**Student Assessment Submission and Declaration**

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

|  |  |  |  |
| --- | --- | --- | --- |
| Student name: Rashed Hasan | | Assessor name: | |
| Issue date (1St Submission):  **14/5/2024** | Submission date (1St Submission): 21/6/2024 | | Submitted on:11/6/2024 |
| In case of resubmission | | | |
| Issue date (2nd Submission):  27/6/2024 | Submission date (2nd Submission):  30/6/2024 | | Submitted on: |
| Programme: Higher National in computing – Software Engineering | | | |
|  | | | |
| Assignment number and title: 1 **Developing an Event Management System** | | | |

**Activity 1:**

1. **Solution:**

To start developing an event management system (EMS) for XYZ events, it is necessary to first verify the basic problem that the application aims to solve. Therefore, I will talk about the definition of the problem and its concept in the scenario, and then I will prepare a solution that will address the problem at hand, as follows:

Therefore, the main problem that XYZ Events aims to solve by using the EMS event management system is the complexity and inefficiency that we currently face in planning and implementing events. Traditional approaches often involve disparate systems for communications, resource allocation, attendance management, and compliance, leading to confusion, errors, and wasted resources. Additionally, manual processes can be time-consuming and prone to human error, affecting the overall quality and success of events.

This may also lead to poor resource allocation, poor communication between team members, lack of insights based on data and information, and lack of attendance interaction. These shortcomings can lead to suboptimal event experiences, higher costs, and lower customer satisfaction. Existing manual or semi-automated processes may not provide the scalability and accuracy required to handle diverse and complex events, leading to errors such as double booking, mismanagement of attendance information, and insufficient compliance with regulations.

Therefore, to address these challenges, the solution is to develop a comprehensive and easy-to-use event management system (EMS) that takes advantage of advanced and effective technology in order to simplify and enhance the entire event management life cycle. XYZ Events requires an integrated and efficient solution that streamlines all aspects of event management into a cohesive, easy-to-use platform. This system needs to facilitate seamless communication between team members, improve resource allocation, provide detailed data analysis to make informed decisions, enhance attendee engagement through a seamless registration process, and ensure regulatory compliance. By addressing these issues, EMS aims to enhance efficiency, reduce errors, deliver exceptional value to customers, and set new standards in the event management industry.

The proposed solution is to develop an Event Management System (EMS) that incorporates advanced technologies and best practices to simplify the entire event management process. The EMS will feature an easy-to-use interface, comprehensive capabilities for event management, attendance recording and ticketing, and powerful data processing for compliance and analysis.

**Outline the Needs of Both Users and the System Requirements:**

**User Requirements:**

1. Event Organizers:

* Account Creation and Management: Ability to create and manage their accounts with personal details like name, email, and phone number, Password.
* Event Creation and Management: Tools to create, edit, and manage events with specifications such as event name, date, location, capacity, type, and description.
* Event Schedule View: Access to a comprehensive list of all their events to avoid scheduling conflicts.

1. Attendees:

* Registration: Simple registration process requiring personal details like name, email, and phone number.
* Event Selection: Ability to browse and select events from a list of available events.
* Ticketing: Option to select ticket types and prices.

**System Requirements:**

System requirements refer to the specific criteria that a system must meet in order to effectively deliver high-quality services to its users. System requirements for the XYZ Events System encompass:

Integration of databases: The integration of the XYZ Events System with the database should ensure that any newly added, altered, or removed events are securely and consistently saved. When a certain alteration occurs in the system, the database should accurately mirror that modification on the XYZ Events System by simply retrieving the data that already exists in the database.

**Physical components of a computer system, such as the processor, memory, and storage devices.**

To ensure optimal functionality and delivery of necessary services to system users and administrators, it is imperative to furnish the system with critical hardware components. Some of the components may comprise the following:

Server: To provide efficient and effective data processing, a server is necessary to handle the numerous requests from users on the online application. Furthermore, all crucial data pertaining to the events, and beyond, will be securely stored and preserved, and may be readily accessed at any time by authorized personnel. With a reliable server, consumers have the ability to utilize the system at their convenience for booking and viewing events. Additionally, administrators can efficiently manage events and perform other necessary tasks as needed.

An essential element for the system is a robust cooling system to provide regular maintenance and prevent the server from overheating, which could lead to system failure or catastrophic consequences. The purpose of the cooling systems is to maintain the server's temperature within an optimal range, neither too high nor too low. By doing this action, the server will be capable of operating efficiently and handling data and traffic in accordance with the necessary requirements.

Memory: Another essential hardware need for the system is a sufficient amount of RAM. This is due to the fact that having ample memory results in a substantial enhancement in the speed and efficiency of the system, enabling it to effectively handle numerous procedures and demands with greater speed and efficiency. By possessing a proficient memory, the system will have the capacity to promptly exhibit any modifications implemented within the system. For instance, when an administrator endeavors to modify some information regarding a preexisting event, the system should promptly exhibit the recently-updated data to the users.

Secondary storage is a crucial hardware component that is necessary for the efficient operation of the XYZ Events System. This feature is crucial for the system as it enables the storage of information pertaining to users, events, and administrators. Secondary storage has the capacity to efficiently store and retain large amounts of data over an extended period of time. This guarantees the safety, security, and preservation of data, even in the event of a system shutdown, in contrast to main memory or RAM.

**Software requirements** are an additional aspect of system requirements. These are computer programs or applications that the system can utilize to provide certain services or improve them. Software requirements encompass a variety of examples.

Firewall: One of the important things that the XYZ Events System can certainly take advantage of is using a firewall to ensure that no unauthorized access occurs. The firewall essentially acts as a layer of security or defense for the network, and will only allow certain individuals to access the system depending on how it is set up; as the firewall manages the ongoing and outgoing traffic for the network. In the context of the XYZ Events System, because the web application is a public service, there is no problem with pretty much any IP address or user accessing it. However, it should be ensured that no one other than the admin is able to access the administrator dashboard and use it. Thus, the firewall can be set up for the private network side of the company to ensure that only authorized people are allowed there. From this, we see that the firewall essentially prevents private data from being disclosed and used inappropriately.

Data Backup: Another thing the XYZ Events System can take advantage of regarding the software side of the requirements is to have a backup service for the system. This is because the data inside the system can easily be vulnerable, and if correct measures are not followed, then the data of the employees, users, and many more can be lost, resulting in even bigger issues for the company. By performing a backup on the data at least once a week, the company will make sure that in the event the data is lost, destroyed, or the company faces a cyber-attack of any sort, there is already a copy of the data that was not destroyed and was protected. The XYZ Events System can then operate on the data from the backup instead of suffering from the destruction of all data.

UML Design Tools: One of the important things for the XYZ Events System is for it to be designed well in terms of functionalities, database, and more. UML design tools are very useful as they give the developer a visual representation of how the XYZ Events System is going to look, along with the actions and methods it provides to the users prior to building the system itself. The design step is quite essential as it is also one of the SDLC phases for any system or software being built. After the design is completed, the developer can bring life to the design by applying it on the system using programming languages to develop the web application for the XYZ Events System.

Software Testing Tools: This is one of the essential things for the XYZ Events System as well; because it is also one of the SDLC phases of testing. Testing the software or system being built is very important to make sure that the validation and verification are being done properly. In the context of the XYZ Events System, it is absolutely important to apply testing to different areas within the system such as unit testing, performance testing, and more. With this, the company ensures that the software is performing as required, and the clients or customers will be pleased with using and taking advantage of the system.

**2. Solution:**

During the development of the XYZ Events System, many risks may arise. These risks can encompass the following:

Technology risk refers to the potential risks that a project may encounter in relation to its technological aspects. It may encompass aspects such as system or software security, as well as compliance matters pertaining to significant policies. Several of these difficulties can lead to project failure or failure in specific areas of the project.

In the context of the XYZ Events application, the technology could potentially encounter issues if the developers and the organization did not take sufficient precautions to ensure the app's security. It is crucial to safeguard the private information of passengers when they log in and use the application, and to comply with GDPR regulations. Regarding the compliance issue, it is crucial that the organization adheres to all the required regulations for this project and any additional policies established by the organization. It is essential to ensure that the technological aspect of the project is functioning optimally and free from any issues.

**Communication Risk:** This risk is prevalent in projects due to inadequate communication among project participants, whether inside teams or between various teams. It can lead to project risks that have a detrimental impact on the organization and may even end in the complete failure of the project. Effective communication between stakeholders and managers is essential for identifying necessary modifications, additions, and updates to the system. Constructive feedback obtained via this communication process can greatly enhance the project's prospects for success.

In the context of the XYZ Events application, it is imperative to ensure that all individuals involved in the project have a clear understanding of their assigned tasks and responsibilities. In addition, there should be a high level of communication among team members and between different teams. The purpose of this is to guarantee that all individuals have a comprehensive understanding of the capabilities and features of the system, and that the system operates in accordance with the requirements of the users. An illustrative instance of this is that the team responsible for designing the web application should maintain a satisfactory degree of contact with the team responsible for coding the functionalities and features. This is to guarantee that the design is optimized to effectively integrate with the system's operation and many features, ensuring proper performance. If this unimpeded current were to persist, it might potentially give rise to significant complications for the project, leading to financial losses and various other problems. If a communication risk arises, it could potentially jeopardize the success of the project.

**Scope Creep Risk:** This risk is a regular occurrence in projects, as the project's scope expands beyond its intended boundaries and becomes difficult to manage. The project's cost will increase when additional features are introduced due to the expanding scope. In order to incorporate these characteristics into the program or system, the project may necessitate the acquisition of supplementary hardware and software. Furthermore, this increases the intricacy of the project and also makes it more susceptible to failure. It is crucial to exercise caution regarding present risks and strive to comprehend and execute the project correctly in order to prevent the occurrence of such risks. If the existing risk is not effectively mitigated, it could result in significant financial losses for the project. This could potentially determine whether it is feasible to proceed with the project or if it would be more prudent to abandon it and begin preparing for a new project.

In the context of the XYZ Events System, it is crucial to meticulously design the project and thoroughly analyze and comprehend all pertinent elements and needs. By implementing this approach, the project's scope will be universally understood, hence reducing the likelihood of encountering such a risk. Regardless, this risk must be mitigated to avoid potential financial consequences for the organization or corporation. The potential magnitude of this loss could have a detrimental effect on both the project and the organization.

**Cost Risk:** A potential risk that commonly arises in any project is insufficient financial resources to sustain the project. This is due to the fact that if a project necessitates the integration of specific software, hardware, or database, and the organization lacks the financial resources to support it, it would pose a significant risk to the project and diminish its likelihood of success. Moreover, the potential cost risk not only poses a threat to the project, but also puts the entire organization at risk, including some essential internal procedures. To mitigate this risk, it is crucial to engage in meticulous planning and identify al l the essential expenses associated with a project. Furthermore, by conducting a thorough feasibility study to assess the viability and sustainability of the project, the potential risks can be significantly reduced. This is because the organization should refrain from initiating a project if they are aware that they lack the necessary financial and resource allocation to support it adequately. Avoiding the current danger is imperative, since it will have a very detrimental impact on the project.

When considering the context of the XYZ Events System, it is crucial to engage in meticulous planning regarding the necessary resources, costs, and other funds that are vital for the project's success. To construct the XYZ Events System, it is imperative that all essential resources are readily available and prepared for usage as required. Embracing such risk for the XYZ Events System will inevitably lead to a significant failure, either affecting the entire project or a specific part of it. The success of the XYZ Events System heavily relies on the thorough study and effective planning of the project.

**Operational Risk:** refers to the potential for disruption or underperformance in the essential processes and operations of an organization. This type of risk poses significant harm to the project as the unavailability of daily or critical activities not only introduces extra risks but also reduces the project's likelihood of success. There are various ways in which this risk could arise, including human errors, natural disasters, cyberattacks, and other factors. By implementing meticulous strategizing and optimizing productivity, the potential for operational risk can be mitigated or minimized to the greatest extent possible.

**Skill Resource Risk:** Skill resource risk is a frequently encountered risk in projects, referring to the lack of sufficient experience or resources among team members. This risk is highly consequential since it has the potential to significantly harm the project if team members lack sufficient expertise to create the system effectively, or if the planning team fails to account for all the required expenses and resources. Neglecting to safeguard the organization against the risk might result in a catastrophic failure of the project, as well as significant financial losses. Several factors contribute to the occurrence of such a risk, including:

Not adequately assessing the skills of developers can lead to significant risks in terms of resource allocation. Assigning tasks to team members that they are unable to perform or unable to perform effectively can result in a lack of expertise and hinder the completion of tasks.

Within the XYZ Events System framework, the skills resource risk arises when the developer lacks the expertise and knowledge to effectively plan and construct various components of the system. An instance of this would be if the developer failed to incorporate crucial functionalities into the XYZ Events System. Therefore, it is imperative to ensure that the developer responsible for the existing system possesses the capability to effectively execute the project and is also proficient in making any required modifications to the system as needed. If the developer lacks sufficient expertise in web app development and fails to meet the essential design specifications, it could pose a significant risk to the firm.

**Market risk:** refers to the potential for a project to fail in achieving its intended standards and outcomes. Additionally, the project may have been deployed after the deadline, which could have contributed to further complications. When such a situation arises, competitors operating in the same market have the opportunity to monitor and exploit your faults and blunders, so causing significant harm or even eliminating your organization or firm from the market. Absolutely avoiding this risk is imperative, since it has the potential to result in significant financial losses or even the complete demise of the organization, namely owing to the unsuccessful completion of a major project.

Within the XYZ Events System framework, it is imperative to mitigate market risk. Additionally, during the project planning phase, it is crucial to accurately and efficiently determine the project's attributes to ensure successful implementation in the future. If the planning and analysis of the XYZ Events System were executed accurately and crucial facts were documented and stored appropriately, the likelihood of encountering market risk would be significantly reduced. This is because if the system underperforms, fails to meet market standards, or is released after the set deadline, it can pose a significant market risk. Such risks have the potential to harm the entire project and cause significant damage to the organization.

