World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

A guide to the methodology and system analysis section of a computer science project

Obinna Ogbonnia Otuu * and Felix Chukwuma Aguboshim

Department of Computer Science, Federal Polytechnic Oko, Anambra State, Nigeria.

World Journal of Advanced Research and Reviews, 2023, 19(02), 322–339

Publication history: Received on 22 June 2023; revised on 29 July 2023; accepted on 01 August 2023

Article DOI: https://doi.org/10.30574/wjarr.2023.19.2.1547

Abstract

The methodology section of every research work remains essential to the presentation of research findings. Methodology supports research findings by explaining the research techniques applied and creating a roadmap of how a researcher reached to a conclusion. Knowing how to structure the methodology section has been a challenge to undergraduate computing students. This guide will address these challenge and enable computing students to argue for the methods they have chosen and demonstrate how they directly addressed the questions they posed during their study. In this guide, we define methodology, discuss its significance, describe popular research techniques, and provide instructions on how to develop one with examples.

Keywords: Methodology; System Analysis; Guide; Computing; Students

1. Introduction

According to Mishra and Alok (2022), methodology is a detailed description of a research process that you choose to conduct your research as a scientist or a researcher. In other words, it's a contextual framework that presents a logical path for answering questions that you raise at the beginning of your thesis or paper. Typically, the methodology makes up its own section in a paper, in which you can describe your method for gathering, grouping and analysing observational, experimental, simulation and derived data (Bougie & Sekaran, 2019).

Sharing your methodology gives legitimacy to your research. This is especially important if you're conducting scientific or academic research. In this case, your reader expects you to follow common practices that can lead you to a reliable, logical and coherent conclusion. It's also critical that your methodology is repeatable, meaning anyone who uses the same methods can reach the same conclusions you reached. Methodology can be qualitative, quantitative or mixed, to mention but a few (Jensen, 2020). In computer science, a student is expected to follow the information system development methodology (ISDM), which could still have a touch of any of quantitative, qualitative or mixed-method. This ISDM follows a process from planning, through analysis, implementation to cessation; a process called Software development life circle (SDLC).

Example of these methodologies followed are user-centred design, participatory design, structured system analysis and design methodology, object oriented design and methodology etc. Hence, when you are asked your research methodology, be bold to say that it is the SSADM, User-centred design or the others, depending on your project topic.

1.1. System analysis

System analysis refers to the process of studying and understanding a system to identify its components, interactions, and functionalities (Gandhmal & Kumar, 2019). It involves a thorough examination of the system's requirements, processes, data, and user interactions to ensure effective system design and development. It helps in understanding

* Corresponding author: Otuu Obinna Ogbonnia

user needs, modeling system components and interactions, and assessing the feasibility of the proposed system. Effective system analysis ensures that the resulting system meets user expectations, optimizes processes, and delivers value to stakeholders. Here are key steps involved in system analysis (Tilley, 2019):

1.1.1. Requirement Gathering

The system analysis process begins with gathering requirements. This involves interacting with stakeholders, such as users, clients, and subject matter experts, to Identify their needs, expectations, and desired outcomes from the system. Requirements are typically documented and prioritized to serve as a foundation for system design.

1.1.2. System Modeling

System modeling techniques are used to represent the structure and behavior of the system. This includes creating diagrams, such as use case diagrams, activity diagrams, and sequence diagrams, to visualize how different components interact and how data flows within the system. These models help identify the system's functional and non-functional requirements.

1.1.3. Data Analysis

Analyzing the data requirements of the system is a crucial step in system analysis. This involves identifying the types of data needed, how it should be organized, stored, and processed, and any data dependencies or relationships within the system. Data analysis helps define data models, such as entity-relationship diagrams or data flow diagrams, to capture the system's data structure.

1.1.4. Process Analysis

Process analysis involves examining the activities, tasks, and workflows within the system. It aims to understand how data is processed, transformed, and stored at different stages of the system. This analysis helps identify opportunities for process optimization, efficiency improvements, and automation.

1.1.5. User Interaction Analysis

Understanding how users interact with the system is essential for effective system design. User interaction analysis involves studying user interfaces, usability requirements, and user experience expectations. This analysis helps identify user roles, tasks, and preferences, ensuring that the system is intuitive, user-friendly, and aligned with user needs.

1.1.6. Feasibility Assessment

System analysis includes assessing the feasibility of implementing the proposed system. This involves considering technical, economic, and operational aspects to determine if the system is practical, cost-effective, and sustainable. Feasibility assessment helps identify potential risks, constraints, and alternatives that may impact the system's viability.

1.1.7. Documentation

Throughout the system analysis process, it is crucial to document findings, requirements, and analysis results. Documentation ensures that system requirements and design decisions are captured, shared, and validated by stakeholders. Clear documentation helps in subsequent system design, development, and maintenance phases.

2. Guide to writing methodology and system analysis

2.1. Step 1: provide a brief overview of the project and its objectives.

A project overview is a proper description of the project with the steps to be followed to achieve the goals and objectives (García-Peñalvo, 2019). Below is an example of a project overview using some selected computing project topics.

2.1.1. Project Overview for an intelligent tutoring system for novice programmers

This project develops an intelligent tutoring system (ITS) for teaching programming concepts to novice learners. The system aims to enhance the learning experience and facilitate the acquisition of programming skills in an interactive and personalized manner. Traditional programming education often involves passive lectures or static textbooks, which can be challenging for beginners to grasp complex concepts effectively. Therefore, our project focuses on creating an

ITS that offers adaptive learning experiences, provides personalized feedback, and fosters engagement and motivation among learners.

2.2. Objectives

2.2.1. Develop an interactive and user-friendly ITS platform

We aim to create a web-based platform that enables students to learn programming through hands-on exercises, interactive coding challenges, and real-time feedback. The platform should be accessible to learners of varying skill levels and provide a seamless user experience.

2.2.2. Design and implement adaptive learning algorithms

Our project aims to incorporate machine learning techniques to adapt the learning materials and instructional strategies based on the individual learner's progress, strengths, and weaknesses. The system will analyze learner performance data to dynamically adjust the difficulty level of exercises and provide personalized recommendations for further study.

2.2.3. Evaluate the effectiveness of the ITS

We will conduct a comprehensive evaluation of the developed ITS to assess its impact on learning outcomes, engagement, and user satisfaction. We will compare the performance of students who use the ITS with those who follow traditional programming education approaches to measure the system's effectiveness and identify areas for improvement.

2.2.4. Expectations

By developing an intelligent tutoring system with adaptive learning capabilities, we expect our project to contribute to the field of computer science education by providing an innovative and effective learning tool for programming novices. The project aims to address the challenges associated with traditional instructional methods and offer a personalized and engaging learning experience to enhance students' programming skills and knowledge.

