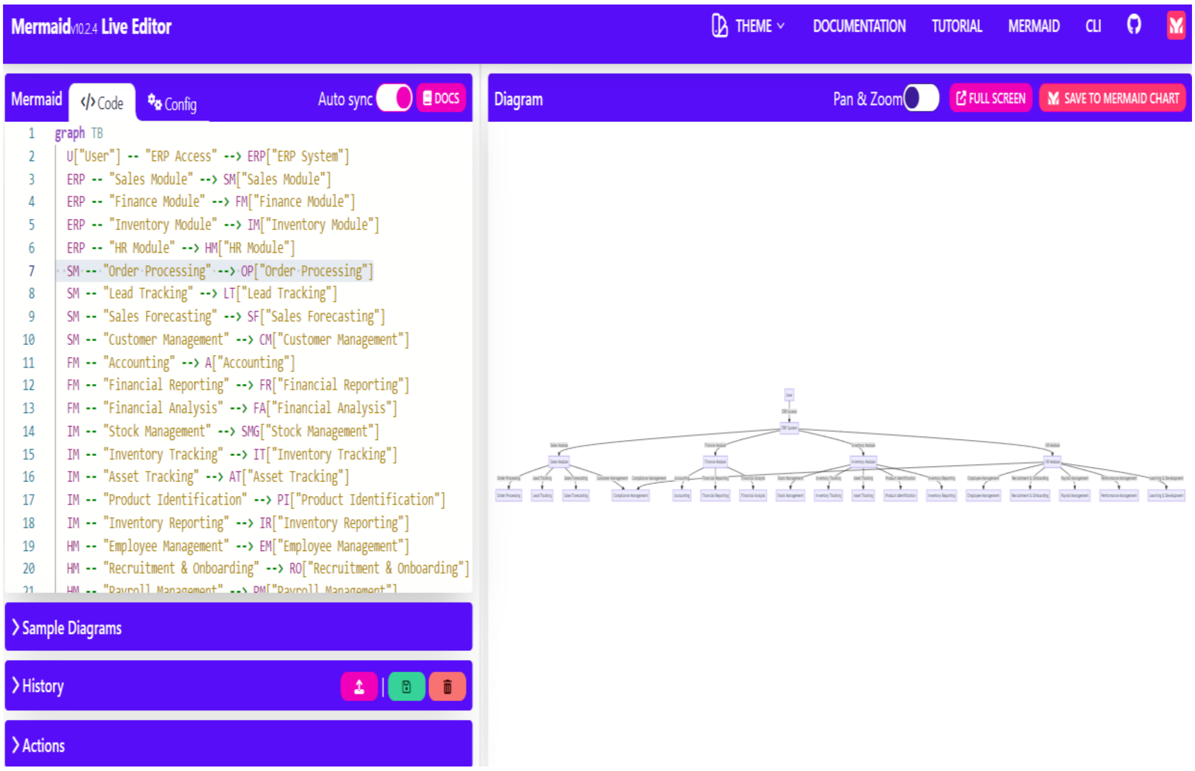


Figure 3. Flowchart for an ERP project

After this answer, ChatGPT-4 suggested "To view ideas for improving the diagram, use the key phrase "show ideas"." This question was asked again, taking into consideration the advice given by ChatGPT-4. Afterward, the flowchart was expanded a little more based on the suggestions given by ChatGPT-4.

Answer for (Q2): Ideas to improve the diagram:

Add More Detail: You could add more detail to each module. For example, the Sales Module could be broken down into "Lead Tracking", "Sales Forecasting", "Order Management", etc. Similarly, the Finance Module could be broken down into "Accounts Payable", "Accounts Receivable", "General Ledger", etc.



```
graph TB
  U["User"] -- "ERP Access" --> ERP["ERP System"]
  ERP -- "Sales Module" --> SM["Sales Module"]
  ERP -- "Finance Module" --> FM["Finance Module"]
  ERP -- "Inventory Module" --> IM["Inventory Module"]
  ERP -- "HR Module" --> HM["HR Module"]
  SM -- "Order Processing" --> OP["Order Processing"]
  SM -- "Lead Tracking" --> LT["Lead Tracking"]
  SM -- "Sales Forecasting" --> SF["Sales Forecasting"]
  SM -- "Customer Management" --> CM["Customer Management"]
  FM -- "Accounting" --> A["Accounting"]
  FM -- "Financial Reporting" --> FR["Financial Reporting"]
  FM -- "Financial Analysis" --> FA["Financial Analysis"]
  IM -- "Stock Management" --> SMG["Stock Management"]
  IM -- "Inventory Tracking" --> IT["Inventory Tracking"]
  IM -- "Asset Tracking" --> AT["Asset Tracking"]
  IM -- "Product Identification" --> PI["Product Identification"]
  IM -- "Inventory Reporting" --> IR["Inventory Reporting"]
  HM -- "Employee Management" --> EM["Employee Management"]
  HM -- "Recruitment & Onboarding" --> RO["Recruitment & Onboarding"]
  HM -- "Payroll Management" --> PM["Payroll Management"]
```

Figure 4. Code page for editing flowchart on Mermaid program

There is a screen where the sections on the Code tab can be deleted and added and the given diagram can be edited as desired. A part of the new diagram obtained after various modifications is shown in Fig.5 below