**3. Solution:**

a) XYZ Events aims to develop an innovative Event Management System (EMS) to streamline event organization and enhance client and attendee experiences. The EMS will incorporate advanced technology and best practices to address challenges such as resource allocation, team communication, data analysis, attendee engagement, and regulatory compliance. This document outlines the functional and non-functional requirements of the EMS to ensure a clear understanding of the application's scope.

**Functional Requirements:**

Functional requirements are the specific actions or behaviours that the Event Management System (EMS) must perform to meet the needs of its users. These requirements dictate the functionalities and features that the system should provide to facilitate event organization effectively. Functional requirements often pertain to certain functions or features of a system that necessitate user-triggered activation for the desired functionality to occur. This could involve actions such as pressing a button, and potentially more. Functional requirements refer to the necessary actions that a system must take. In the context of XYZ Events' EMS, functional requirements include:

* User Account Creation: Organizers should be able to create accounts by providing essential information such as their name, email, and phone number. This functionality allows organizers to access the system securely and manage their events efficiently.
* Event Creation and Management: The EMS should empower organizers to create new events with detailed specifications, including event name, date, location, capacity, and description. Additionally, organizers should have access to a comprehensive list of all available events to avoid scheduling conflicts when creating new events.
* Attendee Registration and Ticketing: Attendees should be able to register for events by providing their personal details like name, email, and phone number. They should also have the option to select their desired event from a list and choose ticket types with associated prices. The system must efficiently manage attendee information, ensuring accurate linkage to respective events.
* User Interface Design: The EMS should feature a user-friendly interface with a navigation bar and footer for easy navigation. This design aspect is crucial for enhancing user experience and ensuring that users can navigate through the system effortlessly.

**Non-Functional Requirements:**

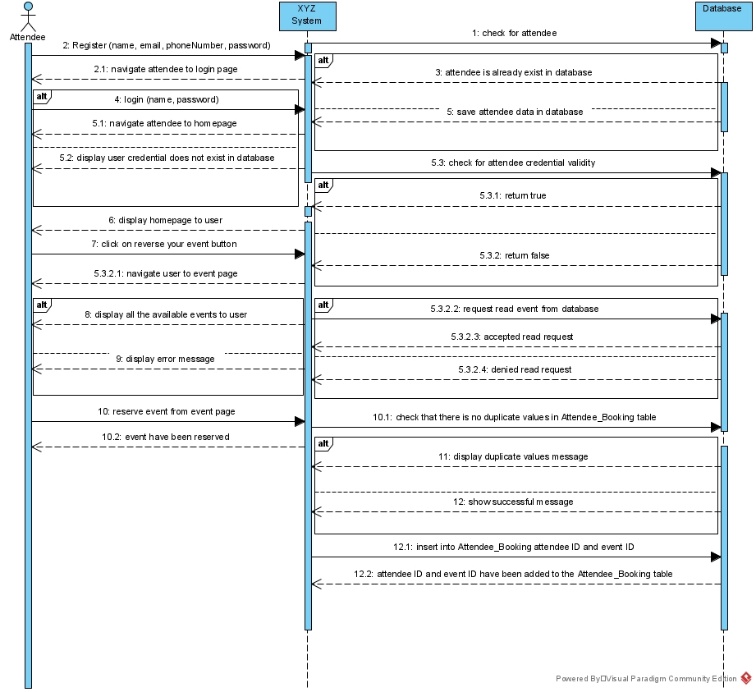
Non-functional requirements define the operational attributes and quality characteristics of the EMS. These requirements focus on aspects such as performance, security, usability, scalability, reliability, and maintainability. Meaning that the it doesn’t really require user involvement for the non-functional aspect of the system. For instance, when users submit their personal information through the registration form, the system should encrypt this data using SSL/TLS encryption protocols to protect it from unauthorized access (relating to security). This encryption ensures that sensitive user data, such as names, email addresses, and phone numbers, remains confidential and secure while being transmitted over the internet. It adds an extra layer of protection against potential threats such as eavesdropping or interception of data packets during transmission, thereby safeguarding user privacy and maintaining compliance with data protection regulations. Another one could be involved with how secure the system is in terms of saving user data and being able to keep their private data hidden from unauthorized individuals. For XYZ Events' EMS, non-functional requirements include:

* Performance: The EMS should exhibit high performance, ensuring fast page loading times and efficient processing of user requests. Performance optimization is crucial to provide a seamless user experience and prevent delays in event management tasks.
* Security: Security measures must be implemented to protect user data and ensure compliance with relevant regulations. This includes encryption of sensitive information, secure authentication protocols, and measures to prevent unauthorized access to the system.
* Usability: The EMS should be easy to use, with a clear and intuitive interface that allows users to perform tasks efficiently. Usability considerations are essential for enhancing user satisfaction and adoption of the system.
* Scalability: The system should be scalable to accommodate a growing number of events and users without sacrificing performance. Scalability ensures that the EMS can meet increasing demands as XYZ Events expands its operations.
* Reliability: The EMS should be reliable, ensuring high availability and minimal downtime. This reliability is crucial to prevent disruptions in event management activities and maintain customer trust.

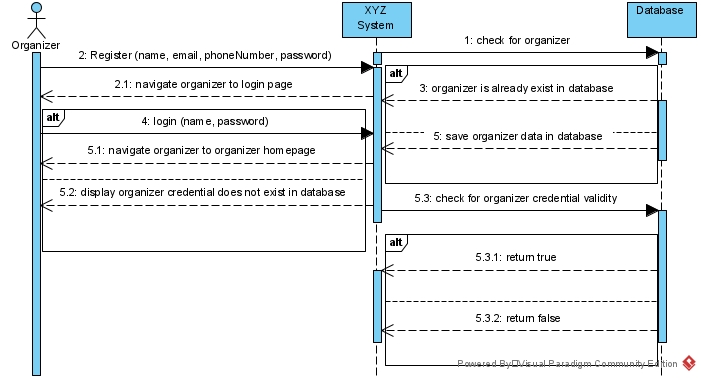
b) First, the attendee goes to the attendee register page where the name, email, phoneNumber, and password is entered in order to create an account in the XYZ System. After the attendee clicks on the submit button the system then checks if there is an existent account for this user based on the name, and password. If there is, it shows that the account is existent, if not it would save the current data entered into the database and creates an account with those values. Additionally, after the data is saved in the database, the user is navigated to the login page by the XYZ System in order to enter the name and password for the account that they created. The system then sends a request to the database checking if the login credentials is correct. If the login credentials are correct, then it would return true and the user would be navigated to the homepage for the attendees.

Else, it would return false and the XYZ System will display a message to the user saying that the login credentials are incorrect. From the homepage, the user can click on the reserve event button which will navigate the user to the event page. Once the user gets there, the application sends a request to the database in order to read the current rows in the event table in the database. If the request was accepted, then the attendee will be able to view all the available events.

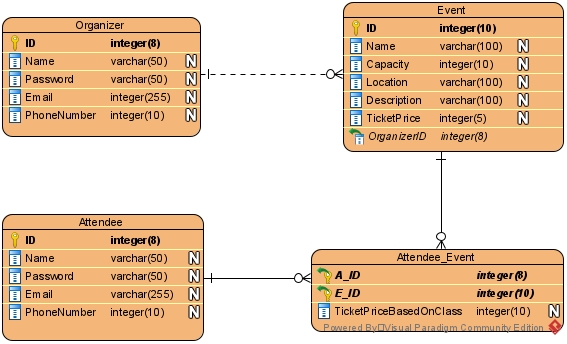
Otherwise, it will send an error message to the user explaining the matter at hand. Once the seas all the available events and would like book one of them, as they click on the submit button, it would first check if the same attendee has booked this particular event before, i.e. if there are any duplication in the Attendee\_Booking table which is a result from the Many-to-Many relationship. If there are no duplication, then it would show a successful message and the database will allow the XYZ system to insert the ID of the current attendee as well as the ID of the event as foreign keys in the Attendee\_Booking table, and then once it all goes successfully, the event is reserved for this attendee.



What would happen is that the organizer would navigate to the organizer register page and enter the details about him such as the name, email, phoneNumber, password. After the organizer clicks on the submit button the system then checks if there is an existent account for this user based on the name, and password. If there is, it shows that the account is existent, if not it would save the current data entered into the database and creates an account with those values. After the data is saved in the database, the organizer is navigated to the organizer login page by the XYZ System in order to enter the name and password for the account that they created. The system then sends a request to the database checking if the login credentials is correct. If the login credentials are correct, then it would return true and the user would be navigated to the homepage for the organizers. Else, it would return false and the XYZ System will display a message to the user saying that the login credentials are incorrect.



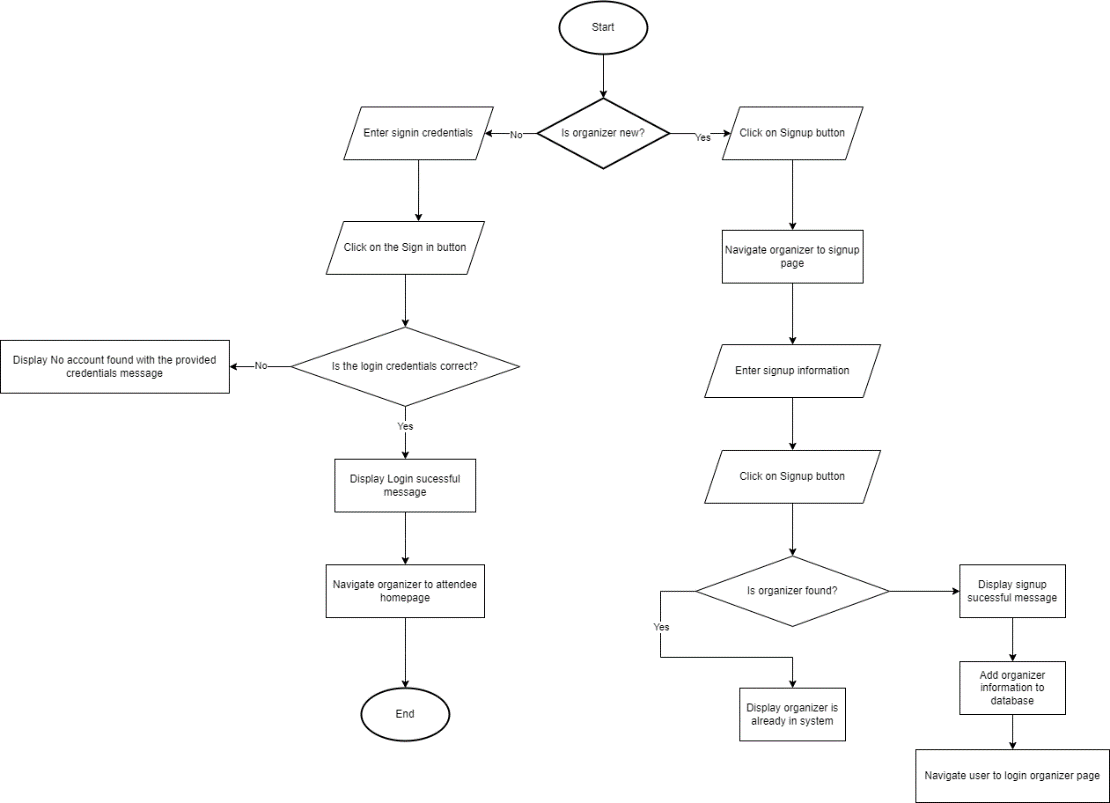
c) Entity Relationship Diagram (ERD)



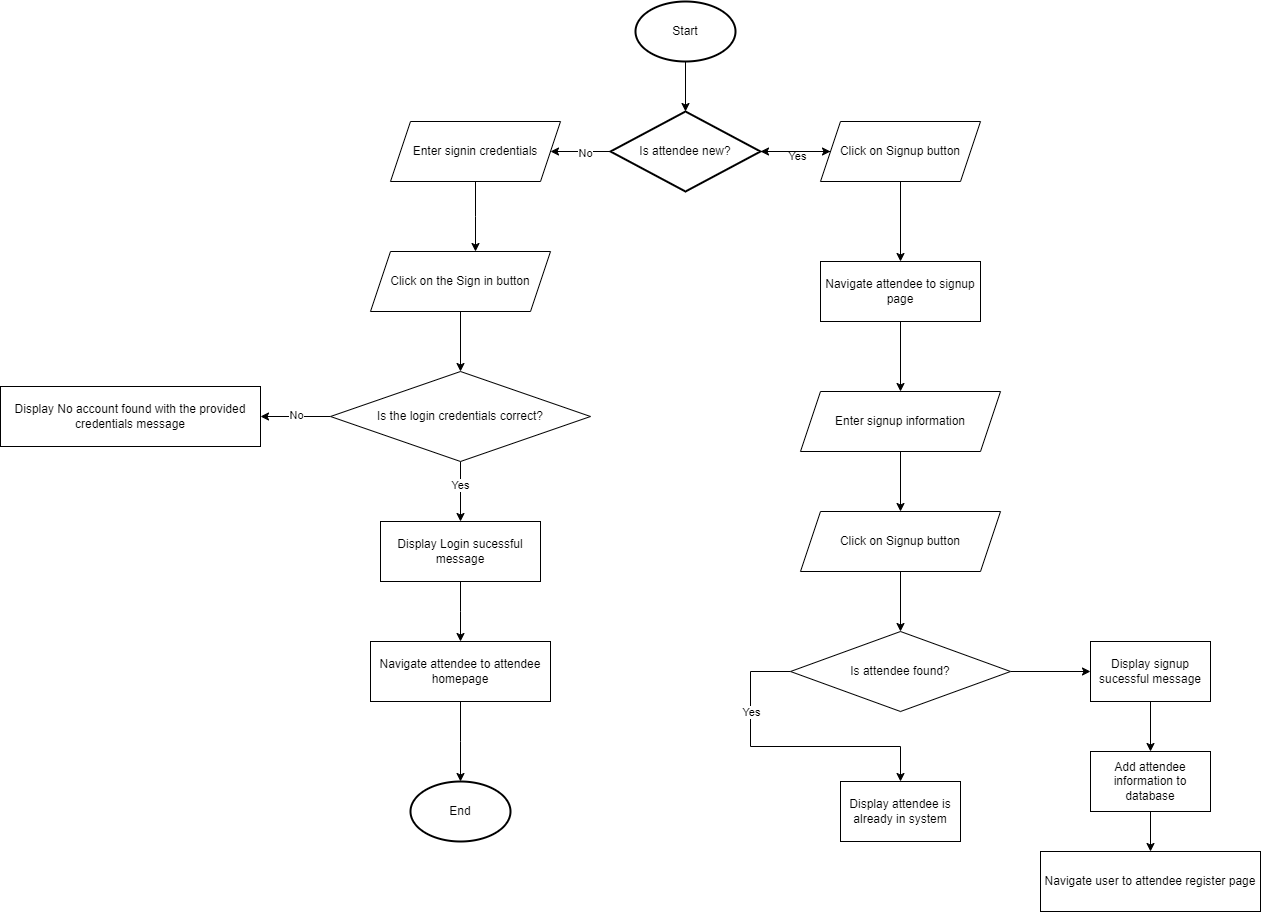
d) Employ flow chart techniques to implement the system functionality, ensuring a lucid depiction of the application's logic and the interaction between various components.