2.3. Step 2: clearly state the research questions or goals of the system analysis.

Research Design or System Analysis Approach

According to Sileyew (2019), a research design is a strategy for answering your research question using empirical data. A well-planned research design helps ensure that your methods match your research objectives and that you use the right kind of analysis for your data (Borgianni & Maccioni, 2020). Below is an example following our ITS project topic.

For our project, we adopted a user-centred research design that combines qualitative and quantitative approaches to address our research questions and system analysis goals effectively. These approaches allow us to gather rich and diverse data, gain in-depth insights, and validate our findings through statistical analysis.

2.4. Qualitative Phase

In the qualitative phase, we conducted semi-structured interviews with experienced software developers and educators to gain a deeper understanding of the challenges faced by novice programmers and the pedagogical strategies that could be incorporated into the intelligent tutoring system. We used purposive sampling to select participants with diverse backgrounds and expertise, ensuring a comprehensive exploration of the research topic. The interviews were audio-recorded and transcribed verbatim for thematic analysis, using techniques such as open coding and constant comparison to identify key themes and patterns in the data.

2.5. Quantitative Phase

In the quantitative phase, we collected data from a large sample of novice learners who used the intelligent tutoring system over a semester. We designed a pre- and post-test questionnaire to assess the students' programming knowledge and self-efficacy. The questionnaire consisted of multiple-choice questions, Likert-scale items, and open-ended prompts. We administered the questionnaire before and after the semester to measure the effectiveness of the ITS in improving programming skills and self-confidence. Additionally, we collected log data from the system, including the number of exercises completed, time spent on each task, and error rates, to analyze learner behavior and performance.

2.6. Data Integration and Analysis

To integrate the qualitative and quantitative data, we used a convergent parallel design approach. This involved analyzing the interview transcripts for emerging themes and then connecting these themes to the quantitative findings to provide a comprehensive understanding of the impact of the intelligent tutoring system on programming learning outcomes. We employed a triangulation approach to identify areas of convergence and divergence between the qualitative and quantitative data, enhancing the validity and reliability of our results.

Statistical analysis was conducted on the quantitative data using descriptive statistics, paired t-tests, and regression analysis to examine the effectiveness of the ITS. The qualitative data were analyzed using thematic analysis, and findings were supported by verbatim quotes from the interview transcripts to ensure transparency and credibility.

Justification: By employing a mixed-method research design, we could capture a holistic view of the research topic and gain insights from multiple perspectives. The qualitative phase allowed us to explore the nuanced experiences and perceptions of experts, while the quantitative phase provided statistical evidence to support our claims and measure the impact of the intelligent tutoring system on programming learning outcomes.

2.7. Step 3: explain the research design or system analysis approach adopted.

This could be experimental, survey-based, case study, or a specific methodology/framework. You are meant to choose only one except in very rare cases when you combined two or more. Remember that no method is best as earlier stated in this guide, however some are suitable for a particular project, while some are not. Therefore, don't just choose an approach or method because you saw your classmates using it in their work. Also, you must give justification (good reason) for choosing a particular approach For instance, a wrong method for a topic can slow the time of system development and delay software delivery. There are tens of research designs or system analysis approaches, however for this guide, we shall discuss on five. For example 1, we will use user-centred design approach on the ITS design. Other examples shall show how structured system analysis and design methodology (SSADM), Prototyping Expert system methodology, object oriented design methodology can be used with different or related topics. These examples are below.

2.8. Example 1 User-centred methodology

In the ITS project, we employed a user-centred design approach combined with a comparative analysis to guide the development of the website (Wright & McCarthy, 2022). This approach allowed us to gather user feedback, understand their needs and preferences, and evaluate the performance of our system against existing competitors.

Based on the gathered information, we created user personas and user scenarios to represent different types of users and their typical interactions with the app. These personas helped us empathize with users and make design decisions that catered to their specific needs.

To iteratively develop and refine the application, we conducted user testing sessions where participants interacted with prototypes or early versions of the mobile app. We observed their interactions, recorded their feedback, and made iterative improvements to the user interface, navigation flow, and feature set based on their input. This iterative feedback loop allowed us to incorporate user preferences and expectations throughout the development process.

2.9. Comparative Analysis

In addition to the user-centered design approach, we conducted a comparative analysis of related applications. We carefully selected a set of competitor applications that were popular and well-established in the market. We thoroughly examined their features, user interface design, performance, and customer reviews to identify areas of improvement and innovation.

Justification: We chose the user-centred design because it has a comparative analysis process, which helps to understand the strengths and weaknesses of existing solutions and identify opportunities for differentiation. It also enables researchers to benchmark application's performance against established competitors and set targets for key performance indicators, such as app loading time, responsiveness, and ease of use.

2.9.1. Example 2 Structured System Analysis and Design Methodology (SSADM):

There are topics which need structured system analysis and design methodology (SSADM), than the user centred design and others. The SSADM methodology provides a systematic framework for analysing user requirements, designing the system architecture, and implementing the desired functionalities (Ananda et al., 2022), hence is best for dynamic applications. Hence, the topic "Design and implementation of an e-commerce mobile application system" will need SSADM to be easily delivered.

In the above topic, we will employ the Structured Systems Analysis and Design Methodology (SSADM) to guide the development as listed in the below steps.

2.10. System Analysis

We began by conducting a thorough analysis of the current e-commerce landscape and the specific needs of our target users. We used techniques such as interviews, surveys, and data gathering to understand the requirements, expectations, and pain points of both customers and merchants.

Using SSADM, we modeled the existing system and identified areas of improvement. We created data flow diagrams (DFDs) to visualize the flow of information within the system, entity relationship diagrams (ERDs) to capture the relationships between different data entities, and process specifications to document the system's functions and processes. So below is the ERD for the ITS system. The DFD can be found in section 3.5.9. Meanwhile, your DFD and ERD must adhere to the flow of your application. So, don't copy a DFD or ERD because someone has used the diagram in a related topic, your system if run, may behave otherwise.