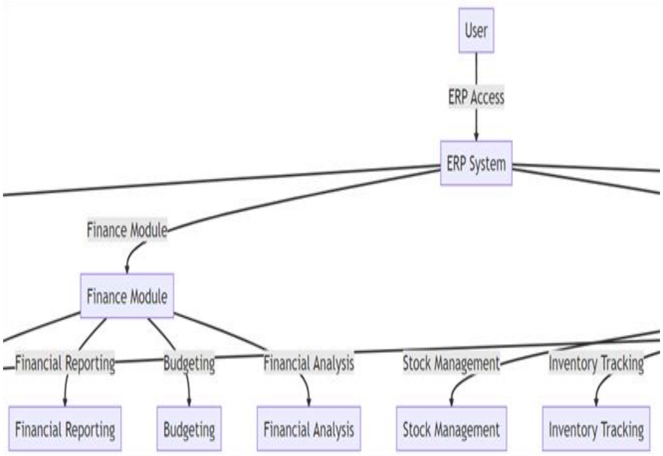


Figure 5. A part from the final version of the modified flowchart on Mermaid

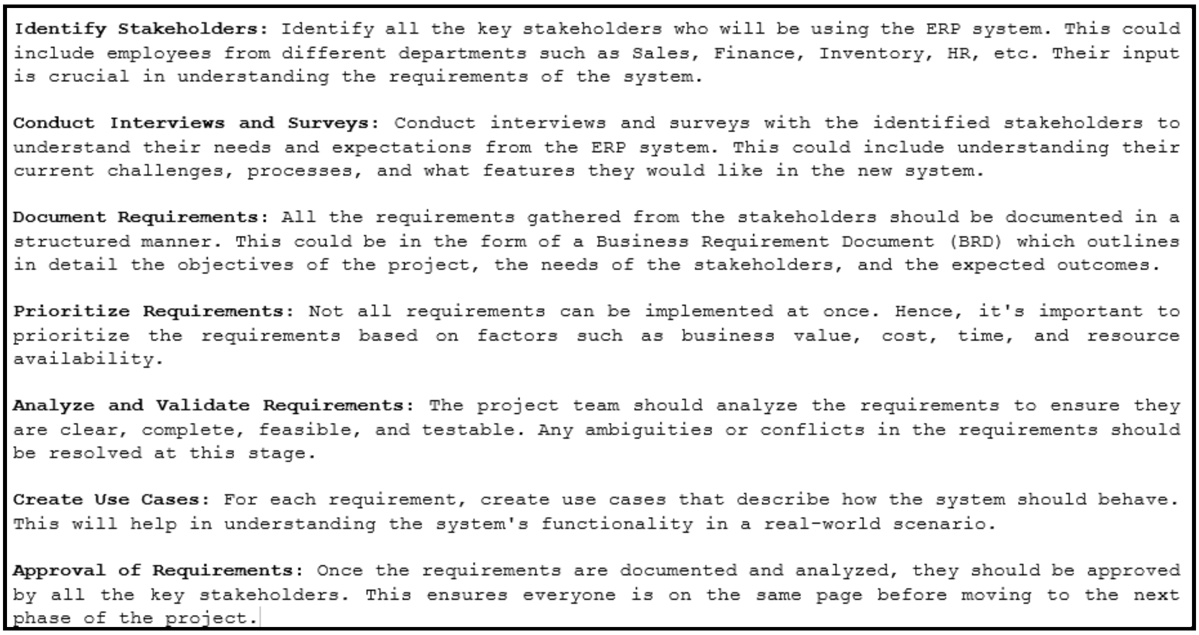
After this diagram, ChatGPT-4 was asked to give the steps of the software lifecycle in accordance with the project (Q3). ChatGPT's mixed response to this request is presented in the following items.

Answer for (Q3):

- Requirement Gathering and Analysis: This is the initial stage where the project team (including business analysts, project managers, and key stakeholders) define the business requirements. They identify what modules are needed (Sales, Finance, Inventory, HR, etc.) and what features each module should have.
- Design: Based on the requirements, the system architecture is designed. This includes how the different modules will interact with each other and the database schema. The user interface for each module is also designed in this phase.
- Implementation/ Coding: In this phase, the actual coding of the system begins. Developers start building the system module by module. For example, they might start with the Inventory Module, implementing features like Stock Management, Inventory Tracking, Reorder Point Setting, etc.
- Testing: After a module is developed, it's tested to ensure it works as expected. This includes unit testing, integration testing, system testing, and user acceptance testing. Any bugs or issues found are fixed.
- Deployment: Once all modules are developed and tested, the system is deployed. This could be on-premises or on the cloud, depending on the company's needs.
- Maintenance: After deployment, the system enters the maintenance phase. This includes fixing any issues, adding new features, and improving existing ones. For example, if the company decides they want to add a "Product Identification" feature to the Inventory Module, this would be done in the maintenance phase.

In line with these answers, the software development process will be started with ChatGPT-4. The questions and answers for each stage are given below.

Answer for (Q4): During the Requirement Gathering and Analysis phase for the ERP project, several key activities should be undertaken:



Identify Stakeholders: Identify all the key stakeholders who will be using the ERP system. This could include employees from different departments such as Sales, Finance, Inventory, HR, etc. Their input is crucial in understanding the requirements of the system.

Conduct Interviews and Surveys: Conduct interviews and surveys with the identified stakeholders to understand their needs and expectations from the ERP system. This could include understanding their current challenges, processes, and what features they would like in the new system.

Document Requirements: All the requirements gathered from the stakeholders should be documented in a structured manner. This could be in the form of a Business Requirement Document (BRD) which outlines in detail the objectives of the project, the needs of the stakeholders, and the expected outcomes.

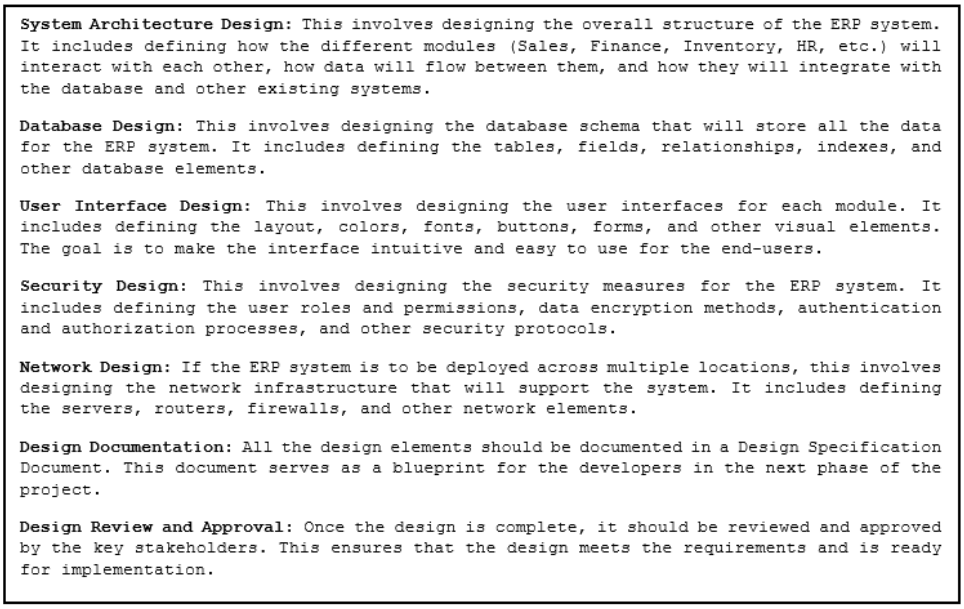
Prioritize Requirements: Not all requirements can be implemented at once. Hence, it's important to prioritize the requirements based on factors such as business value, cost, time, and resource availability.