I will demonstrate the operations of several system functionalities by employing flowchart diagrams. To substantiate my claims, I will present visual evidence in the form of screenshots depicting the flowcharts that have been created:

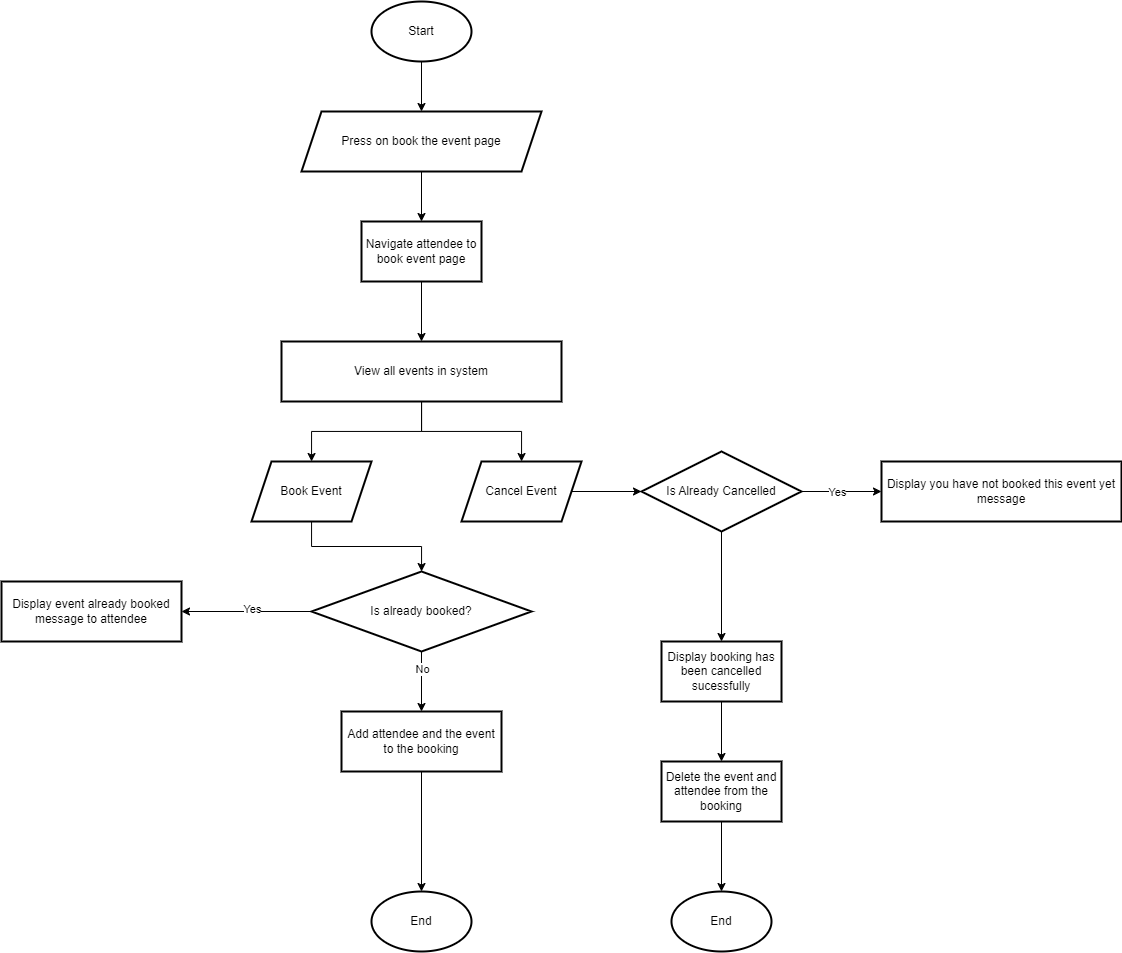
* Organizer register page:



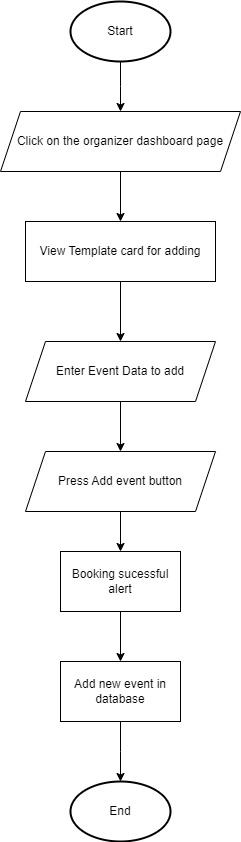
* Attendee register page:



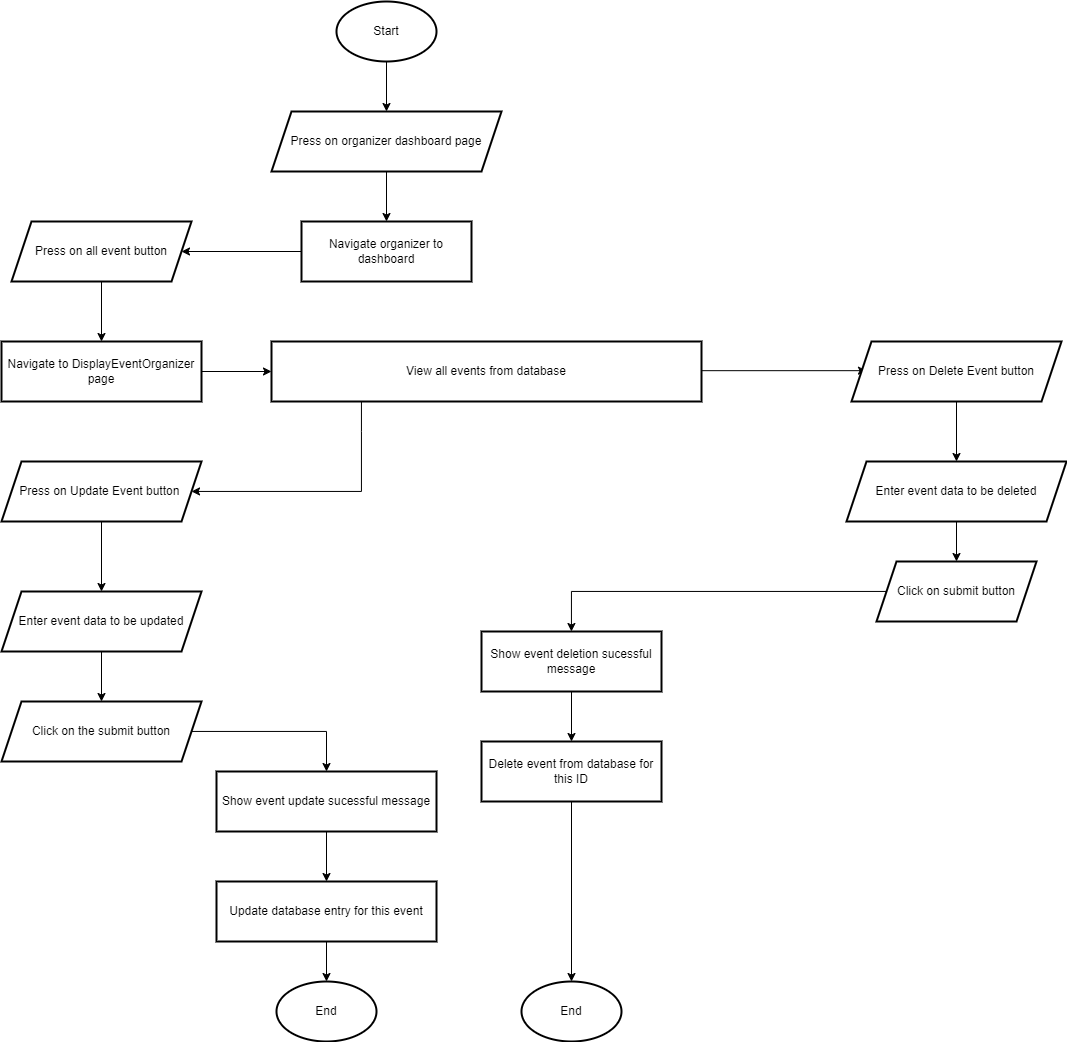
* Book events page for attendees:



* Organizer page to add events:



* Organizer page to view, edit, and delete events:



e) **Effective testing techniques for EMS for XYZ events:**

In the dynamic landscape of event management, the creation of an efficient and reliable Event Management System (EMS) is paramount to the success of companies like XYZ Events. To ensure the seamless orchestration of events, XYZ Events recognizes the critical role of comprehensive testing techniques in validating the functionality, reliability, and performance of their EMS. By leveraging a tailored suite of testing methodologies, XYZ Events aims to fortify their EMS against potential pitfalls, ensuring it not only meets but exceeds the expectations of organizers and attendees alike. This article delves into a selection of effective testing techniques meticulously crafted to align with the intricate requirements and functionalities of XYZ Events' EMS, offering insights into their application and significance in ensuring the robustness of the system. Through a judicious combination of unit testing, performance testing, Blackbox testing, Whitebox testing, and database testing, XYZ Events endeavors to refine their EMS into a seamless, user-centric platform that sets new standards in event coordination and management.

In the process of developing XYZ Events' Event Management System (EMS), the use of various testing techniques is essential to ensure the reliability, functionality and performance of the system. Below are some effective testing techniques tailored to the requirements and functions of an EMS:

Unit Testing: Unit testing involves testing individual components or modules of a system to ensure that they work properly separately. For an EMS, unit testing can be applied to important functions such as user account creation, event creation and management, attendance registration, and ticketing. Each module, such as account creation, should be tested to verify that it handles input validation, authentication, and navigation correctly. For example, unit tests can verify organizers' ability to successfully create accounts, log in securely, and navigate to the appropriate sections of the system.

For unit testing in EMS for XYZ Events, in the user account creation process. Unit testing could focus on ensuring that when an organizer creates an account, the system correctly validates the data entered, authenticates the user, and moves them to the appropriate sections upon successful login. The test case will simulate the organizer providing valid information (name, email, phone number) and verifying that the account creation process is completed without errors. Additionally, it will confirm that on subsequent login attempts using the correct credentials, the system properly authenticates the user and redirects them to the home screen or dashboard.

Performance Testing: Performance testing evaluates the responsiveness and stability of the EMS under various conditions. Given the importance of smooth operation in event management, performance testing is crucial to identify potential bottlenecks and improve system performance. For XYZ Events' EMS, performance testing can focus on actions such as event creation, attendance registration, and data retrieval. Tests can measure a system's response time, throughput, and resource usage to ensure it meets performance requirements. For example, performance testing can verify that the system displays event listings immediately and processes attendance registrations efficiently without delay.

Performance testing in EMS for XYZ Events can target the response of the event generation function. A performance testing scenario can include simulating multiple concurrent users generating events simultaneously to evaluate the system's ability to handle peak loads. Testing metrics will include measuring system response time to event generation, throughput (number of events generated per unit time), and resource usage (CPU, memory usage). By analyzing these metrics, developers can identify any performance bottlenecks and optimize the system to ensure events are generated promptly without causing delays or system crashes.

Blackbox Testing: Blackbox testing examines the functionality of an EMS without considering its internal implementation. This technique evaluates the system's inputs and outputs to ensure they conform to expected behavior. For XYZ Events' EMS, black box testing can validate user interactions such as event creation, attendance registration, and ticket purchasing. Test cases can simulate user actions, verify that the system responds appropriately, navigate users through the interface, and process requests accurately. Blackbox testing can uncover usability issues, ensuring that EMS delivers a seamless experience for organizers and attendees.

For black box testing, consider testing the attendee registration process in XYZ Events' EMS. A black box test case will focus on validating user interactions without considering internal implementation details. Testing scenarios will include simulating various user inputs during registration, such as providing valid and invalid personal details, selecting events, and selecting ticket types. Testing will verify that the system responds appropriately to each input, guides users through the registration flow and accurately records attendance information into the database. This testing approach helps uncover usability issues and ensures a smooth experience for attendees.

Whitebox Testing: Whitebox testing delves into the internal structure and logic of an EMS to evaluate its correctness and efficiency. This technology examines the code base, algorithms, and data structures to identify flaws or potential improvements. In the context of EMS for XYZ Events, Whitebox testing can evaluate the backend logic responsible for handling event management, user authentication, and database interactions. By reviewing the code base and performing code inspections, developers can ensure that the EMS follows best practices, adheres to coding standards, and maintains data integrity.

Whitebox testing in EMS for XYZ Events can examine the backend logic responsible for event management processes. For example, a white box test case could analyze the function of deleting a codebase handling event. The test will examine the code to ensure that when an administrator deletes an event, the system correctly updates the database by removing the event entry and its associated data (for example, attendance records). In addition, the test will check the error handling mechanisms to ensure that the system behaves safely in case of exceptions or unexpected scenarios during the event deletion.