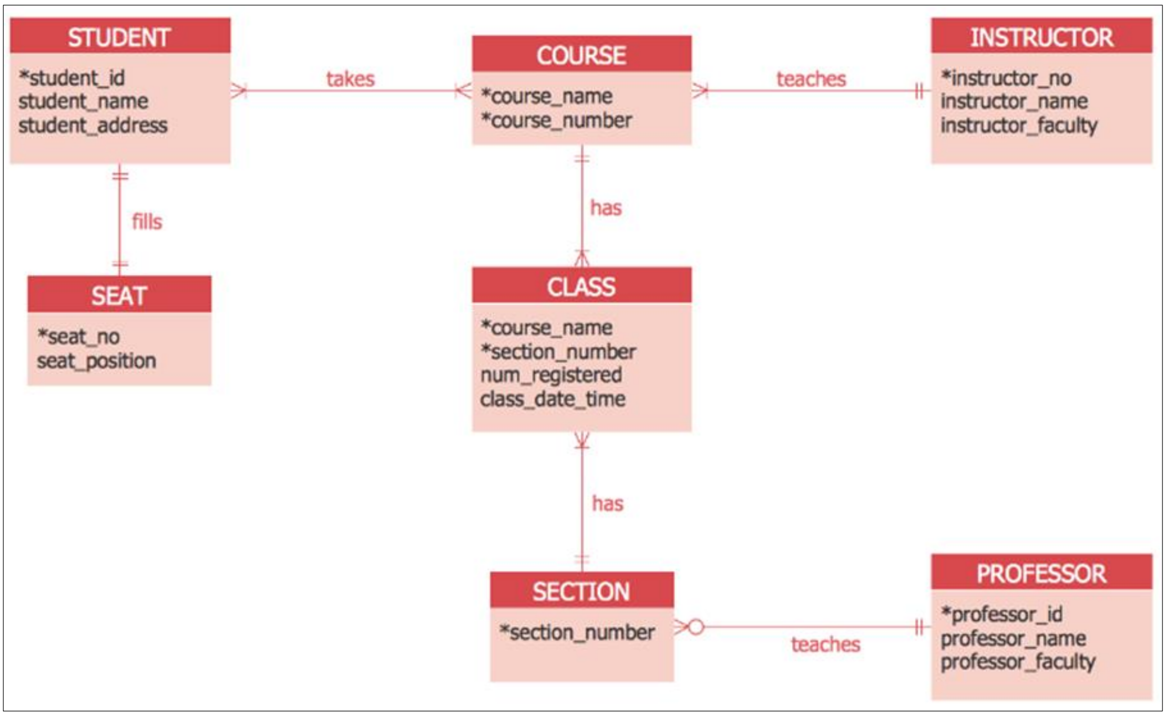

Figure 1 Entity relationship diagram. Source: fordfiringorder.com

Based on the analysis, we derived a set of functional and non-functional requirements that served as the foundation for the design phase.

2.11. System Design

In the system design phase, we focused on creating a modular and scalable architecture for the e-commerce mobile application. We used techniques such as structured walkthroughs, prototyping, and user feedback sessions to refine the design.

Using SSADM, we developed a logical data model that defined the system's data entities, attributes, and relationships. We also created a structure chart to depict the modular design of the system, illustrating the main components, subsystems, and their interactions.

Additionally, we designed the user interface (UI) and user experience (UX) aspects of the application. This involved creating wireframes, storyboards, and interactive prototypes to visualize the app's screens, navigation flow, and overall user interaction.

Once the design was finalized, we conducted a comprehensive feasibility study to assess the technical, operational, and economic viability of the proposed system.

2.12. System Implementation

In the implementation phase, we transformed the design specifications into a functional e-commerce mobile application. We followed industry-standard coding practices and utilized appropriate development frameworks and programming languages.

During implementation, we adopted an iterative and incremental approach, where we developed and tested system functionalities in small increments. This allowed us to identify and address any issues early on, ensuring a smoother development process.

2.13. System Testing and Deployment

After the implementation, we performed a series of rigorous testing activities to verify the system's functionality, performance, and security. We conducted unit testing, integration testing, and system testing to ensure that all components work seamlessly together.

Once the testing phase was completed, we deployed the e-commerce mobile application to a production environment, making it accessible to users and monitoring its performance in real-world scenarios.

Justification: We chose the SSADM methodology as to be able to systematically analyze user requirements, design a robust system architecture, implement the desired functionalities, and thoroughly test the application. This approach will ensure that the e-commerce mobile application met the specific needs of customers and merchants, provided an intuitive user experience, and operated reliably in a production environment.

3. Example 3 Prototyping Methodology

Prototyping will also suite the e-commerce mobile application project. This methodology emphasizes iterative design and development cycles, allowing us to gather user feedback early and refine the system based on user input (Lauff et al., 2019). Below is an example of how to write a prototyping section.

3.1. Prototyping

We began by creating an initial prototype of the e-commerce mobile application using low-fidelity mockups or wireframes. These prototypes focused on the basic functionality and core features of the system. We then conducted usability testing sessions with a representative group of users to gather feedback on the prototype's user interface, navigation flow, and overall user experience.

Based on the feedback received, we refined the prototype by incorporating suggested improvements and addressing any usability issues identified. This iterative process helped us better understand user requirements and refine the design.

With each iteration, we enhanced the fidelity of the prototype, gradually transitioning to higher-fidelity interactive prototypes. These prototypes included more realistic visual designs, interactive elements, and simulated functionalities. We continued to collect feedback from users, incorporating their suggestions and preferences into subsequent iterations.

3.2. Iterative Development

As we progressed through the iterative prototyping process, we gradually implemented the actual functionalities of the e-commerce mobile application. We followed an incremental development approach, where we added new features and refined existing ones in each iteration.

During development, we used rapid prototyping tools and frameworks to expedite the creation of functional prototypes. These prototypes allowed us to showcase specific features and interactions to gather targeted user feedback and ensure the system met their needs.

3.3. Testing and User Feedback

At each iteration, we conducted usability testing and user feedback sessions to evaluate the usability, effectiveness, and satisfaction with the evolving system. We observed user interactions, gathered feedback through surveys or interviews, and recorded any issues or suggestions for improvement.

Based on the feedback received, we made iterative changes to the design and functionality, ensuring that the e-commerce mobile application met user expectations, improved usability, and aligned with their preferences.

3.4. Final Implementation

Once we completed several iterations of prototyping, incorporating user feedback and refining the system, we finalized the design and implemented the e-commerce mobile application. The iterative prototyping process helped us ensure that the final implementation addressed user needs and preferences effectively.

Justification: We chose Prototyping methodology because it will engage users throughout the development process, gather valuable feedback, and iteratively refine the e-commerce mobile application. This approach also creates a user-centric system that meet the specific requirements of customers and provide an intuitive and satisfying user experience.

3.5. Example 4:Expert System Methodology

According to Liebowitz (2019), this methodology is suitable for developing artificial intelligent systems which simulates the behaviour and reasoning of a human. Hence, a student doesn't just wake up to choose expert system method when his topic objective is to provide an ordinary kiosks or e-library; of course, it will be a wrong choice of method.

For the purpose of this guide, our topic of choice is "Design and implementation of an intelligent decision support system for aged retirees".