Analyze and Validate Requirements: The project team should analyze the requirements to ensure they are clear, complete, feasible, and testable. Any ambiguities or conflicts in the requirements should be resolved at this stage.

Create Use Cases: For each requirement, create use cases that describe how the system should behave. This will help in understanding the system's functionality in a real-world scenario.

Approval of Requirements: Once the requirements are documented and analyzed, they should be approved by all the key stakeholders. This ensures everyone is on the same page before moving to the next phase of the project.

Answer for (Q5): During the design phase of the ERP project, several key activities should be undertaken:



System Architecture Design: This involves designing the overall structure of the ERP system. It includes defining how the different modules (Sales, Finance, Inventory, HR, etc.) will interact with each other, how data will flow between them, and how they will integrate with the database and other existing systems.

Database Design: This involves designing the database schema that will store all the data for the ERP system. It includes defining the tables, fields, relationships, indexes, and other database elements.

User Interface Design: This involves designing the user interfaces for each module. It includes defining the layout, colors, fonts, buttons, forms, and other visual elements. The goal is to make the interface intuitive and easy to use for the end-users.

Security Design: This involves designing the security measures for the ERP system. It includes defining the user roles and permissions, data encryption methods, authentication and authorization processes, and other security protocols.

Network Design: If the ERP system is to be deployed across multiple locations, this involves designing the network infrastructure that will support the system. It includes defining the servers, routers, firewalls, and other network elements.

Design Documentation: All the design elements should be documented in a Design Specification Document. This document serves as a blueprint for the developers in the next phase of the project.

Design Review and Approval: Once the design is complete, it should be reviewed and approved by the key stakeholders. This ensures that the design meets the requirements and is ready for implementation.

Answer for (Q6):

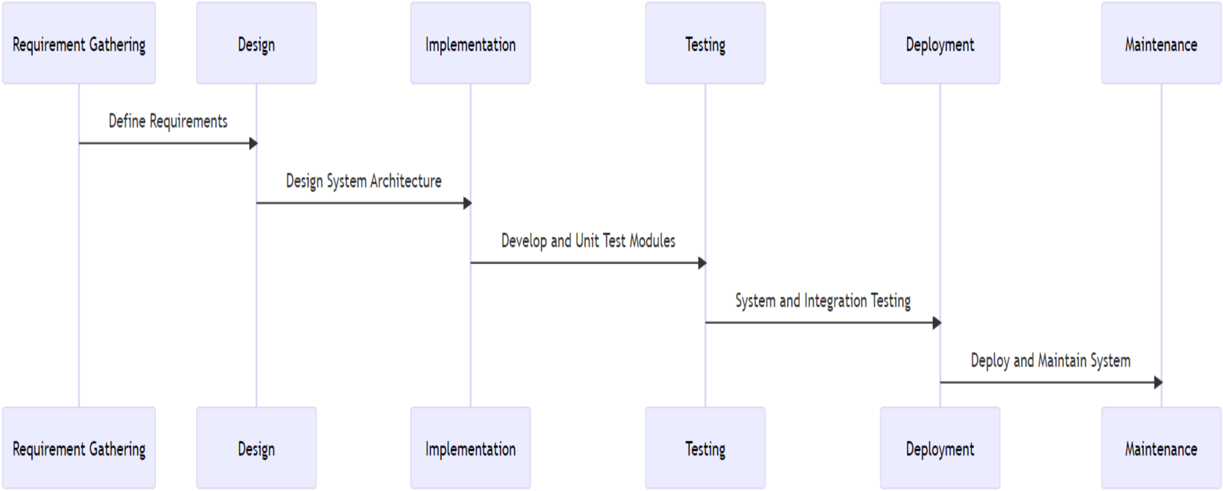


Figure 6. Drawing of the UML diagram for the project to be done.

Fig. 6 shows the UML diagram of the proposed scheme. In the previous questions, the "Show ideas" statement suggested by ChatGPT-4 has been used again to see the suggestions that it would give.

Ideas to improve the diagram:

Add More Details: You could add more details to each phase. For example, during the Requirement Gathering phase, you could include activities like "Identify Stakeholders", "Conduct Interviews", "Document Requirements", etc. Similarly, you could add more details to the other phases as well.

Add Interactions: You could add interactions between the phases. For example, after the Design phase, there could be a feedback loop back to the Requirement Gathering phase in case any changes are needed in the requirements.

Use Different Colors: You could use different colors for each phase to make the diagram more visually appealing and easier to understand. You can use the 'init' directive to customize the theme of an individual diagram. For example, %%{init: {'theme':'forest'}}%%.

Explore the following themes:

default: This is the default theme for all diagrams.

neutral: This theme is great for black and white documents that will be printed.

dark: This theme goes well with dark-colored elements or dark-mode.

forest: This theme contains shades of green.

base: This is the only theme that can be modified. Use this theme as the base for customizations.

To customize the theme of an individual diagram, use the init directive. For example, %%{init: {'theme':'forest'}}%%