Database Testing: Database testing verifies the integrity, reliability, and performance of EMS database operations. Since attendance and event data are important components of the system, database testing is necessary to ensure data consistency and accuracy. For XYZ Events' EMS, database testing can verify that CRUD (Create, Read, Update, Delete) operations work properly, data constraints are enforced, and database transactions are handled effectively. Tests can validate event information retrieval, attendance registrations, and ticket purchases to ensure accurate data storage and retrieval.

Database testing for XYZ Events' EMS system can focus on validating CRUD operations and data integrity. A database testing scenario can include testing retrieving event information from the database. The test case will query the database to fetch the event data and compare it with the expected results based on the pre-defined test data. Additionally, testing will verify that CRUD (Create, Read, Update, Delete) operations for events and attendance recordings are working properly without causing data inconsistencies or integrity violations. By ensuring the reliability and accuracy of database operations, XYZ Events can maintain data consistency and provide a robust event management ecosystem.

Therefore, through these uses of a range of testing techniques, XYZ Events can thoroughly validate and verify the work carried out in its EMS, identify any potential issues and ensure that the system meets the requirements and expectations of both organizers and attendees.

f) **Frontend**: The frontend refers to the part of the software or application that users interact with directly. It encompasses the user interface (UI) components, layouts, and features that users see and interact with when using the system. In the context of XYZ Events' Event Management System (EMS), the frontend would include elements such as the user interface for event creation, attendee registration forms, event listings, navigation menus, and any other visual components that organizers and attendees interact with. The frontend is responsible for presenting data and enabling user interactions in a visually appealing and intuitive manner. Technologies commonly used in frontend development include HTML, CSS, JavaScript, and frontend library and frameworks like React, Angular, or Next.

**Frontend Technologies:**

**HTML**: HTML (Hypertext Markup Language) serves as the foundation of web pages, providing the structure and content of the EMS interface. By leveraging HTML, XYZ Events can ensure the creation of structured and semantically meaningful web pages. HTML facilitates the organization of content elements such as headers, paragraphs, forms, and lists, enabling a clear and intuitive user interface (UI). Additionally, HTML supports the integration of multimedia elements like images and videos, enhancing the visual appeal and interactivity of the EMS interface. With HTML, XYZ Events can design user-friendly pages that are accessible across different devices and platforms, ensuring a seamless experience for organizers and attendees alike.

**CSS**: CSS (Cascading Style Sheets) plays a crucial role in styling and formatting the visual presentation of HTML elements within the EMS interface. By implementing CSS, XYZ Events can customize the appearance of various UI components, including fonts, colors, layouts, and animations. CSS empowers XYZ Events to create visually appealing and cohesive designs that align with their brand identity and enhance user engagement. Moreover, CSS enables responsiveness, allowing the EMS interface to adapt dynamically to different screen sizes and resolutions, optimizing the user experience across desktops, tablets, and mobile devices. Through the strategic use of CSS, XYZ Events can elevate the aesthetics and usability of their EMS, leaving a lasting impression on organizers and attendees.

**JavaScript**: JavaScript serves as the backbone of dynamic and interactive functionalities within the EMS interface. By leveraging JavaScript, XYZ Events can enhance user engagement and interactivity by implementing features such as real-time updates, form validations, and event-driven interactions. JavaScript enables XYZ Events to create dynamic UI elements that respond to user actions, providing instant feedback and improving usability. Additionally, JavaScript frameworks like React can be utilized to streamline the development process and build complex UI components efficiently. With JavaScript, XYZ Events can enrich the EMS interface with seamless navigation, responsive design, and interactive elements, ensuring a delightful experience for users.

**Backend**: The backend, also known as the server-side, refers to the part of the software or application that runs on the server and is responsible for handling data storage, processing, and business logic. In the context of XYZ Events' EMS, the backend would include functionalities such as user authentication, event management, data storage and retrieval, and communication with external services or databases. The backend processes requests from the frontend, performs necessary operations, retrieves data from the database, and sends responses back to the frontend. Technologies commonly used in backend development include programming languages like PHP, Python, Ruby, or Node.js, as well as frameworks like Laravel, Django, Ruby on Rails, or Express.js. Additionally, databases such as SQL or NoSQL databases are often used to store and manage data in the backend.

**Backend Technologies:**

**SQL**: SQL (Structured Query Language) is a powerful tool for managing and querying relational databases, serving as the backbone of the EMS backend infrastructure. By utilizing SQL, XYZ Events can design and maintain a robust database schema to store and retrieve event-related data efficiently. SQL enables XYZ Events to perform complex database operations such as data insertion, retrieval, updating, and deletion, ensuring the integrity and consistency of attendee and event information. Moreover, SQL supports the implementation of advanced database features like transactions, constraints, and indexes, optimizing performance and reliability. With SQL, XYZ Events can build a scalable and secure backend system that meets the evolving needs of event management.

**PHP**: PHP (Hypertext Preprocessor) serves as the server-side scripting language for dynamic web application development, providing the logic and functionality of the EMS backend. By leveraging PHP, XYZ Events can handle user requests, process form submissions, and interact with the database to perform CRUD (Create, Read, Update, Delete) operations. PHP enables XYZ Events to implement business logic and server-side validation, ensuring data integrity and security. Additionally, PHP frameworks like Laravel can be utilized to expedite development and maintainability, offering features such as routing, authentication, and session management. With PHP, XYZ Events can build a scalable and performant backend system that powers the EMS with efficiency and reliability

**Activity 2:**

**Solution:**

**1.** I will introduce and explain the different categories of software development tools, including operating systems, integrated development environments (IDEs), and charting software tools. I will also delve into different Software Development Life Cycle (SDLC) methodologies to find the most appropriate approach for my EMS project:

First, the operating system: This is an important system program that has the ability to take care of the hardware and manage it effectively along with the device software. The operating system also contains many applications and services that any user can use. For an EMS system, there are a few operating systems that can be used to develop the system and be able to maintain it properly. Operating systems that can be used include:

Therefore, operating systems are an integral part of the functioning of any computerized system, as they act as the main backbone that supports all operations and other software and hardware functions. It also ensures that different programs can run simultaneously without conflicts, manages system resources efficiently and effectively, and provides easy-to-use interfaces. In the context of an EMS system, choosing the right operating system is crucial to improve performance, ensure reliability, and facilitate easy maintenance. The choice of operating system can greatly impact the system's efficiency, security, and overall user experience.

When developing an environmental management system, it is important to take into account the specific needs and constraints of the project. Factors such as compatibility with existing infrastructure, the technical expertise of the development team in how to maintain it, and the specific functionality required by the EMS system should guide the choice of operating system. Each operating system offers distinct advantages and potential disadvantages, making it necessary to carefully evaluate them to ensure the chosen platform aligns with project objectives and operational requirements.

Windows: This is the default operating system that can be used to develop and maintain the system in a good and efficient way. Windows itself provides a lot of great, important and efficient services along with having a clean and nice-looking user interface which is also easy to use for the users. With Windows, the process can be much simpler. Windows is definitely useful for developing projects and working on them strategically, effectively and with high efficiency.

Windows is a multi-tasking operating system, known for its easy-to-use interface and comprehensive support for a wide range of applications and services. For EMS, using Windows can simplify the development process, thanks to its comprehensive development tools and broad compatibility. The availability of integrated development environments (IDEs) and comprehensive documentation makes Windows a practical choice for both novice and experienced developers. Moreover, regular Windows updates and robust security features can help effectively maintain system health over time.

In addition, Windows provides comprehensive support for third-party software and hardware, making it easier to integrate various components into an EMS system. The widespread use of the operating system in enterprise environments means that there is a wealth of community and professional support available, which can be invaluable during the development and maintenance phases. The Windows graphical user interface (GUI) can also improve the user experience, making the EMS system more intuitive and effective for users with different levels of technical expertise.

Secondly, Linux: This is a lightweight operating system, meaning that it is well optimized and has the bare minimum that the operating system actually needs, and for any additional add-on that the developer may need for the project, he/she will just have to install it and set up themselves. Compared to Windows, Linux in its various distributions can be quite difficult to use, however, it also offers an aesthetically pleasing user interface. Linux is very suitable for programming and project development due to its well-optimized nature.

Linux also stands out as a highly efficient, effective and customizable operating system, and is ideal for EMS due to its flexibility and improved performance. Its lightweight nature means that it consumes fewer system resources, which can result in faster performance and greater stability, especially for systems that require high reliability. The ability to customize a Linux system to suit specific project needs and requirements by installing only the necessary components can result in a more secure and streamlined system, which certainly reduces potential vulnerabilities and unnecessary overhead.

Furthermore, Linux is widely known for its strong security capabilities, features, and open-source nature, making it particularly useful for EMS. The open-source community offers a wide range of tools and resources that can be used to improve system capabilities. Although Linux may take more effort to understand than Windows, its capacity for more control and customization makes it a formidable choice for developers who are willing to take the time necessary to fully understand its intricacies. The wide range of Linux distributions (distributions) allows developers to choose the one that best suits their project's needs, providing the ideal combination of performance, ease of use, and security.

**As a developer for the EMS System**, I personally chose to use the Windows operating system because it is easy to use, inexpensive, and widespread popularity. Additionally, Windows allows for smoother and faster downloading and utilization of software tools compared to Linux.

So, I will talk about integrated development environments (IDEs), also known as integrated development environments, which are comprehensive software applications that provide developers with the tools they need to write, test, and debug code efficiently. Integrated development environments (IDEs) provide various features such as syntax highlighting, code completion, and integrated compilers, which translate human-readable code into machine-readable binary code. These environments are essential for enhancing productivity and ensuring high-quality code. Common IDEs include:

IDEs are also indispensable in modern software development, as they provide a unified workspace where developers can manage their projects seamlessly. By integrating multiple functions such as text editing, debugging, and version control into a single platform, an IDE simplifies the development process. This integration reduces the time spent switching between different tools, allowing developers to focus more on programming and problem solving. The choice of IDE can greatly impact the efficiency and ease of development, making it a critical decision for any project.

For EMS, choosing the right IDE can greatly improve your development experience. The IDE should support the programming languages and frameworks used in the project, offer powerful debugging tools, and integrate well with version control systems. Moreover, it should provide a smooth and intuitive user interface to reduce the learning curve for new developers joining the project. With a variety of IDEs available, each with unique strengths, it is important to choose one that best matches the specific needs of the project and the development team's workflow.

Visual Studio Code: Visual Studio Code, or VS Code, is one of the most popular integrated development environments (IDEs) for developing projects efficiently and effectively and is most popular with users. This IDE provides a wide range of tools and plugins that enhance its functionality, allowing users to extensively customize their development environment. These plugins range from theme changing tools to language-specific support, making VS Code a versatile option. Furthermore, it is lightweight, which contributes to its great performance and ease of use, making it an excellent choice for both large and small projects.

VS Code's popularity stems from its versatility and extensive customization options. Developers and programmers can customize the IDE to suit their own needs by installing extensions that add functionality, such as advanced debugging tools, code analysis systems, and integrated version control. Its lightweight nature ensures that it runs smoothly even on resource-constrained systems, which is especially useful for developing complex systems such as EMS. The active community around VS Code continually contributes to its add-on marketplace, ensuring a constant flow of new tools and improvements.

In addition, VS Code's integration with cloud services and container platforms like Docker enhances its appeal for modern development practices. These integrations facilitate seamless deployment and testing of the EMS system in different environments, ensuring consistency and reliability. Strong support for multiple programming languages and frameworks establishes VS Code as a powerful and versatile tool, making it an ideal choice for EMS system development.

IntelliJ IDEA Ultimate: IntelliJ IDEA Ultimate is another highly regarded IDE, preferred by many developers and programmers due to its powerful features and intuitive interface. Similar to VS Code, it offers a wide range of tools and plugins that improve productivity and facilitate efficient development. Although heavier than VS Code, IntelliJ IDEA Ultimate provides extensive services and advanced features that can meet even the most complex project requirements, making it a strong contender for any development project.

IntelliJ IDEA Ultimate also excels in providing easier and more effective intelligent code help, refactoring tools, and comprehensive debugging capabilities. Its deep integration with build tools and version control systems ensures a seamless development workflow. For EMS, the advanced features of IntelliJ IDEA Ultimate can be particularly useful, providing developers with the tools to efficiently manage large code bases and complex dependencies. The IDE's powerful performance analysis tools help optimize system performance, ensuring it runs smoothly and efficiently.

Furthermore, IntelliJ IDEA Ultimate supports a wide range of programming languages and frameworks, making it a versatile choice for projects involving diverse technologies. Powerful search and navigation features allow developers to quickly locate and manage code across large projects, boosting productivity and reducing development time. These features, combined with strong community support and regular updates, make IntelliJ IDEA Ultimate a formidable tool for developing sophisticated systems like EMS.

As an EMS system developer, I chose to use the Visual Studio Code IDE to develop a web application. This decision is based on the lightweight nature of VS Code, extensive customization options, tools that can help me build large projects in efficient ways, and strong support for a wide range of programming languages and frameworks. An active community and constant updates ensure that VS Code remains a cutting-edge tool, providing all the features and functionality needed to efficiently develop and maintain an EMS system.

Diagram software tools: which are the basic tools that enable developers and programmers to create diagrams and design systems or applications effectively and more efficiently. By allowing the project to be visualized and visualized before actual development begins, these tools help in planning and understanding the system architecture and workflow. Diagramming software tools are invaluable for drawing system components and their interactions, ensuring a clearer and more organized development process. Many of these tools include features that simplify the process of creating different types of diagrams, which can be shared and collaborated on within the development team. Here are some notable diagramming software tools that can be used to develop an EMS system:

Draw.io: This free drawing tool is widely used to design and visualize systems effectively. Draw.io offers an easy-to-use interface and a comprehensive set of features that allow developers to create detailed diagrams at no cost. It supports a variety of diagram types, including flowcharts, network diagrams, and UML diagrams, making it versatile for different project needs. Its cloud integration capabilities also make it easy to share and collaborate on designs with team members.