In the above topic, we will adopt the expert system methodology to guide the development. This methodology leverages knowledge representation and inference techniques to emulate the decision-making abilities of human experts in a specific domain.

3.6. Knowledge Acquisition

We began by acquiring domain-specific knowledge from subject matter experts (SMEs) in the target field. We conducted interviews, workshops, and knowledge elicitation sessions to extract the expertise and rules that guide decision-making in the domain. The knowledge acquired included facts, heuristics, rules, and relationships that experts use to solve complex problems.

3.7. Knowledge Representation

Using the acquired knowledge, we represented the domain expertise in a structured format suitable for the expert system. This involved organizing the knowledge into a knowledge base, which could include rule-based systems, ontologies, or other knowledge representation techniques. We created a knowledge representation model that captured the relevant concepts, relationships, and rules necessary for making informed decisions.

3.8. Inference Engine

Next, we developed an inference engine, which serves as the reasoning component of the expert system. The inference engine utilizes the knowledge base to make logical deductions, apply rules, and generate conclusions. It employs techniques such as forward chaining, backward chaining, or a combination of both to perform reasoning and draw inferences based on the available information.

3.9. User Interaction

To enable interaction with the expert system, we designed a user interface that allows users to input problem-specific data or answer questions relevant to the decision-making process. The user interface provides a means for users to communicate with the system and receive expert advice or recommendations based on their inputs.

3.10. Validation and Testing

Once the expert system was developed, we conducted extensive validation and testing to ensure its accuracy and reliability. We used test cases and scenarios derived from real-world problems to evaluate the system's performance. We compared the expert system's recommendations or decisions with those of human experts to validate its effectiveness and identify any discrepancies or limitations.

3.11. Maintenance and Refinement

The expert system methodology also emphasizes the iterative nature of system development. As new knowledge becomes available or domain expertise evolves, we incorporated updates into the knowledge base and refined the inference engine to improve system performance and accuracy. Ongoing maintenance and refinement ensure that the expert system remains up-to-date and aligned with the latest advancements in the domain.

3.12. Justification

We chose this method because we want to create an intelligent decision support system that emulates human expertise in a specific domain. The system's knowledge base, inference engine, user interface, and iterative refinement process combine to provide users with valuable insights, recommendations, and decision support based on expert knowledge and reasoning.

3.13. Example 5 Object-Oriented Design Methodology (OOADM)

Object-oriented analysis and design (OOADM) is a technical approach for analyzing and designing an application, system, or business by applying object-oriented programming, as well as using visual modeling throughout the software development process to guide stakeholder communication and product quality (Lobov & Tran, 2020). In other words, immediately you organize software design around data, rather than logic and function, you could choose OOAD as your methodology. This is only chosen when you are designing a complex system, or a gigantic applications. The topic we will choose for this method is “Design and implementation of a hand-simulated system for plastic manufacturing industry”

In the above project, one can adopt the object-oriented design methodology to guide the development of the software system. This methodology focuses on organizing the system's functionality into objects, encapsulating data and behaviour, and promoting modularity, reusability, and maintainability.

3.14. Analysis and Requirements Gathering

We began by conducting analysis and requirements gathering to understand the problem domain and identify system requirements. We interacted with stakeholders, such as users and domain experts, to elicit requirements, use cases, and functional specifications. We also identified entities, relationships, and constraints within the problem domain.

3.15. Object Modeling

Using the identified requirements, we created an object model that represented the system's entities, their attributes, and the interactions between them. We used techniques such as Unified Modeling Language (UML) to develop class diagrams, sequence diagrams, and state diagrams. The object model helped us visualize the system's structure and behavior and served as a blueprint for the design phase.

3.16. Class and Object Design

Based on the object model, we proceeded with class and object design. We identified classes, their attributes, and methods, ensuring that each class had a clear responsibility and encapsulated related data and behavior. We defined relationships between classes, such as inheritance, composition, and association, to establish the structure and interactions among objects.

We applied object-oriented design principles, such as abstraction, encapsulation, inheritance, and polymorphism, to create modular and extensible class designs. We also considered design patterns to address common design problems and promote best practices in object-oriented development.

3.17. Interface Design

As part of the object-oriented design methodology, we focused on designing interfaces that defined the contracts and interactions between objects. We identified the operations and messages exchanged between objects, ensuring clarity and consistency in the interface definitions. This allowed for loose coupling between objects and facilitated future modifications and enhancements.

3.18. Implementation and Testing

Following the design phase, we implemented the system using an object-oriented programming language, adhering to the class and object designs. We created class implementations, instantiated objects, and established the necessary relationships and collaborations. We conducted unit testing to verify the correctness and functionality of individual classes and their interactions.

Throughout the implementation and testing process, we followed agile development practices, iteratively refining the design and addressing any issues or feedback discovered during testing.

3.19. Maintenance and Evolution

The object-oriented design methodology supports ongoing maintenance and evolution of the system. As new requirements emerge or modifications are needed, we leverage the modularity and encapsulation offered by object-oriented design to make targeted changes without impacting the entire system. We can extend existing classes or create new ones, ensuring the system remains adaptable and maintainable over time.

Justification: We adopted this methodology because our proposed system needs modular design, code reuse, and maintainability. Hence, it is only the object-oriented principles and practices that can facilitate a clear and organized structure, enabling efficient development and enhancing the flexibility and extensibility of the system.

3.20. Data collection

Describe the data collection process, including the sources of data and the methods used to gather the required information.

3.20.1. Data Collection Process

This is the process of collecting data from your case study for the proposed system (Lobe et al., 2020). We shall go back to our ITS example to demonstrate data collection processes.

To gather the necessary data for our Intelligent Tutoring System (ITS), we employed a combination of primary and secondary data collection methods. This allowed us to obtain a comprehensive understanding of the learners' interactions and progress within the system.

3.20.2. Primary Data Collection

User Interactions: We collected user interaction data directly from the ITS platform. This included information such as the exercises attempted, the time spent on each task, the responses provided, and any hints or feedback accessed during the learning sessions.

Performance Metrics: We recorded learners' performance metrics, such as the number of correct and incorrect answers, completion time for exercises, and overall progress through the learning materials.

User Feedback: We incorporated feedback mechanisms within the ITS interface to allow users to provide qualitative feedback on their learning experience. This could include opinions on the system's usability, helpfulness of the instructional materials, and suggestions for improvement.

3.20.3. Secondary Data Collection

Pre-existing Datasets: We also leveraged pre-existing datasets from relevant research studies or educational resources that contained anonymized learning data. These datasets provided additional insights into learner behaviors, common challenges, and effective instructional strategies.