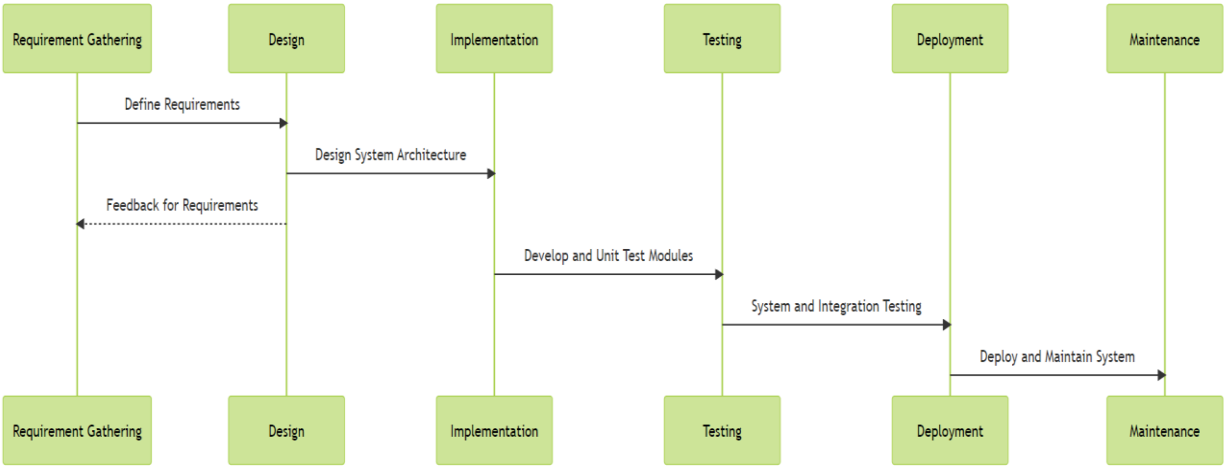
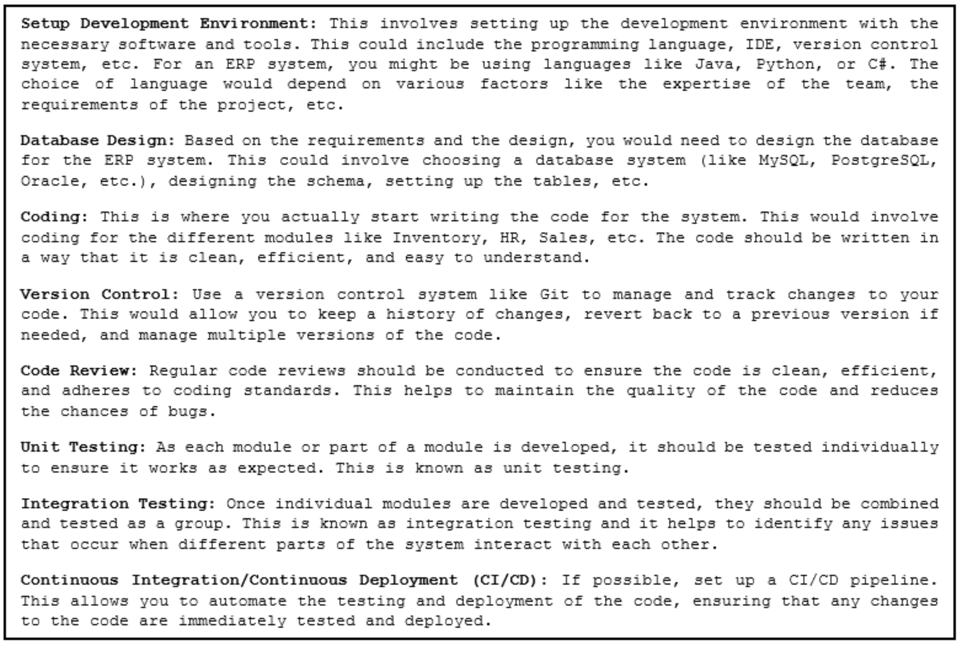


Figure 7. UML diagram edited using the "Forest" theme

The diagram given in Fig 7. was drawn using the "Forest" theme. After drawing the UML diagram, the responses of GTP-4 were taken again for the Implementation/Coding phase.

Answer for (Q7): During the Application/Coding phase of the ERP project, you would be translating the design documents into actual code. Here are some steps that could be followed:



Setup Development Environment: This involves setting up the development environment with the necessary software and tools. This could include the programming language, IDE, version control system, etc. For an ERP system, you might be using languages like Java, Python, or C#. The choice of language would depend on various factors like the expertise of the team, the requirements of the project, etc.

Database Design: Based on the requirements and the design, you would need to design the database for the ERP system. This could involve choosing a database system (like MySQL, PostgreSQL, Oracle, etc.), designing the schema, setting up the tables, etc.

Coding: This is where you actually start writing the code for the system. This would involve coding for the different modules like Inventory, HR, Sales, etc. The code should be written in a way that it is clean, efficient, and easy to understand.

Version Control: Use a version control system like Git to manage and track changes to your code. This would allow you to keep a history of changes, revert back to a previous version if needed, and manage multiple versions of the code.

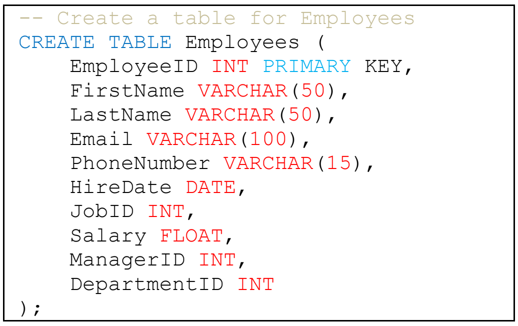
Code Review: Regular code reviews should be conducted to ensure the code is clean, efficient, and adheres to coding standards. This helps to maintain the quality of the code and reduces the chances of bugs.

Unit Testing: As each module or part of a module is developed, it should be tested individually to ensure it works as expected. This is known as unit testing.

Integration Testing: Once individual modules are developed and tested, they should be combined and tested as a group. This is known as integration testing and it helps to identify any issues that occur when different parts of the system interact with each other.

Continuous Integration/Continuous Deployment (CI/CD): If possible, set up a CI/CD pipeline. This allows you to automate the testing and deployment of the code, ensuring that any changes to the code are immediately tested and deployed.

Answer for (Q8): Creating a database for an ERP system is a complex task that involves careful planning and design. Here's a simplified example of how you might create a few tables in an Oracle database for your ERP system using SQL.



```
-- Create a table for Employees
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    PhoneNumber VARCHAR(15),
    HireDate DATE,
    JobID INT,
    Salary FLOAT,
    ManagerID INT,
    DepartmentID INT
);
```

Please note that this is a very basic example and a real-world ERP system would be much more complex. In this example, we have four tables: Employees, Departments, Inventory, and Sales. Each table has a primary key, which uniquely identifies each record in the table. There are also foreign keys which are used to link tables together. For example, the ManagerID in the Departments table would be a foreign key linking to the EmployeeID in the Employees table.

After the database was created, the request to draw the Entity-Relation (ER) diagram was directed as specified. ER diagram generated by ChatGPT after this request is shown in Fig.8.