In addition to being free, Draw.io is an open-source tool, which means it's constantly being improved by a community of developers. This ensures that it stays up to date with the latest features and security standards. Its integration with popular cloud services like Google Drive, OneDrive, and GitHub enhances its collaborative capabilities, allowing team members to work together in real-time. For an EMS system, using Draw.io can streamline the planning and design stages, ensuring that all stakeholders have a clear understanding of the system architecture.

Visual Model: This diagramming software tool is known for its powerful features that help developers understand and design systems efficiently. Visual Paradigm offers a range of functions, including UML, ERD, and BPMN diagramming, that are essential for complex system design. It provides an intuitive interface that helps in creating organized and visually attractive diagrams, which can greatly assist in the actual development phase by providing clear guidelines and structure.

Visual Paradigm also offers different versions, including the free Community Edition and affordable paid versions that provide additional features and support. Its extensive library of templates and examples can be especially useful for developers new to system design. Furthermore, Visual Paradigm supports integration with various development tools and platforms, making it easy to integrate diagrams into the overall project workflow. For an EMS system, a visual model can enhance the design process by providing detailed, professional diagrams that clearly illustrate the system components and their interactions.

Lucidchart: This diagramming software tool is highly regarded for its ease of use and powerful collaborative features. Lucidchart allows developers to quickly create and share diagrams, facilitating better communication and understanding between team members. It supports a wide range of diagram types, including flowcharts, wireframes, and network diagrams, making it a versatile tool for system design. Lucidchart's real-time collaboration features ensure that all team members can contribute to and stay on top of the design process.

In addition to its easy-to-use interface, Lucidchart integrates seamlessly with many popular productivity and collaboration tools like Google Workspace, Microsoft Office, and Slack. This makes it easy to integrate diagrams into existing workflows and boosts team productivity. For an EMS system, Lucidchart can help ensure the design phase is efficient and that the resulting diagrams are clear and comprehensive, aiding in accurate system development.

Personally, I prefer using Visual Paradigm and Draw.io for diagramming in EMS system development. These tools offer a combination of powerful features, ease of use, and affordability that make them ideal for creating detailed and accurate system designs. Using these tools, I can ensure that the design phase is thorough and that the diagrams produced provide clear guidance to the development team, ultimately leading to a more organised, successful and understandable project.

SDLC Methodologies: The Software Development Life Cycle (SDLC) methodology defines the approach taken in managing a software project through its various stages: planning, analysis, design, implementation, testing, and maintenance. Each methodology also offers unique strategies for navigating these stages, offering distinct advantages, disadvantages, and risks. Choosing the right SDLC methodology is crucial to an effective and smooth project success. Here we explore some common methodologies before choosing the most appropriate EMS methodology:

SDLC methodologies provide an organizing framework that guides project teams through the complexities of software development. It also helps ensure that all aspects of a project are systematically addressed and analyzed, from initial concept to final deployment and beyond. Different methodologies meet different project needs and team dynamics, making it essential to understand their specific features and how they align with project goals. In this discussion, we will delve into the characteristics of several popular methodologies and evaluate their suitability for the EMS system.

Agile: In short, it divides the project into small, manageable units called sprints, each of which lasts a few weeks. During each sprint, a cross-functional team works on different aspects of the system, allowing for continuous delivery of parts of the software. This iterative approach, combined with continuous and effective customer feedback, enables rapid adaptation to changes and ensures that the final product closely matches user requirements.

Agile's flexibility and focus on collaboration with customers make it particularly suitable for projects with dynamic requirements. Its recurring cycles allow for frequent re-evaluation and modification, enhancing the project's ability to meet evolving needs. However, Agile success relies heavily on active customer engagement and a dedicated team, which can increase complexity and cost. The need for regular meetings and ongoing feedback loops requires significant commitment from all stakeholders, but the resulting adaptability and user satisfaction often justify these demands.

Waterfall model: The waterfall model is one of the simplest, most traditional, and common SDLC methodologies. It follows a linear and sequential approach where each stage must be completed before the next stage can begin. This straightforward process is easy to understand and manage, because it clearly defines every step from requirements gathering to deployment. However, its strict structure means that any changes or issues that arise midway through the process cannot be addressed until the entire cycle is complete and must be revisited from the beginning.

The simplicity of the waterfall model can be both an advantage and a disadvantage. For live projects with well-defined requirements, this methodology can be very effective. However, its inflexibility makes it less suitable for projects whose requirements may evolve over time. This lack of adaptability can lead to significant delays and increased costs if changes are needed beyond the initial stages. Despite these limitations, the clarity and predictability of the waterfall model makes it a reliable choice for projects with stable and clear requirements.

Iterative model: The iterative model improves the waterfall model by allowing stages to be repeated in iterations. Each iteration goes through the full SDLC process, addressing a subset of the overall requirements. This method reduces risk by allowing developers to improve and expand the system through iterative cycles, incorporating feedback and making modifications as necessary. It is especially useful for large projects where requirements may change or become clearer over time.

The flexibility of the iterative model and focus on incremental development also helps in managing complexity and reducing project risks. By breaking the project down into smaller, manageable iterations, it becomes easier to identify and correct problems early in the development cycle. This approach also facilitates ongoing customer feedback, ensuring that the final product best meets the user's needs. However, for smaller projects or those with well-defined requirements, the benefits of the iterative model may not justify its more complex process and potentially higher costs.

I will choose a methodology for EMS which is Agile, which for EMS, the Agile model stands out as the most suitable choice. His ability to adapt and focus on continuous improvement aligns well with the project's potential needs for flexibility and responsiveness to changing requirements. Agile iterative cycles and regular feedback loops ensure that the development process remains dynamic and user-focused, making it best suited for EMS system development.

**2. Solution:** Now, I will talk in more detail about why I chose the above tools and, in this discussion, we will compare and justify the choices of operating systems, integrated development environments (IDEs) and diagramming software tools., and justify it in a reasonable, logical and justified way. Therefore, when developing an Environmental Management System (EMS), choosing the right tools and methodologies is crucial to ensuring the success of the project. The tools selected should match the project requirements, the expertise of the development team, and the specific needs of the EMS system. Here, I'll explain in detail the reasons for choosing Windows as the operating system, Visual Studio Code as the integrated development environment (IDE), and Draw.io and Visual Paradigm for the drawing software. In addition, I will justify the choice of Agile methodology for the development process:

**Operating systems**

**Windows**:

Windows is a widely used operating system, known for its easy-to-use interface and comprehensive support for applications and development tools. It provides several advantages to an EMS project:

* User-friendly interface and accessibility: Windows provides an intuitive and visually attractive interface, simplifying navigation and operation. This ease of use is beneficial for developers and end users, reducing the learning curve and enhancing productivity. Windows familiarity with a broad user base can also facilitate smoother transitions and user acceptance of the EMS system.
* Comprehensive support and compatibility: Windows supports a wide range of third-party software and hardware, making it easy to integrate different components into an EMS system. The comprehensive documentation and community support available for Windows is invaluable during the development and maintenance phases, ensuring that issues can be resolved quickly.
* Robust security and regular updates: Regular updates from Microsoft ensure Windows stays secure and up-to-date, protecting the EMS system from vulnerabilities and improving overall system health. This ongoing support helps keep the system reliable and secure over time.

**Linux**:

Linux, on the other hand, is known for its lightweight nature, customizability, and strong security features. Linux offers the following advantages:

* Optimized performance and efficiency: Linux is highly efficient, consuming fewer system resources, which can lead to faster performance and greater stability. This is especially useful for systems that require high reliability and uptime, such as EMS.
* Customizability: Linux allows developers to install only the necessary components, creating a more secure, streamlined system tailored to a project's specific needs. This level of customization can reduce potential vulnerabilities and unnecessary overhead.
* Strong Security: Linux's open-source nature means that it is constantly being improved by a community of developers. This ensures that security patches and updates are readily available, making Linux a strong choice for systems that prioritize security.

For EMS, I chose Windows as the operating system because of its ease of use, comprehensive support, and strong security features. Familiarity with and accessibility of Windows can facilitate smoother development and maintenance processes, ensuring that the EMS system is easy to use and reliable.

This is because among all other operating systems, Windows is very easy, very cheap and also has a lot of tools and software that can help in the project. One of these tools is for example the ability to back up project data and more. As a developer, Windows has a nice feel, is very easy to use and does not require any user to install certain things in order to be able to use the system better. Unlike Linux, for example, Windows does not rely on CMD or Terminal to install programs or tools. Rather, the user can do it from the Internet and it will be installed easily. A developer can certainly take advantage of CMD as well, but the OS doesn't rely on it quite as much as Linux. In addition, Windows is also very cheap. Depending on all of these different variables and comparisons, we can see that Windows is the better choice for EMS.

**Integrated development environments (IDEs)**

**Visual Studio Code (VS Code):**

Visual Studio Code (VS Code) is a popular integrated development environment (IDE) known for its versatility and extensive customization options. For an EMS project, VS Code offers several advantages:

* Lightweight and Efficient: VS Code is a lightweight IDE, ensuring it runs smoothly even on resource-constrained systems. This efficiency is particularly useful for developing complex systems such as EMS, where performance is critical.
* Comprehensive customization and plug-in support: Developers can customize VS Code to suit their needs by installing extensions that add functionality, such as advanced debugging tools and many extensions that may help developers in building EMS projects, code analysis systems, and integrated control. Release. This flexibility allows for a custom development environment that improves productivity.
* Strong Community and Continuous Improvement: The active community around VS Code constantly contributes to its mentorship marketplace, ensuring a constant flow of new tools and improvements. This support helps keep the development environment up to date with the latest features and best practices.

**IntelliJ IDEA Ultimate:**

IntelliJ IDEA Ultimate is another highly regarded IDE, preferred by many developers due to its powerful features and intuitive interface. For an EMS project, IntelliJ IDEA Ultimate offers:

* Advanced Features and Services: IntelliJ IDEA Ultimate provides intelligent code help, refactoring tools, and comprehensive debugging capabilities. These features can greatly improve development efficiency and code quality, especially for large and complex projects such as EMS.
* Integration with development tools: The IDE's deep integration with build tools and version control systems ensures a seamless development workflow. This integration is essential to manage the complexity of EMS development, ensuring good coordination of all parts of the project.
* Versatility and comprehensive language support: IntelliJ IDEA Ultimate supports a wide range of programming languages and frameworks, making it a versatile choice for projects involving diverse technologies. This broad support can be particularly useful for EMS, which may require different programming languages and tools.

IDE selection for EMS

For EMS, I chose Visual Studio Code (VS Code) as my IDE due to its lightweight nature, extensive customization options, and strong community support. The versatility and efficiency of VS Code makes it an ideal choice for EMS system development and maintenance.

IDE: For the previous question, you have chosen Visual Studio Code as the Integrated Development Environment (IDE) to develop the EMS system. One reason is that VS Code provides a fun and convenient programming experience. Furthermore, the platform offers a wide range of tools and plugins that developers can easily use, providing convenience by placing them all centrally in one place. Additionally, when compared to competing integrated development environments (IDEs) such as IntelliJ Idea Ultimate, Visual Studio Code is characterized by its lightweight nature, minimal resource requirements, and excellent performance during programming tasks. IntelliJ Idea Ultimate contains a wide range of tools and plug-ins, providing users with many alternatives. However, in terms of weight, it is much heavier than VS Code. Additionally, with VS Code, a developer can work with any programming language and any code by ensuring that the appropriate language plugin and language pack are loaded and configured correctly. Although IntelliJ Idea Ultimate has this feature, it is not as good in terms of user experience and efficiency as VS Code. Given Visual Studio Code's lightweight nature and comprehensive toolset, as well as its compatibility with many programming languages, I chose VS Code as my IDE for EMS development.

**Software tools for diagramming**

**Draw.io:**

Draw.io is a free, open-source diagramming tool widely used for designing and visualizing systems. For the EMS project, Draw.io provides:

* Ease of Use and Comprehensive Features: Draw.io provides an intuitive interface and a wide range of features that allow developers to create detailed diagrams. Its support for different diagram types, including flowcharts, network diagrams, and UML diagrams, makes it versatile for different project needs.
* Cloud Integration and Collaboration: Draw. Io’s integration with popular cloud services like Google Drive, OneDrive, and GitHub enhances its collaboration capabilities. This allows team members to work together in real-time, ensuring designs are constantly updated and accessible.
* Open Source and Continuous Improvement: As an open-source tool, Draw.io benefits from continuous improvements and updates from the community. This ensures that it stays up to date with the latest features and security standards.

**Visual Paradigm:**

Visual Paradigm is another powerful diagramming tool known for its powerful features and ease of use. For an EMS project, Visual Paradigm offers:

* Extensive diagramming capabilities: The visual model supports a range of diagram types, including UML, ERD, and BPMN diagrams, necessary for complex system design. Its intuitive interface helps create organized and visually attractive diagrams, providing clear development guidelines.
* Integration with development tools: The visual model integrates well with different development tools and platforms, making it easy to integrate diagrams into the overall project workflow. This integration ensures that designs are constantly aligned as development progresses.
* Community Support and Professional Support: Visual Paradigm offers different versions, including the free Community Edition and affordable paid versions with additional features and support. This range of options makes it accessible to different project needs and budgets.

I have selected the Diagramming Software tool Visual Paradigm and Draw.io The reason is that although draw.io is a commendable and cost-free application used for designing and drawing system diagrams, the resulting diagrams lack the realistic look and feel that Visual Paradigm offers. In addition, Visual Paradigm offers a free version as well as other paid versions. However, the free version is very easy to use, and the cost of the other versions is somewhat affordable for developers. The Visual Model tool displays a superior aesthetic quality overall, with its attractive and visually pleasing alternatives for each diagram.

It provides a convenient and attractive user experience for developers. In addition, it can be concluded that the Visual Paradigm tool is the most suitable software tool for EMS system diagramming. Additionally, I will use draw.io to create flowcharts for the specific web application.

For EMS, I chose Draw.io and Visual Paradigm due to their ease of use, comprehensive features, and cloud integration capabilities. The open-source nature of Draw.io and ongoing community support ensures that it remains a powerful and modern tool for EMS system design and visualization.