Literature Review: We conducted a comprehensive literature review to gather insights from prior studies and publications related to intelligent tutoring systems. This helped us identify best practices, relevant metrics, and potential factors influencing the effectiveness of ITS implementations.

3.20.4. Data Collection Methods

To collect the primary data, we employed the following methods:

Log Files: We logged user interactions and system events using log files generated by the ITS platform. These log files captured detailed information about each user's actions and system responses.

Surveys and Questionnaires: We designed and administered surveys or questionnaires to gather qualitative feedback from learners. These surveys could be administered at specific intervals or upon completion of significant milestones to assess learner satisfaction, self-perceived progress, and perceived usefulness of the system's features.

Observations: We conducted direct observations of learners using the ITS, either in controlled settings or in their natural learning environments. Observations allowed us to gather contextual information, observe learner behavior, and capture any challenges or opportunities for improvement.

3.21. Ethical Considerations

Throughout the data collection process, we ensured the protection of user privacy and adhered to ethical guidelines. We obtained informed consent from participants, explained the purpose of data collection, and assured them of data confidentiality and anonymity. Any personal identifying information was securely stored and treated with strict confidentiality.

Explain the data analysis techniques you employed to interpret the collected data or evaluate the system.

3.21.1. Data Analysis Techniques

To interpret the collected data and evaluate the system's performance, we employed a combination of quantitative and qualitative data analysis techniques. These techniques allowed us to gain insights, identify patterns, and draw meaningful conclusions from the data.

Descriptive Statistics: We utilized descriptive statistics to summarize and describe the collected data. This included calculating measures such as mean, median, standard deviation, and frequency distributions. Descriptive statistics provided a quantitative overview of various aspects of the data, enabling us to understand the central tendencies, variations, and distributions of relevant variables.

Inferential Statistics: To draw inferences and make generalizations about the larger population from the collected data, we employed inferential statistics. This involved techniques such as hypothesis testing, confidence intervals, and regression analysis. Inferential statistics allowed us to assess the significance of relationships, test hypotheses, and derive statistical conclusions about the data.

Data Visualization: We used data visualization techniques to present the collected data in a visually appealing and easily understandable manner. This included creating charts, graphs, and plots such as bar charts, line graphs, scatter plots, and heatmaps. Data visualization enhanced the clarity and interpretability of the data, enabling us to identify trends, patterns, and outliers more effectively.

Qualitative Analysis: In addition to quantitative analysis, we conducted qualitative analysis of the collected data. This involved thematic analysis, content analysis, or other qualitative techniques, depending on the nature of the data. Qualitative analysis helped us uncover underlying themes, recurring patterns, and rich insights that may not be easily captured through quantitative measures alone. We coded and categorized qualitative data to identify commonalities, differences, and emerging themes, allowing for a deeper understanding of user experiences and perceptions.

User Feedback Analysis: We systematically analyzed qualitative user feedback, such as survey responses or interviews, to identify recurring themes, suggestions, and areas for improvement. We employed techniques such as open coding, thematic analysis, or sentiment analysis to extract insights from user feedback and understand user experiences, satisfaction, and perceptions of the system.

3.22. Organisation and environment

The organization and environment of a system provide the structure and context within which the system operates. Here's an example of how the organization and environment of a system can be described using the ITS system:

3.22.1. Organization

The ITS system is organized as a centralized platform that facilitates the delivery of personalized instruction and support to learners. Within the organization, there are key roles and components that contribute to the functioning of the system:

System Administrators: System administrators are responsible for managing and maintaining the ITS platform. They handle tasks such as user registration, system configuration, database management, and system updates. They ensure the availability, security, and reliability of the system.

Content Developers: Content developers are responsible for creating and curating instructional materials and learning resources that are integrated into the ITS. They collaborate with subject matter experts, educators, and instructional designers to develop high-quality content aligned with the learning objectives and curriculum.

Data Analysts: Data analysts play a crucial role in the organization of the ITS system. They analyze the collected learner data, evaluate system performance, and provide insights to improve the effectiveness of the system. They collaborate with educators and researchers to identify trends, patterns, and areas for improvement based on the data analysis.

Educators and Instructors: Educators and instructors utilize the ITS platform to deliver instruction, monitor learner progress, and provide guidance. They interact with learners, interpret system-generated reports and analytics, and offer personalized feedback and support. Educators play a vital role in integrating the ITS into the broader educational context and tailoring instruction to meet learners' needs.

3.22.2. Environment

The Intelligent Tutoring System operates within a broader environment that includes various stakeholders and factors:

Learners: Learners are the primary users of the ITS system. They access the platform to engage in interactive learning activities, complete exercises, receive feedback, and track their progress. Learners come from diverse backgrounds, including students, professionals, or individuals seeking self-paced learning in the specific domain.

Educational Institutions: The ITS system may be deployed within educational institutions such as schools, colleges, or training centers. In these settings, the system aligns with existing curricula and educational goals, complementing traditional instructional approaches and supporting personalized learning.

Industry and Research Communities: The ITS system interacts with industry and research communities. Industry professionals may contribute domain-specific expertise, provide real-world applications, or offer feedback to improve the system's relevance and applicability. Research communities contribute by exploring the effectiveness of the ITS, identifying new instructional strategies, and advancing the field of intelligent tutoring systems.

Technological Infrastructure: The ITS operates within a technological infrastructure that includes servers, networks, and cloud computing resources. The system relies on these infrastructural components to ensure seamless access, storage, and retrieval of data, as well as to support scalable and efficient delivery of instructional content.

Regulatory and Ethical Considerations: The environment of the ITS system is influenced by regulatory and ethical considerations. Compliance with data protection and privacy regulations is paramount to ensure the security and confidentiality of learner data. Ethical considerations include informed consent, data anonymization, and ensuring the system does not perpetuate biases or discriminatory practices.

3.23. Modus operandi of the present system

3.23.1. Input, output and process analysis

The input, output, and process analysis of any system demonstrate how the system receives inputs, generates personalized outputs, and employs a dynamic process to deliver adaptive instruction and support. This analysis enables the system to cater to individual learner needs, enhance learning outcomes, and provide valuable insights. We shall use the ITS system to give an example of how the modus operandi of the present system should be structured below.

3.23.2. Input Analysis

The Intelligent Tutoring System (ITS) receives various types of inputs to support personalized instruction and learning experiences. Here are examples of input sources in the ITS system:

User Profile and Preferences: Learners provide input through their user profiles, which may include personal information, academic background, and preferences. This input helps the system tailor the learning experience based on individual needs and preferences.

Initial Assessments: Learners may complete initial assessments or diagnostic tests that gauge their knowledge level and identify areas of strength and weakness. These assessments provide valuable input for the system to adapt instructional content and set appropriate learning goals.