Answer for (Q9):

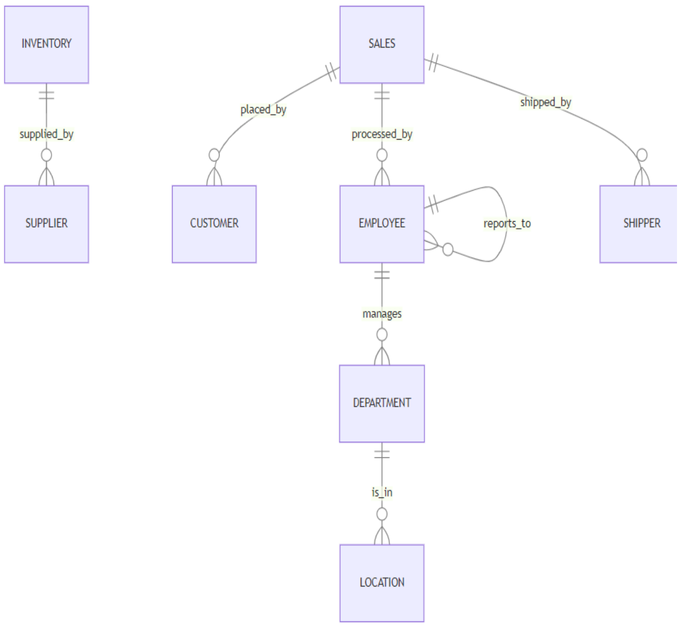


Figure 8. ER diagram of the database.

Answer for (Q10): Creating Python code for an entire ERP system is beyond the scope of this platform, as it would involve thousands of lines of code and a deep understanding of your specific business requirements. However, I can provide you with some basic examples of how you might use Python to interact with the SQL Server database we created earlier. Here's an example of how you might connect to the database:

```
import pyodbc
# Create a connection string
conn_str = (
    r'DRIVER={ODBC Driver 17 for SQL Server};'
    r'SERVER=your_server;'
    r'DATABASE=your_database;'
    r'UID=your_username;'
    r'PWD=your_password;'
)
# Create a connection
conn = pyodbc.connect(conn_str)
```

Once you're connected, you can execute SQL queries. Here's an example of how you might insert a new employee into the Employees table:

```
# Create a cursor
cursor = conn.cursor()
# SQL query
sql = """
INSERT INTO Employees (EmployeeID,
FirstName, LastName, Email, PhoneNumber,
HireDate, JobID, Salary, ManagerID,
DepartmentID)
VALUES (?,?, ?, ?, ?, ?, ?, ?, ?, ?)
"""
# Values
values = (1, 'John', 'Doe',
'john.doe@example.com', '555-555-5555',
'2023-01-01', 1, 50000, None, 1)
# Execute the query
cursor.execute(sql, values)
# Commit the transaction
conn.commit()
```

You can also retrieve data. Here's how you might get all employees:

```
# SQL query
sql = "SELECT * FROM Employees"
# Execute the query
cursor.execute(sql)
# Fetch all rows
rows = cursor.fetchall()
# Print rows
for row in rows:
    print(row)
```

These are very basic examples. A real-world ERP system would have much more complex interactions with the database, and would likely involve a web interface or other user interface, business logic, error handling, security measures, and more.

The code blocks given above contain some of the answers given by ChatGPT-4. it has the competence to realize the database connection of the ERP system with basic system codes. However, it needs an expert to do the necessary coding for the interface design. He was repeatedly asked in different ways to create a system in Python. But his answer to all of them was that he could not create such coding. Each time ChatGPT-4 provided Python code that provided a connection to the database.

After the answers were received in the Implementation/Coding phase, questions were asked to ChatGPT-4 for the next phase, the testing phase (Q11). The answers given by ChatGPT are presented below.

Answer for (Q11): The testing phase is a critical part of the software development lifecycle, especially for a complex system like an ERP. Here are the steps that can be followed during the testing phase of the ERP project:

- Unit Testing: This is the first level of testing where individual components of the software are tested. The purpose is to validate that each unit of the software performs as designed. In the context of an ERP system, this could mean testing individual functions or methods in the code.
- Integration Testing: After unit testing, integration testing is performed. This involves combining individual units and testing them as a group. The purpose of this level of testing is to detect any faults in the interaction between integrated units.
- System Testing: In this phase, the entire system is tested as a whole. The purpose of this testing is to evaluate the system's compliance with the specified requirements.
- User Acceptance Testing (UAT): This is the final phase of testing, where the software is tested in the "real world" by the intended audience. UAT can be done by the end-users of the software or subject matter experts in the business domain.
- Performance Testing: This is done to ensure that the system performs effectively under varying loads and volumes. It includes load testing (testing under expected load) and stress testing (testing under extreme loads).
- Security Testing: This is done to check how the ERP software protects against threats from external entities and maintains data integrity.
- Regression Testing: This is done after modifications such as enhancements or defect fixes to ensure that the changes haven't introduced new faults.

- ➢ Compatibility Testing: This is done to ensure the ERP system can effectively run in different environments – different operating systems, browser types, hardware configurations, etc.
- ➢ Usability Testing: This is done to ensure that the ERP system is user-friendly and easy to use.

Based on the test suggestions given by ChatGPT-4, one of the tests was selected and Q12 question was asked to conduct a sample study.

Answer for (Q12): Performance testing is a crucial part of ensuring that an ERP system can handle the expected load and perform optimally under different conditions. Here's a simplified example of how you might conduct performance testing on an ERP system using a tool like Apache JMeter, which is an open-source software designed to load test functional behavior and measure performance.

Please note that this is a simplified example and actual performance testing can be quite complex depending on the specific requirements and architecture of your ERP system.

Identify Key Scenarios to Test: Identify the most common and critical operations that users will perform on your ERP system. This could include operations like logging in, accessing and updating records, generating reports, etc.

Create Test Plan: In JMeter, you would create a new Test Plan and add Thread Groups to it. Each Thread Group represents a user scenario that you want to test. For example, you might have one Thread Group for logging in, another for accessing records, etc.

Configure Thread Properties: For each Thread Group, you would configure the number of threads (simulated users), the ramp-up period (how long to take to "ramp up" to the full number of users), and the number of iterations (how many times to execute the test).

Add and Configure Samplers: Samplers tell JMeter to send specific types of requests to a server. You would add HTTP Request samplers to your Thread Groups and configure them to send requests that simulate the user operations you're testing.