**Software Development Life Cycle (SDLC) Methodologies**

**Agile:**

The Agile methodology is characterized by its iterative and flexible approach, dividing the project into small, manageable units called sprints. For the EMS project, Agile provides:

* Flexibility and Adaptability: Agile’s iterative cycles allow for continuous improvement and adaptation to changing requirements. This flexibility is crucial for EMS, where requirements may evolve over time.
* Customer Collaboration: Agile emphasizes regular customer involvement and feedback, ensuring that the final product closely aligns with user needs. This collaboration enhances the system’s usability and relevance.
* Reduced Risk: Agile’s focus on incremental development and regular evaluation helps identify and address issues early in the development cycle, reducing overall project risk.

**Waterfall Model:**

The Waterfall model follows a linear and sequential approach, completing each stage before moving on to the next. For the EMS project, Waterfall offers:

* Simplicity and Predictability: The straightforward, step-by-step process of the Waterfall model is easy to understand and manage, providing clear milestones and deliverables at each stage.
* Well-Defined Requirements: The Waterfall model works well for projects with stable and clear requirements, ensuring that each phase is thoroughly documented and understood before proceeding.
* Ease of Management: The linear approach of the Waterfall model makes it easier to manage and monitor progress, ensuring that all aspects of the project are systematically addressed.

**Iterative Model:**

The Iterative model allows for stages to be repeated in iterations, addressing a subset of the overall requirements each time. For the EMS project, Iterative offers:

* Incremental Development: The Iterative model’s incremental approach allows for ongoing improvements and expansions, reducing risk by enabling early problem identification and correction.
* Ongoing Customer Feedback: Like Agile, the Iterative model facilitates ongoing customer feedback, ensuring that the system remains aligned with user needs and expectations.
* Flexibility and Scalability: The Iterative model’s flexibility makes it suitable for large projects where requirements may change or become clearer over time, ensuring that the system can adapt to evolving needs.

As for EMS system, is a reason I chose Agile methodology due to its flexibility, focus on customer collaboration and ability to adapt to changing requirements. Agile's iterative approach and regular feedback loops ensure that the EMS system remains dynamic and user-focused, making it the most suitable choice for this project. Its iterative approach allows for continuous improvement and adaptation based on feedback and changing requirements, reducing the risk of project failure due to strict planning.

Agile emphasizes customer collaboration and feedback, ensuring that the final product meets user expectations. Rapid cycles and regular reviews allow stakeholders to provide input and suggest changes, enhancing the user-centred development process. Agile iterative cycles and incremental delivery reduce the risks associated with the development process by breaking the project into manageable units, identifying potential issues early, and ensuring the development process remains efficient and responsive to changes.

In EMS system development, Agile methodology is applied by dividing the project into small, manageable units called sprints. Each sprint typically lasts a few weeks and focuses on delivering specific features or functionality of the system. For example, in the first sprint, the development team might prioritize basic requirements such as creating a user interface for data entry and setting up the database structure for storing environmental data.

Once the initial sprint is complete, I can conduct a review with stakeholders to gather feedback and identify areas for improvement. Based on this feedback, I adjust the development plan for the next sprint, incorporating any necessary changes or improvements. For example, stakeholders may request additional features such as real-time data visualization or integration, which can be prioritized for future sprints.

As development progresses through successive iterations, more advanced features and functions are added to the EMS system. This iterative approach allows the development team to quickly respond to changing requirements and stakeholder needs, ensuring that the final product meets user and stakeholder expectations.

3. The selection of tools and methodologies for an EMS system is closely interconnected, with the goal of creating a cohesive development environment that maximizes efficiency, effectiveness, collaboration, and adaptability. Let me explain and delve into the rationale behind each choice and identify the connections between them.

First, choosing Windows as an operating system (OS) aligns with the need for an easy-to-use interface, comprehensive support, and strong security features. Windows provides an intuitive platform for developers to work on, reducing the learning curve and enhancing productivity. Its compatibility with a wide range of third-party software and hardware ensures seamless integration of essential tools and components for EMS development. Furthermore, regular updates from Microsoft ensure that the operating system remains secure and up-to-date, protecting the EMS system from vulnerabilities. This choice is reinforced by the Agile methodology, where an easy-to-use Windows interface promotes collaboration and seamless transition between project phases, consistent with Agile's focus on customer collaboration and adaptability.

The choice of Windows as the operating system for EMS development is justified by its ease of use, comprehensive support, and security features. Windows provides a familiar and intuitive interface, which reduces the learning curve for developers and end users alike and provides many tools and programs that have always been available in Windows operating systems. This knowledge can facilitate smoother transitions and user acceptance of the EMS system. Furthermore, Windows' compatibility with a wide range of third-party software and hardware simplifies the integration process, which is critical for EMS development that may require communication with different devices and systems.

On the other hand, choosing Windows is consistent with the principles of the Agile methodology, especially its focus on collaboration and adaptability. The easy-to-use Windows interface enhances collaboration among team members, enabling them to work efficiently on different aspects of the project. Additionally, Windows' robust security features ensure the integrity and confidentiality of EMS data, addressing Agile's focus on risk management.

Secondly, in terms of integrated development environments (IDEs), Visual Studio Code (VS Code) is chosen due to its lightweight nature, extensive customization options, and strong community support. The versatility and efficiency of VS Code makes it an ideal choice for EMS development, enabling developers to design their own development environment to suit their specific needs and also providing many tools and extensions that help developers and programmers create effective and efficient projects. The lightweight nature of VS Code ensures smooth performance, which is essential for handling complex EMS projects. Extensive customization options and plug-in support improve productivity and collaboration, aligning with Agile's focus on agility and customer collaboration. Furthermore, VS Code's compatibility with different programming languages ensures compatibility with the diverse requirements of the EMS system, facilitating iterative development cycles.

Also choose Visual Studio Code (VS Code) as the IDE completes the choice of Windows operating system. VS Code's lightweight nature and extensive customization options make it well-suited for EMS development, ensuring smooth performance and enabling developers to tailor their development environment to suit their specific needs. Its compatibility with different programming languages meets the diverse requirements of EMS development, allowing developers to work seamlessly across different aspects of a project.

The choice of VS Code also aligns with Agile's focus on flexibility and customer collaboration. The versatility of VS Code allows for rapid iteration and adaptation to changing requirements, which is essential for an iterative Agile approach. Additionally, strong community support fosters collaboration and knowledge sharing between developers, reinforcing Agile principles of teamwork and communication.

Third, for diagramming software, Draw.io and Visual Paradigm were chosen for their ease of use, comprehensive features, and cloud integration capabilities. Draw.io provides an intuitive interface and a wide range of features for creating detailed diagrams, essential for visualizing and designing an EMS system architecture. Its cloud integration capabilities enhance collaboration, allowing team members to work together in real-time and ensuring designs are constantly updated and accessible. On the other hand, a visual model provides extensive diagramming capabilities and superior aesthetic quality, making it suitable for designing complex systems. Its integration with development tools and comprehensive language support aligns with EMS system requirements, facilitating seamless integration of charts into the overall project workflow.

So, choosing Draw.io and Visual Paradigm for your diagramming software complements the Agile development process by facilitating visual communication and collaboration. Draw. Io ease of use and comprehensive features allow developers to create detailed diagrams depicting the architecture and design of an EMS system. Its integration with cloud services enhances collaboration, allowing team members to work together in real-time and ensuring designs are constantly updated and accessible.

Visual Paradigm provides extensive diagramming capabilities and professional features that match the complexity of EMS development. Its integration with development tools simplifies workflow, enabling seamless integration of diagrams into the development process. The selection of both Draw.io and Visual Paradigm supports Agile methodology's focus on customer collaboration and continuous improvement, enabling stakeholders to provide feedback on EMS visual representations and facilitate iterative improvement.

Fourth, choosing an Agile methodology enhances the choice of tools by focusing on agility, customer collaboration, and risk management. Agile's iterative approach and regular feedback loops complement the features of Windows, VS Code, Draw.io, and Visual Paradigm, ensuring the development process remains dynamic and responsive to changing requirements. The iterative nature of Agile allows for continuous improvement and adaptation based on feedback and evolving stakeholder needs, aligned with the collaborative and iterative features of the identified tools.

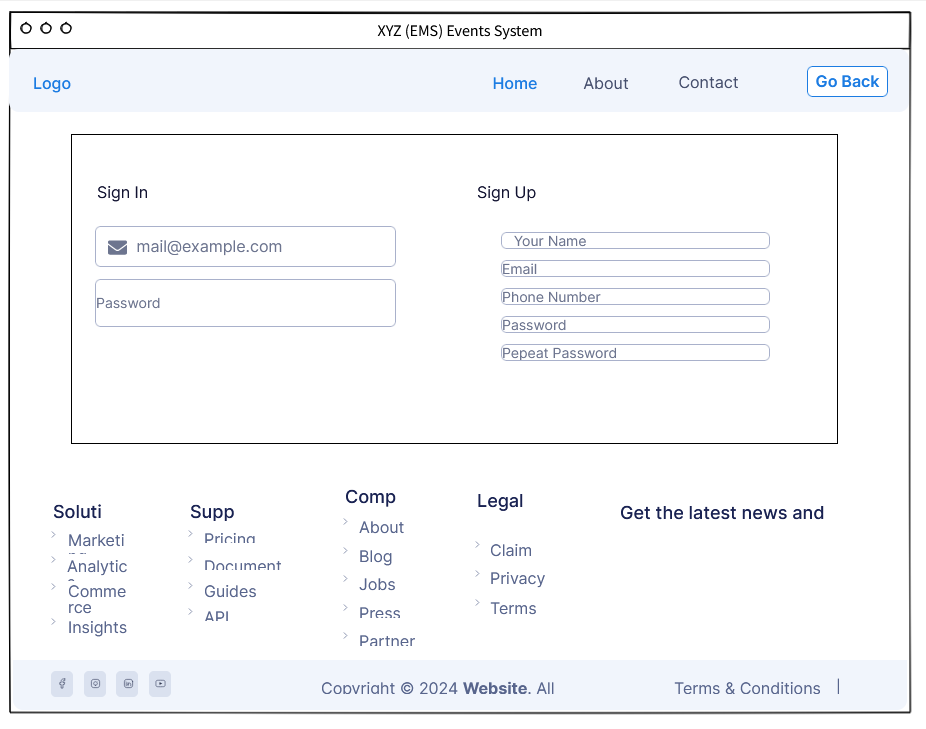
Choosing an Agile methodology also links choices of operating systems, integrated development environments (IDEs), and planning software by providing a framework for effective iterative development, collaboration, and adaptability. Agile's iterative approach allows for continuous improvement and adaptation based on feedback and changing requirements, ensuring that the EMS system remains dynamic and responsive to stakeholder needs. Its focus on customer collaboration enhances communication and alignment between developers and end users, reinforcing the importance and ease of use of the EMS system.

**Activity 3:**

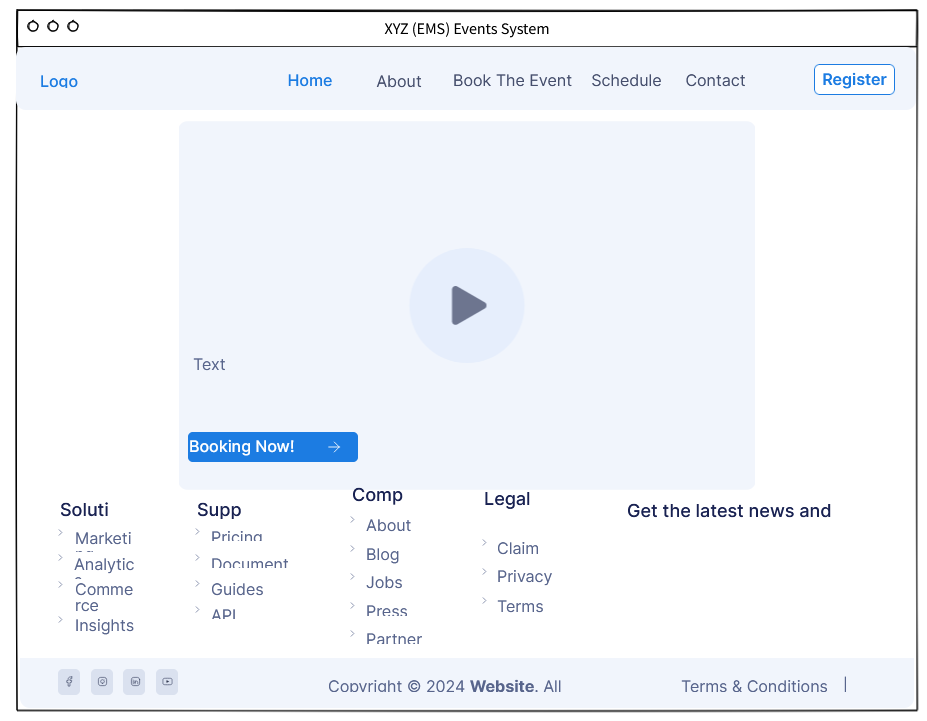
**Solution:**

1. Develop a presentation to assess the following aspects:
2. Business application
3. Problem definition statement
4. Proposed solution that outlines functional and non-functional requirements
5. Development strategy
6. Wireframes:

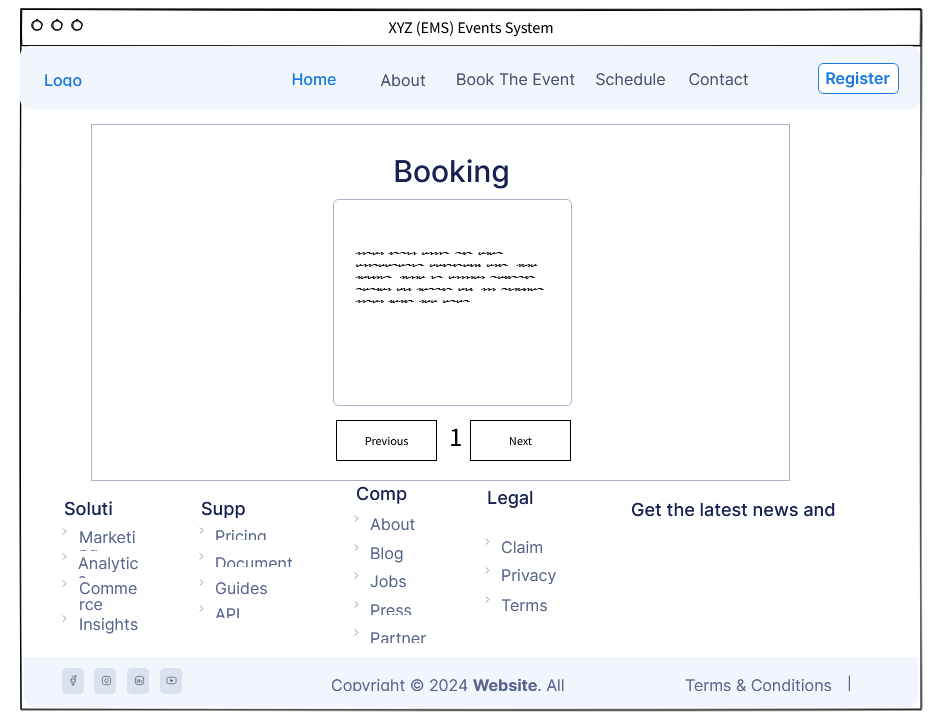
* Sign In, Sign Up page for attendees and organizers:



* Attendee Home Page:



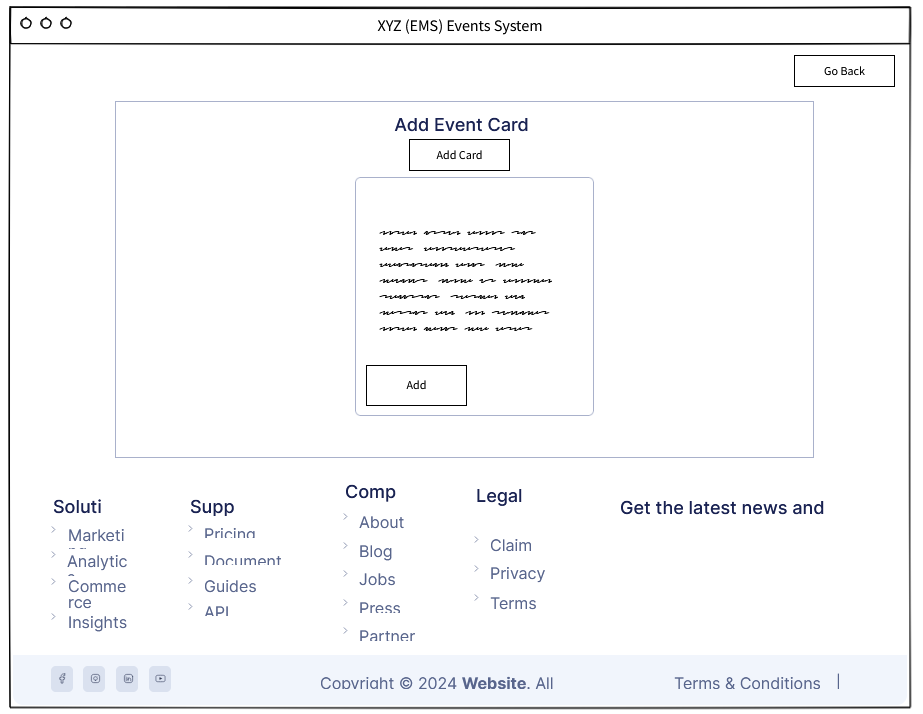
* Event booking page for attendees:



Cancel

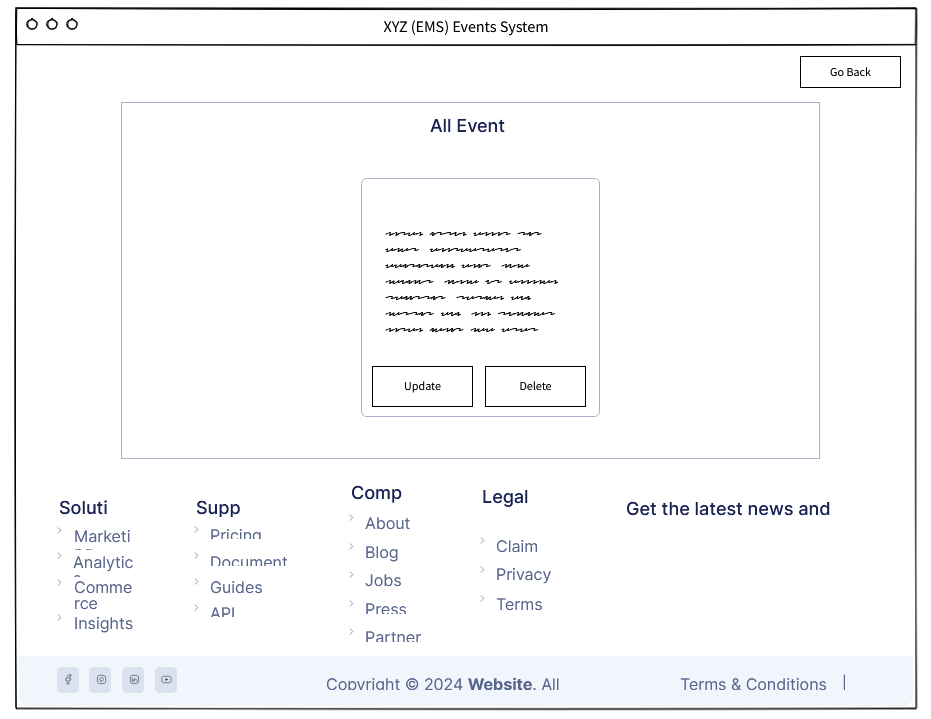
Book

* Add events page for organizers:



All Event

* A page that displays all events for organizers to update or delete:



**2**. Develop the Event Management System (EMS), integrating all requirements identified in activity 1. Additionally, create separate documentation that outlines the development process, providing screenshots and detailed descriptions of each implemented feature.

**3.** I will evaluate the feedback I received through my presentation by explaining and comprehensively evaluating all the feedback and additional areas for exploration and development:

**First:**

**Feedback: Why the Agile SDLC methodology and not the iterative model?**

The Agile SDLC methodology was chosen over the iterative model due to its inherent flexibility, customer-centric approach, and ability to handle changing requirements efficiently. Agile focuses on iterative development through sprints, where each sprint delivers a potential shippable product increment. This allows for continuous feedback and adaptation, ensuring that the final product is closely aligned with the user's needs and expectations. Agile also emphasizes collaboration between cross-functional teams and stakeholders, enhancing transparency and communication throughout the development process.

The preference for Agile over the iterative model has been highlighted due to Agile's adaptability, iterative cycles, and focus on customer collaboration. Agile allows for flexibility and continuous improvement through sprints, each of which produces a potential shippable product increment. This approach is particularly effective given the dynamic nature of EMS development, where regulations and operational requirements evolve frequently. The Agile structure facilitates continuous feedback from stakeholders, ensuring that the system closely aligns with user needs and expectations throughout the development process.

In contrast, while the iterative model also involves iterative cycles of development and improvement, it usually follows a more structured and sequential approach. Agile flexibility is best suited for projects where requirements may evolve or where early and ongoing user feedback is necessary. This makes Agile particularly effective for the dynamic nature of EMS development, where regulations and operational requirements can change frequently, necessitating a development approach that can adapt quickly and efficiently.

**Second:**

**Feedback: Why did you create a page for both sign-in and sign-up, and does it make sense to have one page for both?**

Creating a sign-in and sign-up page can improve the user experience by providing clear, focused interfaces for each action. The login page is designed for existing users, providing a direct way to access their accounts, while registering for new users, guiding them through the registration process. This separation can reduce confusion and simplify the user journey, ensuring that the registration page serves a specific purpose without overwhelming the user with too many options.

However, combining login and registration functions on one page can also be beneficial, especially in terms of simplicity and ease of navigation. One page can include toggles or tabs to switch between login and signup forms, making it easier for users to choose the action they need. The decision to use separate pages or a combined page should be based on user testing and conclusion

**Additional areas for exploration, development and improvement in the web application:**

To address this feedback, it may be useful to conduct A/B testing to determine which approach users prefer. This includes prototyping separate and integrated login/registration pages and collecting user feedback through usability testing sessions. Analyzing metrics such as task completion time, error rates, and user satisfaction can provide data-driven insights into what design enhances the user experience. Additionally, exploring modern authentication methods, such as single sign-on (SSO) and social media logins, can further streamline registration and login processes, making the system more accessible and user-friendly.

**Third:**

**Feedback: why aren't all three buttons on one card page instead of separate pages for adding, and one page for update and delete?**

It was proposed to consolidate all card-related actions (add, update, delete) on a single page to simplify user interactions and enhance efficiency. This approach reduces the need to navigate between multiple pages, making it easier for users to manage cards. However, separating these actions can reduce the risk of accidental actions, such as deleting the card when you want to update it, by providing clearer interfaces dedicated to each function. Therefore, in the application, I separated the buttons so that the addition buttons are on one page and the update and deletion buttons are on another page, but in on the same page for adding events, there is a button titled All Events. Clicking on it will take you quickly and smoothly to the page for updating and deleting events.

**Additional areas to explore, develop and improve the web application:**

Further exploration could include designing and testing a unified card management interface with clearly marked sections or forms for adding, updating, and deleting cards. User feedback and usability testing can help determine how effective this design is in reducing errors and enhancing user satisfaction. In addition, implementing confirmation dialogs and undo options can mitigate the risk of accidental deletions, ensuring data integrity and user confidence in using the system. Improving the interface with tooltips and contextual help can also improve usability, guiding users through each action with ease.

**Fourthly:**

**Feedback: The overall design is very nice and easy to use.**

Positive comments on the design emphasize the importance of a visually attractive and intuitive user interface (UI). The beautiful and user-friendly design not only enhances ease of use but also contributes to increased user satisfaction and engagement. Strengths of the current design include intuitive navigation, clean layouts, and responsive elements that make interactions smooth and fun. Maintaining consistency in design, using familiar icons, and providing a clutter-free interface are key aspects of its success.

Additional areas for exploration and development:

Building on a strong design foundation, further exploration can include incorporating advanced UI/UX elements such as mobile adaptive design, ensuring a seamless experience across all platforms. Continuously collect user feedback rough surveys and usability tests can identify areas for improvement and new features that users might find useful. Additionally, exploring accessibility improvements, such as keyboard navigation, screen reader compatibility, and high contrast modes, can make the EMS system more comprehensive, catering to a wider range of users and ensuring compliance with accessibility standards. Regularly updating the design to match current UI/UX trends can keep the system modern and attractive.

**Additional Areas for Exploration and Development in the Website Application:**

**Enhanced Security and Privacy Features**

Security and privacy are paramount in any web application, particularly for systems dealing with sensitive environmental data. Enhancing security measures, such as implementing multi-factor authentication (MFA), can add an additional layer of protection against unauthorized access. Encrypting data both in transit and at rest ensures that sensitive information remains secure from potential breaches. Regular security audits and vulnerability assessments can help identify and mitigate potential threats, keeping the system robust and secure.

Additionally, incorporating comprehensive privacy controls can help users manage their data more effectively. Providing users with detailed privacy settings, such as the ability to control data sharing and manage consent, can enhance trust and compliance with data protection regulations like GDPR (General Data Protection Regulation). Transparent privacy policies and regular updates on how user data is handled can further bolster user confidence in the EMS application.

**Incorporating Machine Learning and AI**

Incorporating machine learning (ML) and artificial intelligence (AI) capabilities can significantly enhance the functionality and intelligence of the EMS application. ML algorithms can analyze large datasets to identify patterns and trends that might not be immediately apparent. For instance, predictive analytics can forecast future environmental conditions based on historical data, allowing users to take preemptive actions. AI-powered chatbots can provide instant assistance to users, answering queries and guiding them through complex tasks.

Furthermore, AI can be used to automate routine tasks, such as data entry and validation, freeing up users to focus on more strategic activities. Implementing anomaly detection algorithms can help identify unusual patterns in environmental data, alerting users to potential issues that require immediate attention. By leveraging AI and ML, the EMS application can become more proactive, intelligent, and responsive, ultimately providing greater value to its users.

4. I will promote and demonstrate the correct and effective way in which the tools as well as the methodology used support the work of developing and building an EMS (XYS Events) web application and its proper implementation.

**Windows operating system:**

The use of the Windows Operating System (OS) provides a solid foundation for EMS system development, ensuring a stable, secure and user-friendly environment. The familiar interface and comprehensive support resources available for Windows reduced the learning curve, allowing the development team to focus on building the system efficiently. I was able to build the project in a good and easy environment and obtain an operating system that is easy to use and very suitable for the development process. Therefore, by using the Windows operating system, the project was successfully completed and executed in a brilliant manner. The operating system's compatibility with a wide range of third-party software and hardware ensures seamless integration of core development tools and components, facilitating smooth progress throughout the project.

Furthermore, Windows' strong security features played a crucial role in protecting the EMS system during development. Regular updates and patches from Microsoft protected the system from potential vulnerabilities, ensuring that development continued uninterrupted. This reliability and ease of use supported the Agile methodology's focus on continuous improvement and adaptability, enabling the team to respond quickly to changes and feedback throughout the development process.

**Visual studio code**

Visual Studio Code (VS Code) was instrumental in the development of EMS due to its lightweight nature and extensive customization options. The IDE provides a streamlined environment that improves programming efficiency, allowing developers to write, test, and debug code effectively. I was able to create and develop the EMS web application and transform the system into the current state it is in now. With the numerous tools and plugins present in VS Code, it has made the process of writing code for the system quite easier as well as organizing the code in an efficient manner as well.

The availability of numerous extensions and plug-ins specifically designed to meet different programming needs has enabled the team to customize their development environment, optimizing it for the specific requirements of the EMS project.