Learning Objectives and Curriculum: The system receives input in the form of predefined learning objectives and curriculum guidelines. This input helps align the instructional content and activities with specific educational standards or requirements.

User Interactions: Learner interactions within the ITS platform, such as responses to exercises, quizzes, or interactive activities, serve as input to evaluate their progress, understanding, and areas that require further instruction.

3.24. Output Analysis

The ITS system generates various outputs to support and enhance the learning experience. Here are examples of output elements in the ITS system:

Personalized Instruction: The system generates personalized instruction based on learner profiles, assessments, and interactions. This output includes adaptive learning paths, targeted practice exercises, and instructional resources tailored to individual needs and learning styles.

Feedback and Progress Reports: The system provides immediate feedback on learner responses, helping them understand their mistakes, suggest correct solutions, and reinforce concepts. Additionally, progress reports are generated to track learner achievements, identify areas of improvement, and highlight mastery of specific skills or knowledge areas.

Recommendations and Adaptations: The ITS system generates recommendations for further study, additional resources, or learning materials based on learner performance and progress. These recommendations adapt to individual needs and provide targeted support to address learning gaps.

Analytics and Insights: The system generates analytics and insights based on learner data, providing educators and administrators with a comprehensive understanding of learner performance, engagement, and learning patterns. This output helps identify trends, instructional effectiveness, and areas for system improvement.

3.25. Process Analysis

The ITS system follows a dynamic and iterative process to deliver personalized instruction and support. Here is an overview of the process analysis in the ITS system:

User Profiling: The system collects user input, including learner profiles, preferences, and initial assessments, to create individual learner profiles and establish baseline knowledge levels.

Adaptive Instruction: The system utilizes input from user profiles, assessments, and user interactions to dynamically adapt the instructional content, difficulty level, and pacing to suit individual learners. This process involves matching learning objectives with learner needs and providing targeted instruction accordingly.

Response Evaluation: The system analyzes learner responses to exercises, quizzes, and activities to evaluate performance and provide immediate feedback. This process helps identify misconceptions, errors, or areas that require further instruction.

Progress Tracking and Adaptation: The system continuously tracks learner progress, updating user profiles and adjusting instructional content and recommendations based on performance and mastery of concepts. This process ensures that learners receive appropriate **challenges and support throughout their learning journey.**

Data Analysis and System Enhancement: The system performs data analysis on learner interactions, performance metrics, and feedback to generate insights for continuous improvement. This process involves analyzing patterns, identifying trends, and refining the system's algorithms, instructional strategies, and content recommendations.

3.26. Data flow diagram

This shows the flow of data within a system. The more the system entities, the larger the drawing. You could use as many drawings as possible to represent the flow of data in your proposed system, but try to make them as simple as possible. For example, in our ITS system we will use two dataflow diagrams. One for the students (fig 2) and the other for the staff (fig 3). One could also join the two together to form a larger drawing.

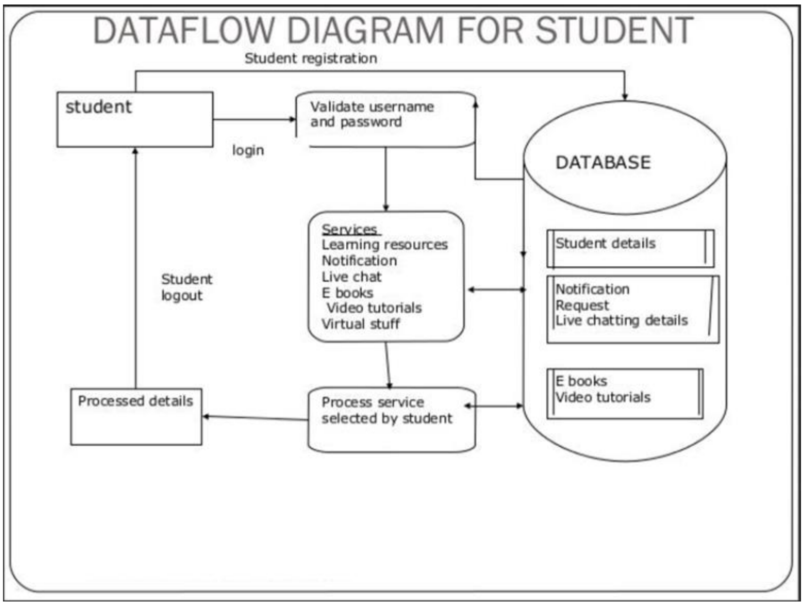

Figure 2 Dataflow diagram for students. Source:slideshare.net

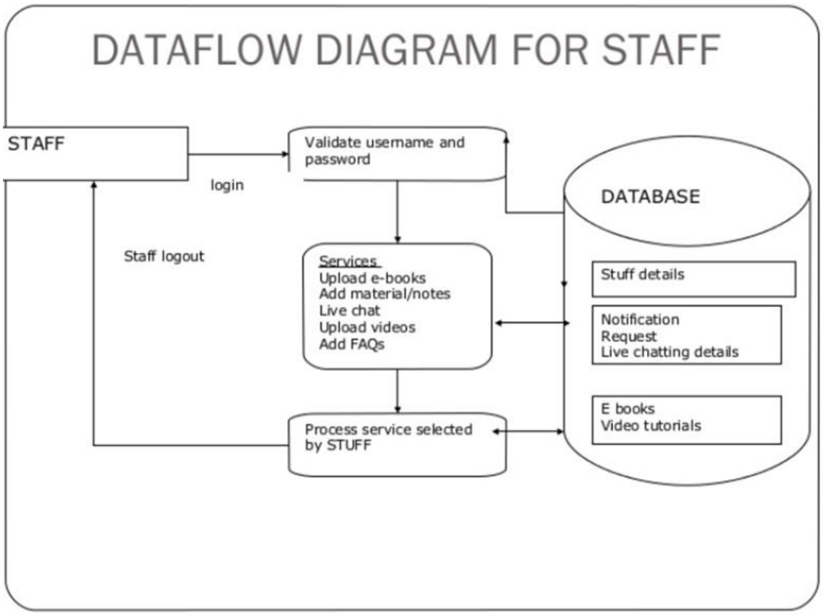

Figure 3 Structural Dataflow Diagram. Source: slideshare.com

3.27. Problems of the present system

In this section of your project, you should state all the problems identified during the system analysis. Make sure you list the problems with explanations to each. We shall give samples of problems using the ITS project topic as shown below.

Here are the problems being faced in the present ITS system as identified by our system analysis:

Lack of Personalization: One common problem with the present system is the limited level of personalization. The ITS may not effectively adapt to the individual needs, learning styles, and skill levels of each learner. This can result in a one-size-fits-all approach that fails to provide optimal support and challenges for all learners.