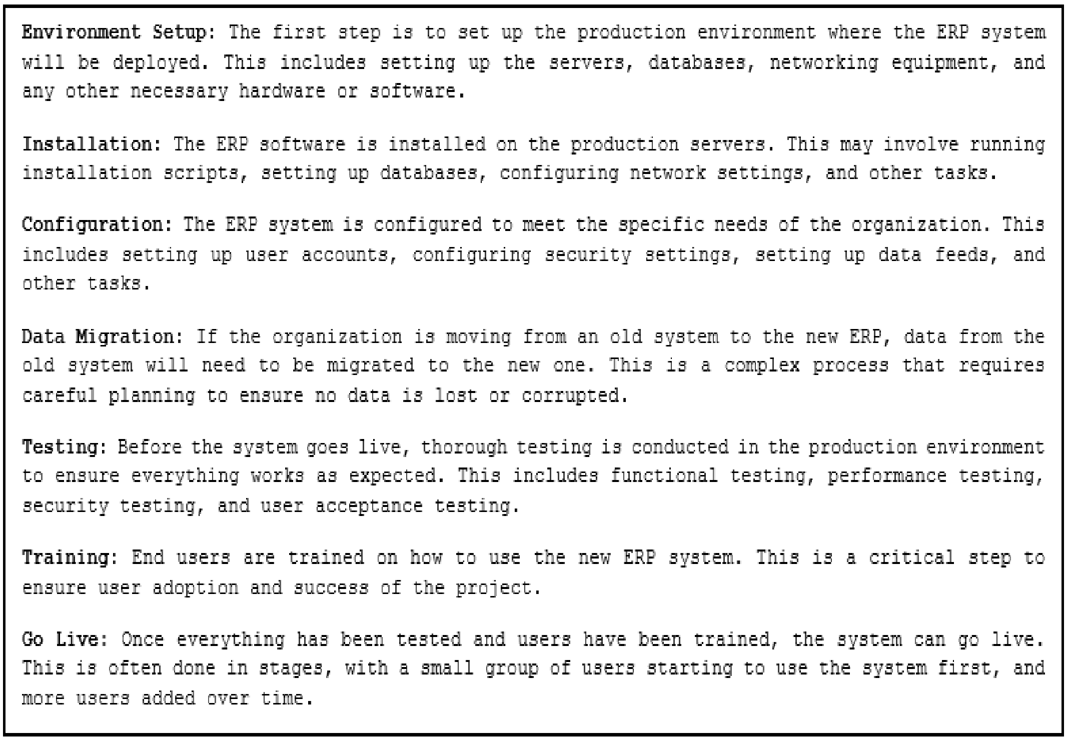
Add and Configure Listeners: Listeners provide access to the data gathered by JMeter about the test cases as a Sampler component is processed. For example, you might add a Summary Report listener to see a table of aggregated results for all samplers, or a Graph Results listener to see a visual graph of the performance results.

Run Tests and Analyze Results: Once everything is set up, you would run your tests and then analyze the results to identify any performance bottlenecks or issues.

As a result of the answers obtained from ChatGPT-4, it is foreseen that it can also provide ideas in the test phase. The answer to Q13 for the last two steps, distribution and maintenance, is as follows.

Answer for (Q13): Sure, deployment and maintenance are critical phases in the software development lifecycle, especially for complex systems like an ERP. Here's a general outline of the steps involved:

Deployment:



Environment Setup: The first step is to set up the production environment where the ERP system will be deployed. This includes setting up the servers, databases, networking equipment, and any other necessary hardware or software.

Installation: The ERP software is installed on the production servers. This may involve running installation scripts, setting up databases, configuring network settings, and other tasks.

Configuration: The ERP system is configured to meet the specific needs of the organization. This includes setting up user accounts, configuring security settings, setting up data feeds, and other tasks.

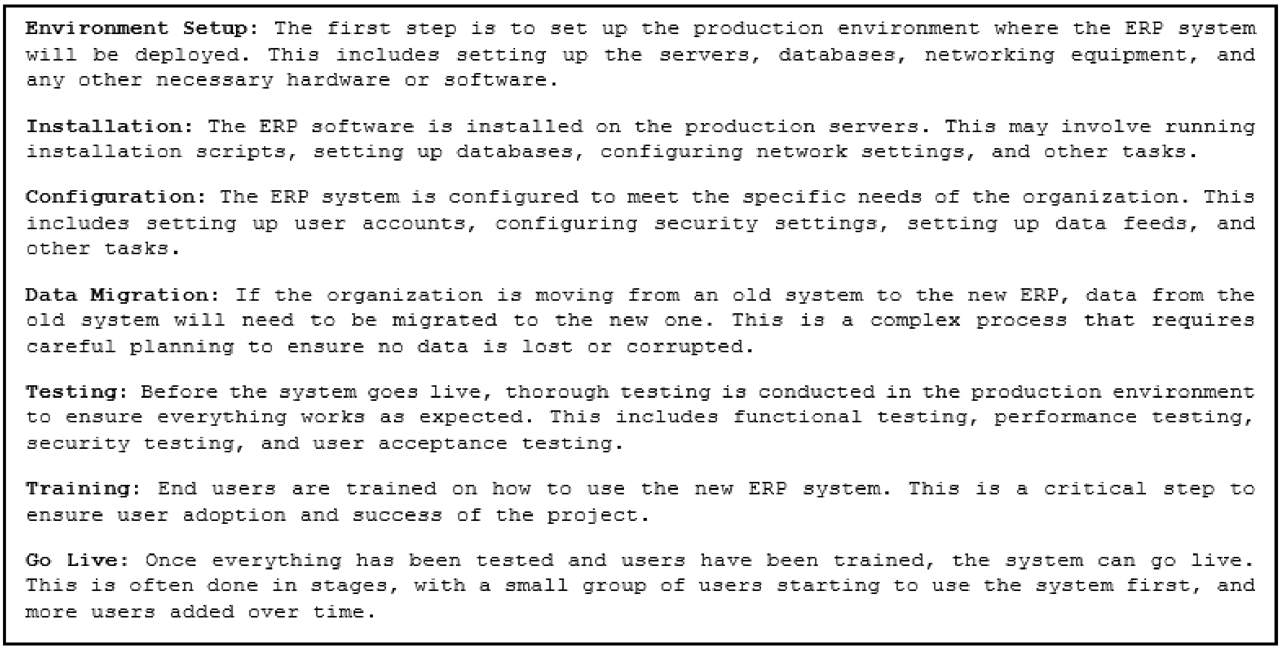
Data Migration: If the organization is moving from an old system to the new ERP, data from the old system will need to be migrated to the new one. This is a complex process that requires careful planning to ensure no data is lost or corrupted.