Insufficient Content Coverage: Another issue is the possibility of insufficient coverage of the curriculum or learning objectives. The present system may not offer a comprehensive range of instructional materials or adequately address all the necessary concepts and skills required for mastery of the subject matter. This can hinder learners' ability to acquire a well-rounded understanding of the topic.

Limited Feedback and Guidance: Inadequate feedback and guidance can be a problem within the current ITS. Learners may receive generic or shallow feedback that fails to pinpoint specific errors, misconceptions, or areas for improvement. This lack of detailed guidance can hinder the learning process and prevent learners from making meaningful progress.

Difficulty in Handling Complex Concepts: The present system may struggle to effectively handle complex concepts or higher-level problem-solving tasks. ITS systems heavily rely on rule-based approaches and may not possess the ability to provide deep insights or support for advanced topics that require critical thinking, creativity, or advanced reasoning skills.

Lack of Engagement and Motivation: The current system might not effectively engage and motivate learners. If the system fails to provide interactive and engaging learning experiences, learners may become disinterested or lose motivation to actively participate in the learning process. This can negatively impact knowledge retention and overall learning outcomes.

Limited Integration with Classroom Instruction: Integration of the ITS system with classroom instruction may be insufficient. The system may not seamlessly align with the curriculum, instructional goals, or teaching methodologies employed by educators. This lack of integration can create a disjointed learning experience and hinder the effectiveness of the ITS as a supportive tool in the classroom.

Technological Limitations: Technological limitations can also pose challenges within the present ITS system. System performance issues, compatibility problems with different devices or operating systems, or limited accessibility features may hinder smooth and reliable system operation. These technological barriers can impede learners' access to the system and impact the overall user experience.

It's important to note that these problems may vary depending on the specific implementation of the ITS system and the context in which it is used. Identifying and addressing these issues can lead to improvements in the design, functionality, and effectiveness of the ITS system.

3.28. The proposed system

In this section, you are required to highlight solutions the proposed system will bring to address the identified challenges in the previous section. Hence, the proposed Intelligent Tutoring System (ITS) aims to address the limitations and challenges of the current system by incorporating several enhancements and improvements. Here's an example of how you can describe the proposed ITS system:

The proposed Intelligent Tutoring System (ITS) is designed to provide an enhanced and personalized learning experience for learners. It leverages advanced technologies and pedagogical approaches to overcome the limitations of the existing system and foster improved learning outcomes. The key features and enhancements of the proposed ITS system are as follows:

Adaptive and Personalized Learning: The proposed ITS system focuses on adaptive and personalized learning. It employs sophisticated algorithms and machine learning techniques to dynamically adjust instructional content, difficulty levels, and learning paths based on individual learner profiles, performance data, and preferences. This personalization enables learners to receive tailored instruction that matches their specific needs, learning styles, and skill levels.

Comprehensive Content Coverage: The proposed system aims to provide comprehensive coverage of the curriculum or learning objectives. It includes a wide range of instructional materials, interactive exercises, simulations, and multimedia resources to ensure learners have access to a diverse set of learning materials. This broad coverage enables learners to gain a well-rounded understanding of the subject matter.

Rich Feedback and Guidance: The proposed ITS system emphasizes the provision of rich and detailed feedback to learners. It employs natural language processing techniques, advanced analytics, and expert-authored feedback to provide learners with specific insights, explanations, and suggestions for improvement. This detailed guidance helps learners identify and correct errors, address misconceptions, and deepen their understanding of the concepts.

Advanced Concept Handling: The proposed system focuses on effectively handling complex concepts and higher-level problem-solving tasks. It integrates advanced cognitive models, reasoning engines, and domain-specific knowledge to support learners in tackling challenging topics. The system offers advanced problem-solving guidance, hints, and step-by-step solutions, fostering critical thinking, creativity, and advanced reasoning skills.

Engaging and Motivating Learning Environment: The proposed ITS system incorporates gamification elements, interactive interfaces, and immersive learning experiences to enhance learner engagement and motivation. It includes features such as badges, rewards, progress tracking, and interactive simulations to make the learning process more enjoyable and interactive. This increased engagement helps to sustain learners' motivation and promote active participation.

Seamless Integration with Classroom Instruction: The proposed ITS system emphasizes seamless integration with classroom instruction. It provides educators with tools to align the system with the existing curriculum, instructional goals, and teaching methodologies. The system supports collaboration between educators and learners, allowing for seamless integration of classroom activities, assignments, and assessments into the ITS platform.

Robust Technological Infrastructure: The proposed system ensures a robust and reliable technological infrastructure. It incorporates scalability, compatibility, and security features to handle a large number of learners, support various devices and platforms, and ensure data privacy and protection. The system leverages cloud computing, distributed databases, and advanced network architectures to deliver a seamless and secure user experience.

3.29. High level model of the proposed system

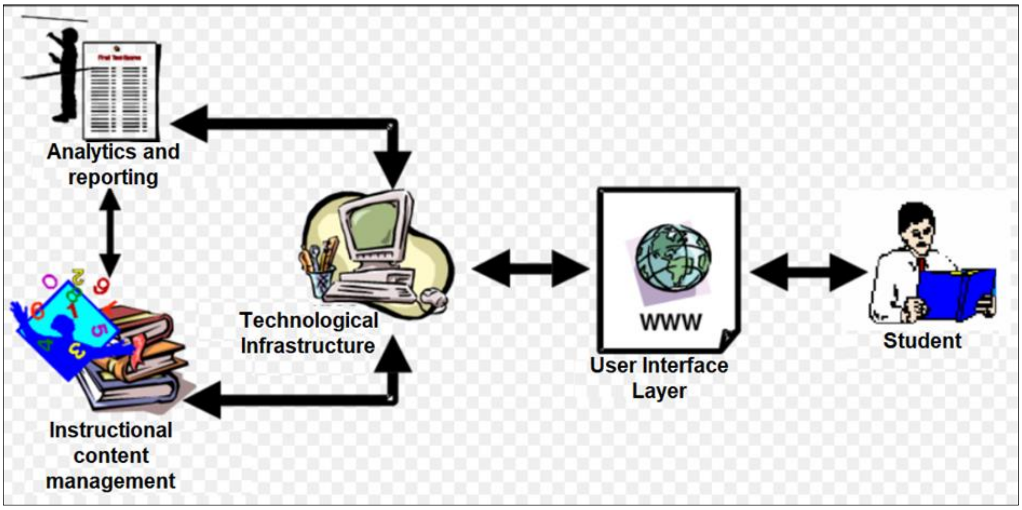

Figure 4 High level model. Source: Researchgatebll

A high-level data model conveys the core concepts and/or principles of a system in a simple way, using concise descriptions. This is mostly depicted pictorially with important system components as shown below.

Here's an example of a high-level model for the proposed Intelligent Tutoring System (ITS):

3.30. User Interface Layer

Learner Interface: This component provides an intuitive and user-friendly interface for learners to access the ITS system. It includes features such as personalized dashboards, progress tracking, interactive exercises, and access to instructional materials.

Educator Interface: This component offers educators tools to manage and customize the system, monitor learner progress, access analytics, and tailor instruction. It enables educators to align the ITS with the curriculum and integrate classroom activities seamlessly.

3.31. Adaptive Learning Engine

Learner Profiling: This component collects and analyzes learner data, including performance, preferences, and learning styles. It builds individual learner profiles and adapts instructional content based on their specific needs.

Content Recommendation: This component utilizes machine learning algorithms and data analytics to recommend personalized instructional materials, exercises, and learning paths for learners. It ensures that learners receive appropriate challenges and targeted support throughout their learning journey.

Progress Tracking and Assessment: This component continuously tracks learner progress, monitors performance metrics, and generates detailed progress reports. It allows educators and learners to identify strengths, areas for improvement, and mastery of concepts.

3.32. Instructional Content Management

Curriculum Mapping: This component maps the ITS content to the curriculum or learning objectives. It ensures comprehensive coverage of the required topics and aligns instructional materials with educational standards and requirements.

Content Authoring and Curation: This component allows content developers and educators to create, curate, and update instructional materials, exercises, quizzes, and multimedia resources. It ensures that the system offers a diverse range of engaging and informative content.

Content Delivery and Accessibility: This component delivers instructional content to learners through various modalities, such as text, audio, video, and interactive simulations. It supports accessibility features to accommodate learners with different learning needs.

3.33. Analytics and Reporting

Data Collection and Analysis: This component collects, stores, and analyzes learner data, interaction logs, and assessment results. It employs data analytics techniques to generate insights, identify patterns, and measure learning outcomes.

Performance Analytics: This component evaluates learner performance, tracks engagement, and provides real-time feedback and progress reports to learners and educators. It helps identify areas of improvement, mastery of concepts, and personalized interventions.

System Improvement and Research: This component leverages analytics to inform system improvements, identify pedagogical strategies, and support research on instructional effectiveness. It facilitates ongoing refinement and enhancement of the ITS system.

3.34. Technological Infrastructure

Server and Database Management: This component manages the server infrastructure, ensuring scalability, reliability, and data security. It handles user authentication, data storage, and system maintenance.

Integration and Compatibility: This component ensures compatibility with various devices, platforms, and learning management systems (LMS). It allows seamless integration with existing educational technologies and infrastructures.

Security and Privacy: This component implements robust security measures to protect learner data and ensure privacy. It adheres to data protection regulations, encryption standards, and user consent protocols.

The high-level model illustrates the main components and functionalities of the proposed Intelligent Tutoring System (ITS). It encompasses the learner and educator interfaces, the adaptive learning engine, instructional content management, analytics and reporting, and the underlying technological infrastructure. The integration of these components aims to provide a personalized, comprehensive, and engaging learning experience for learners while supporting educators in delivering effective instruction.

4. Conclusion

In conclusion, this guide serves as a valuable resource for researchers, students, and professionals undertaking computing projects. The methodology section is a crucial component of any research work, as it outlines the systematic approach used to gather data, conduct experiments, and analyze results. This guide provides a comprehensive and structured framework to help individuals craft a robust methodology section that ensures the reliability, validity, and reproducibility of their findings. Throughout the guide, we have emphasized the importance of clarity, precision, and transparency in describing the research methods. By following the step-by-step guidelines and best practices outlined in this guide, researchers can effectively communicate the process they employed, allowing others to understand, evaluate, and potentially build upon their work. Additionally, the guide acknowledges the diverse nature of computing projects, ranging from software development and algorithm design to data analysis and simulations. As a result, it offers flexible methodologies and highlights the need to tailor the section to match the specific project's goals and requirements.

Compliance of ethical standard

Disclosure of conflict of interest

No conflict of interest to disclosed.

References

[1] Ananda, A. R., Nama, G. F., & Mardiana, M. (2022). Pengembangan Sistem Informasi Geografis Pemerintahan Kota Metro Dengan Metode SSADM (Structured System Analysis and Design Method). *Jurnal Informatika dan Teknik Elektro Terapan*, *10*(1).

[2] Borgianni, Y., & Maccioni, L. (2020). Review of the use of neurophysiological and biometric measures in experimental design research. AI EDAM, 34(2), 248-285.

[3] Bougie, R., & Sekaran, U. (2019). Research methods for business: A skill building approach. John Wiley & Sons.

[4] Gandhmal, D. P., & Kumar, K. (2019). Systematic analysis and review of stock market prediction techniques. Computer Science Review, 34, 100190.

[5] García-Peñalvo, F. J. (2019). W-STEM Project Overview.

[6] Jensen, K. B. (Ed.). (2020). A handbook of media and communication research: Qualitative and quantitative methodologies. routledge.

[7] Lauff, C., Menold, J., & Wood, K. L. (2019, July). Prototyping canvas: Design tool for planning purposeful prototypes. In Proceedings of the design society: international conference on engineering design (Vol. 1, No. 1, pp. 1563-1572). Cambridge University Press.

[8] Liebowitz, J. (Ed.). (2019). The handbook of applied expert systems. cRc Press.

[9] Lobe, B., Morgan, D., & Hoffman, K. A. (2020). Qualitative data collection in an era of social distancing. International journal of qualitative methods, 19, 1609406920937875.

[10] Lobov, A., & Tran, T. A. (2020). Object-oriented approach to product design using extended NX Open API. Procedia Manufacturing, 51, 1014-1020.

[11] Mishra, S. B., & Alok, S. (2022). Handbook of research methodology.

[12] Sileyew, K. J. (2019). Research design and methodology. Cyberspace, 1-12.

[13] Tilley, S. (2019). Systems analysis and design. Cengage Learning.

[14] Wright, P., & McCarthy, J. (2022). Experience-centered design: designers, users, and communities in dialogue. Springer Nature.

[15] https://fordfiringorder.com/

[16] https://www.researchgate.net/publication/303016868_Learning_SQL_through_Intelligent_Tutoring_System/figures?lo=1&utm_source=bing&utm_medium=organic

[17] https://www.slideshare.net/edwinmyalla/enhanced-intelligent-tutoring-system-with-mobile-supportppt-autosaved