Testing: Before the system goes live, thorough testing is conducted in the production environment to ensure everything works as expected. This includes functional testing, performance testing, security testing, and user acceptance testing.

Training: End users are trained on how to use the new ERP system. This is a critical step to ensure user adoption and success of the project.

Go Live: Once everything has been tested and users have been trained, the system can go live. This is often done in stages, with a small group of users starting to use the system first, and more users added over time.

Maintenance:



Environment Setup: The first step is to set up the production environment where the ERP system will be deployed. This includes setting up the servers, databases, networking equipment, and any other necessary hardware or software.

Installation: The ERP software is installed on the production servers. This may involve running installation scripts, setting up databases, configuring network settings, and other tasks.

Configuration: The ERP system is configured to meet the specific needs of the organization. This includes setting up user accounts, configuring security settings, setting up data feeds, and other tasks.

Data Migration: If the organization is moving from an old system to the new ERP, data from the old system will need to be migrated to the new one. This is a complex process that requires careful planning to ensure no data is lost or corrupted.

Testing: Before the system goes live, thorough testing is conducted in the production environment to ensure everything works as expected. This includes functional testing, performance testing, security testing, and user acceptance testing.

Training: End users are trained on how to use the new ERP system. This is a critical step to ensure user adoption and success of the project.

Go Live: Once everything has been tested and users have been trained, the system can go live. This is often done in stages, with a small group of users starting to use the system first, and more users added over time.

4. DISCUSSION

Software development processes are one of the most fundamental elements that ensure the successful completion of the project given in the environment where engineers work. In order to accurately determine the inputs, outputs and requirements of the given project, each step of the software development life cycle must be worked carefully and diligently. By assigning employees to the necessary parts of the cycle according to their competencies, it ensures that the project emerges efficiently and quickly. The software development process can be a difficult process for engineers until they get to the coding part. AI multilingual chatbots have reached a level that can give ideas in the field of software development. These robots, which many researchers now include in their academic or non-academic studies, make the work of researchers easier. Despite various security vulnerabilities, many questions can be answered. Apart from the most popular ChatGPT, there are also robots such as YouChat, BingChat, JasperChat, Character.AI.

After the latest updates of ChatGPT, its use in academia has become more widespread. The fact that it can provide visual results such as graphics, tables, and pictures are important factors that attract the attention of users. Some studies in the literature reveal the positive and negative aspects of ChatGPT, ethical violations, and various plagiarism possibilities [4-6]. Since ChatGPT is also utilized while conducting these studies, it is actually an example of intelligence whose active learning process is constantly ongoing.

In this study, we investigated what role ChatGPT can play in a software development project. It was asked to realize an ERP project that constitutes the financial information systems of corporate companies by applying all stages of the software life cycle. All questions were planned step by step in accordance with the software life cycle and asked to ChatGPT-4 respectively. In addition to the planned questions, new questions were added to the question pool by taking into account the suggestions given by ChatGPT-4. As a result of the study, it is seen that ChatGPT-4 can be utilized as a very good guide in all steps except Implementation&Coding and Test phases. The ideas obtained from the ChatGPT-4 in the requirements determination and Analysis, Design, Deployment and Maintenance phases are in a structure that will make the software developer's job much easier in this process. If the right questions are directed to ChatGPT-4 at the right points, the completion time and efficiency of the project will increase. The situation that can be stated as a disadvantage during the service received from ChatGPT-4 is that it is not fully efficient in coding. ChatGPT-4, who could provide assistance in coding up to a certain point, stated that the codes should be written under the supervision of a software engineer after a certain point. The codes given by ChatGPT-4 included the creation of the database system required in the software process and how the database connection can be made in Python in the codes to be used afterward. However, he stated that he could not help with the actual Python coding required to complete the project. Due to this shortcoming, the answers he gave in the Test phase could not go beyond suggestions. Although this may seem like a disadvantage for ChatGPT, it can be considered as an advantage from the point of view of the need for software engineers. When the limitations of working with ChatGPT are considered, it is seen that the most basic problems are plagiarism and security vulnerabilities. If questions about a project planned to be implemented are directed to ChatGPT by a third party, it may be possible for others to access the project details. In academic-based research, the literature sources given may be incorrect or non-existent. Although the plagiarism issue has been corrected a little more in ChatGPT-4, it still has deficiencies. The results obtained with the help of ChatGPT should always be checked by the user.

It is envisaged that a software development specialist can facilitate the processes in the initial stages by getting help from a multilingual robot such as ChatGPT while developing a project. Due to the lack of performance in the Implementation & Coding step, it is necessary to complete the software development process under the supervision of a software developer. Due to ChatGPT's lack of performance in this area and its many disadvantages in terms of security, the need for software engineers continues.

In future studies, it is planned to create a software development project with the help of AI-supported robots other than ChatGPT. It is considered to design a working system by completing the missing points of AI systems by a software development specialist. For both ideas, the use of AI-supported software is inevitable.

5. CONCLUSION

The use of AI systems in software development processes has positive effects such as automating the process, reducing errors, and increasing productivity. For this purpose, there are many AI tools available online in the market. ChatGPT is making a difference among robots with the recent emergence of ChatGPT-4 version. In this paper, research questions on how to create a software development process using ChatGPT-4 and ChatGPT's answers to them are presented. As a result of the study, the answers obtained from ChatGPT-4 from the requirements determination, analysis, and design stages are logical and can be easily applied in such a process. Various deficiencies were encountered in the Implementation & Coding phase. One of the most important of these is that the coding part was not fully completed and it was suggested that a software engineer should help after a certain point. The software development process with ChatGPT-4 has progressed in a way to meet expectations despite the shortcomings in the coding and testing phase. The clearest conclusion that can be drawn from this study is that an expert software developer can complete the software development process with minimum errors and maximum efficiency with an AI robot.

REFERENCES

[1] I. Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering. Pearson, 2019.

[2] B. W. Boehm, “A spiral model of software development and enhancement,” Computer, vol. 21, no. 5, pp. 61–72, May 1988, doi: 10.1109/2.59.

[3] M. U. Cheema and Q. Zearlish, “The Choice of Project Management Software by Project Managers; with the Moderating Impact of Top Management Support,” vol. 2, no. 1.

[4] E. Nascimento, A. Nguyen-Duc, I. Sundbø, and T. Conte, “Software engineering for artificial intelligence and machine learning software: A systematic literature review”.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners”.

[6] A. Ahmad, M. Waseem, P. Liang, M. Fahmideh, M. S. Aktar, and T. Mikkonen, “Towards Human-Bot Collaborative Software Architecting with ChatGPT,” in Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, Oulu Finland: ACM, Jun. 2023, pp. 279–285. doi: 10.1145/3593434.3593468.

[7] O. Katar, D. Özkan, G. -3, Ö. Yildirim, and U. R. Acharya, “Evaluation of GPT-3 AI Language Model in Research Paper Writing,” Turkish Journal of Science and Technology, Jun. 2023, doi: 10.55525/tjst.1272369.

[8] M. A. Akbar, A. A. Khan, and P. Liang, “Ethical Aspects of ChatGPT in Software Engineering Research.” arXiv, Jun. 13, 2023. Accessed: Jul. 14, 2023. [Online]. Available: http://arxiv.org/abs/2306.07557

[9] M. Fraiwan and N. Khasawneh, “A Review of ChatGPT Applications in Education, Marketing, Software Engineering, and Healthcare: Benefits, Drawbacks, and Research Directions”.

[10] R. A. Carter, A. I. Anton, A. Dagnino, and L. Williams, “Evolving beyond requirements creep: a risk-based evolutionary prototyping model,” in Proceedings Fifth IEEE International Symposium on Requirements Engineering, Aug. 2001, pp. 94–101. doi: 10.1109/ISRE.2001.948548.

[11] A. C. Nelson and J. T. C. Teng, “Do systems development methodologies and CASE tools decrease stress among systems analysts?,” Behaviour & Information Technology, vol. 19, no. 4, pp. 307–313, Jan. 2000, doi: 10.1080/01449290050086417.

[12] W. Model, “Waterfall model,” Luettavissa: http://www. waterfall-model. com/. Luettu, vol. 3, 2015.

[13] K. Rasheed, M. Imran, M. Noman, and M. Iqbal, “A Study On Traditional And Evolutionary Software Development Models,” vol. 6, no. 07, 2017.

[14] G. Kumar and P. K. Bhatia, “Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies,” in 2014 Fourth International Conference on Advanced Computing & Communication Technologies, Feb. 2014, pp. 189–196. doi: 10.1109/ACCT.2014.73.

[15] G. Regulwar, P. Jawandhiya, V. Gulhane, and R. Tugnayat, “Variations in V Model for Software Development,” Jan. 2021.

[16] A. K. M. Z. Islam and Dr. A. Ferworn, “A Comparison between Agile and Traditional Software Development Methodologies,” GJCST, pp. 7–42, Dec. 2020, doi: 10.34257/GJCSTCVOL20IS2PG7.

[17] T. Clement, N. Kemmerzell, M. Abdelaal, and M. Amberg, “XAIR: A Systematic Metareview of Explainable AI (XAI) Aligned to the Software Development Process,” Machine Learning and Knowledge Extraction, vol. 5, no. 1, Art. no. 1, Mar. 2023, doi: 10.3390/make5010006.

[18] S. Russell, Artificial Intelligence: A Modern Approach, eBook, Global Edition. Pearson Education, Limited, 2016.

[19] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, “What do we need to build explainable AI systems for the medical domain?” arXiv, Dec. 28, 2017. doi: 10.48550/arXiv.1712.09923.

[20] F. A. Batarseh, R. Mohod, A. Kumar, and J. Bui, “10 - The application of artificial intelligence in software engineering: a review challenging conventional wisdom,” in Data Democracy, F. A. Batarseh and R. Yang, Eds., Academic Press, 2020, pp. 179–232. doi: 10.1016/B978-0-12-818366-3.00010-1.

[21] S. Vemuri, S. Chala, and M. Fathi, “Automated use case diagram generation from textual user requirement documents,” in 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Apr. 2017, pp. 1–4. doi: 10.1109/CCECE.2017.7946792.

[22] M. Latinovic and V. Pammer-Schindler, “Automation and Artificial Intelligence in Software Engineering: Experiences, Challenges, and Opportunities: The 54th Hawaii International Conference on System Sciences,” Proceedings of the 54th Annual Hawaii International Conference on System Sciences, HICSS 2021, pp. 146–155, 2021, doi: 10.24251/HICSS.2021.017.

[23] A. Rafael Lenz, A. Pozo, and S. Regina Vergilio, “Linking software testing results with a machine learning approach,” Engineering Applications of Artificial Intelligence, vol. 26, no. 5, pp. 1631–1640, May 2013, doi: 10.1016/j.engappai.2013.01.008.

[24] https://facebook.com/saiftheboss7, “The Future of AI in Software Development in 2023 and Beyond - Appsero,” Mar. 06, 2023. https://appsero.com/user-guide/ai-in-software-development/ (accessed Jul. 13, 2023).

[25] H. Klaus, M. Rosemann, and G. G. Gable, “What is ERP?,” Information Systems Frontiers, vol. 2, no. 2, pp. 141–162, Aug. 2000, doi: 10.1023/A:1026543906354.

BIOGRAPHIES

Zeynep Özpolat obtained her BSc degree from Fırat University, Department of Mathematics in 2012. She completed her master's degree at Fırat University, Department of Mathematics. In 2018, she started her doctorate education at Fırat University, Software Engineering. The doctoral process is still ongoing. Research areas; data science, quantum machine learning, artificial intelligence, software engineering.

Özal Yıldırım received his doctorate. I am a graduate of Fırat University Electrical and Electronics Engineering. He is currently working as an Associate Professor in the Software Engineering Department at Fırat University. He has published over 60 articles in international peer-reviewed journals and conference proceedings. As a result of the research carried out on seven million researchers under the coordination of Stanford University, he was included in the "World's Most Influential Scientists" list. His main research interests include deep learning and medical signal and image processing.

Murat Karabatak received his doctorate. He graduated from Fırat University Electrical and Electronics Engineering. He works as a professor in the Department of Software Engineering at Fırat University. His main research areas are data mining, software engineering, database systems and artificial intelligence.