VS Code's intuitive interface and powerful features make it easy to organize and manage code, ensuring that the development process is efficient and productive. The ability to easily navigate and edit files, as well as support for multiple programming languages, allowed the team to handle different aspects of the project seamlessly. This versatility and ease of use aligns perfectly with the Agile methodology, promoting rapid iteration and continuous integration, which is essential for adapting to evolving project requirements and stakeholder feedback.

**Visual Paradigm:** The visual paradigm has proven to be an invaluable tool for creating the detailed and accurate diagrams necessary for EMS system design and engineering. The software's comprehensive features enabled the development team to create system sequence diagrams and entity relationship diagrams, which were essential for visualizing the system's structure and interactions. These diagrams provided a clear blueprint for development, ensuring that all requirements were accurately implemented.

The ease of use and professional quality of Visual Paradigm diagrams facilitate effective communication and collaboration between team members and stakeholders. Also, using Visual Paradigm's diagramming tool, I was able to create and draw many important diagrams for the EMS web application such as the system sequence diagram as well as the entity relationship diagram as well. The visual model was very useful for me as a developer, as it had a very nice environment that allowed me to create all the different diagrams that later helped and assisted in the EMS web application development process. This visual representation of the system design allowed for better understanding and alignment, which is crucial in an Agile environment where feedback and constant iteration are key. The diagrams also helped identify potential issues early in the development process, allowing for timely adjustments and improvements.

**Draw.io**

Draw.io was another essential tool that helped me to create flowcharts and other diagrams that illustrate the operational flow of an EMS system. Its user-friendly interface and wide range of features made it easy to create clear, informative diagrams depicting how various processes and procedures are performed within the system. These visual aids were essential to understanding and improving the system's workflow.

Draw. Io’s cloud integration capabilities allowed me real-time flexibility, ensuring that diagrams were always up to date and accessible to everyone involved in the project anytime, anywhere. This collaborative approach supported Agile methodology's focus on teamwork and continuous improvement, enabling me to iteratively improve the system design based on ongoing feedback and insights.

**Agile methodology**

Adopting the Agile methodology for EMS development enables a flexible and iterative approach to project management, which is critical to address the dynamic and evolving nature of project requirements. Agile's focus on iterative development cycles allowed me to break the project down into manageable sprints, delivering improvements and additional features with each iteration. This approach allowed for continuous feedback from stakeholders. Another way the Agile model helped me greatly is that it allowed me to document important parts of data related to the project and organize my project in a good way due to its nature. This ensures that the system remains compatible with users' needs and expectations. Additionally, since I was working on the project through the Agile model, I was able to get a lot of feedback from the client and understand the project more as well as identify different areas that needed to remain within the project.

Agile's focus on collaboration and adaptability also played a significant role in mitigating risks and addressing issues promptly. The iterative nature of Agile allowed the team to identify and resolve problems early in the development process, preventing them from escalating and affecting the overall project timeline. The regular feedback loops and reviews facilitated by Agile ensured that the project stayed on track and that any necessary adjustments could be made swiftly, leading to a more robust and user-centric EMS system.

5. While XYZ Events' EMS system effectively addresses and operates the core challenges of event planning and management, several improvement and expansion opportunities can further elevate the system's capabilities and value propositions:

Online Payment Gateway: Integrating the online payment gateway within the EMS system will provide a convenient and streamlined booking experience for event attendees. By allowing users to pay for event registration and related services online via credit card, the system eliminates the need for in-person visits to process payments, thus saving time and reducing friction in the booking process. Moreover, the online payment gateway enhances security and transparency by providing instant payment confirmation and generating digital receipts, ensuring a smooth and reliable transaction experience for users. The implementation of this feature is in line with the goal of enhancing customer satisfaction and ease of use, as it provides greater flexibility and convenience in managing event registrations and payments.

Multilingual Support: Offering multilingual support within your EMS system can greatly expand its accessibility and attract a diverse user base. By enabling users to switch between different languages, such as Spanish, French or Mandarin, the system accommodates non-English speakers and enhances their understanding and engagement with the platform. This feature not only enhances inclusivity, but also opens up new market opportunities by catering to international audiences and expanding the system's reach beyond English-speaking regions. Additionally, by making the system more user-friendly and accessible to a wider audience, multilingual support contributes to increased adoption rates and increased customer satisfaction, ultimately driving growth and competitiveness for XYZ Events.

Event Countdown Widget: One innovative addition to an EMS system can be the integration of an event countdown widget. This widget will display a dynamic countdown timer on your event web page, displaying the time remaining until the start of the event. The countdown timer can be customized to reflect different time zones and display options, allowing attendees to easily track time regardless of their location or device. Additionally, the tool can include interactive features such as social sharing buttons, allowing attendees to share the countdown with their network, thus generating excitement and anticipation for the event.

An event countdown widget offers many benefits to both event organizers and attendees. For organizers, it serves as a promotional tool to build anticipation and increase attendance by creating a sense of urgency around the event. By displaying the countdown prominently on the event webpage, organizers can effectively communicate key event details and deadlines, such as registration deadline or session start times. Furthermore, the widget provides a visual focal point that draws attention to the event, thus increasing visibility and engagement among potential attendees.

Interactive Event Map: Introducing an interactive event map feature within the EMS system can offer attendees a novel way to navigate event venues and access important information in real time. This feature would provide users with a digital map interface that displays the layout of the event venue, including various booths, stages, restrooms, and amenities. Attendees can interact with the map to explore different areas of the venue, view detailed descriptions of each location, and access relevant information such as session schedules, speaker profiles, and exhibitor details.

The interactive event map can enhance the attendee experience by providing visual guidance and making it easier for users to find their way around the event venue. Additionally, the map can incorporate dynamic features such as real-time updates on session locations, live tracking of shuttle buses or parking availability, and notifications for important announcements or changes in schedule. By leveraging geolocation technology, users can also receive personalized recommendations based on their current location within the venue, such as nearby food vendors or networking opportunities.

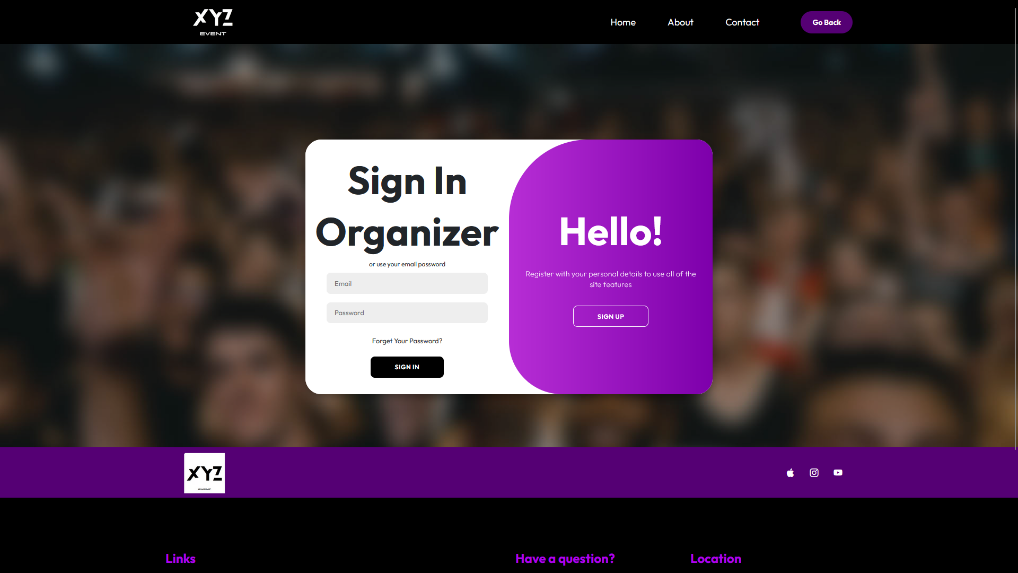
This feature offers a unique value proposition for XYZ Events by combining the convenience of digital navigation with the engagement of interactive content. It empowers attendees to make the most of their event experience by facilitating seamless navigation, personalized recommendations, and access to relevant information at their fingertips. Moreover, the interactive event map aligns with the broader goal of enhancing event efficiency and attendee satisfaction, making it a valuable addition to the EMS system.

**Activity 4:**

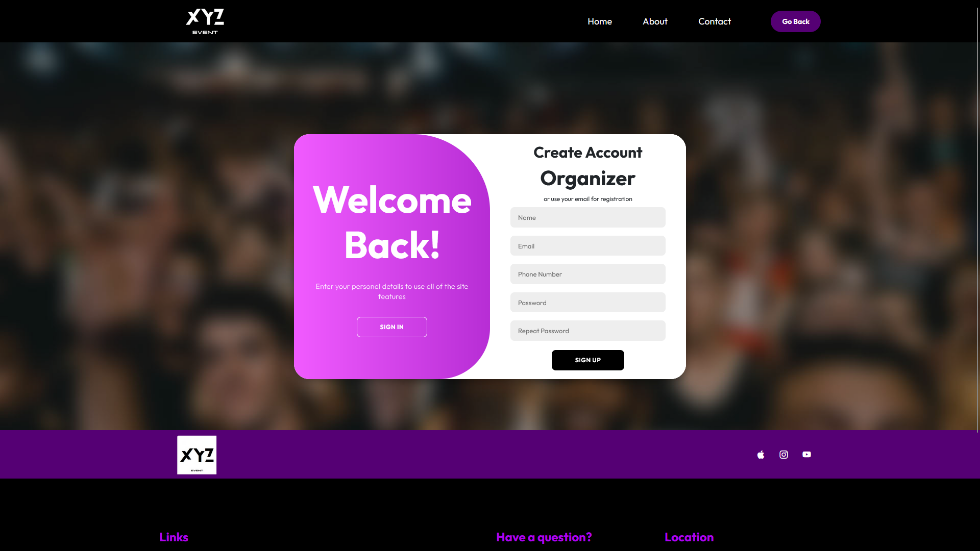
1. **Solution:**

**Organizer Requirements**

**1. Account Creation and Management:**



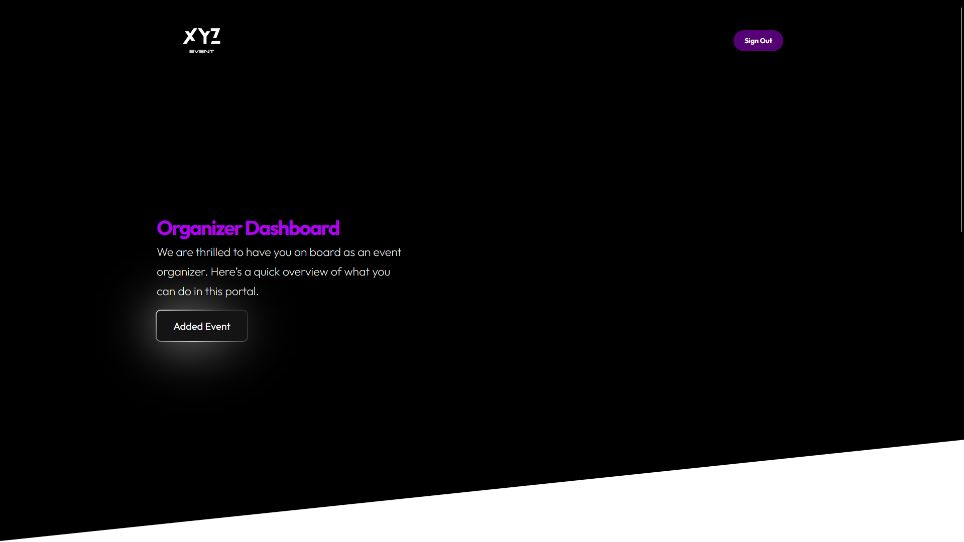
The organizer's account creation and management process are implemented in the loginOrganizer.php script. When an organizer fills out the registration form with their personal details (name, email, phone number, password), the system first checks if an account with the same credentials already exists in the database. This is done using a prepared statement to query the database for existing records. If no matching account is found, the system inserts the new organizer's details into the organizer table. This process ensures that each organizer has a unique account, preventing duplicates and ensuring data integrity. The session management then stores the organizer's information, enabling secure and personalized access to their account.

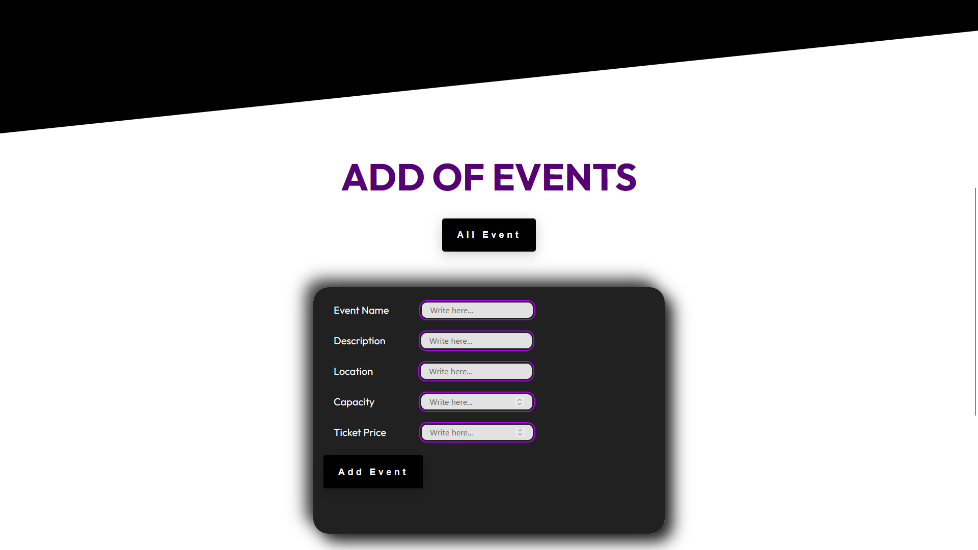


The code uses SQL prepared statements for both checking existing accounts and inserting new records to prevent SQL injection attacks, ensuring security. Once the account is created, the organizer is redirected to the login page, where they can log in using their email and password. Upon successful login, a session is initiated, and the organizer's ID and name are stored in the session variables, allowing the system to track the logged-in user and personalize their experience throughout their session.

The organizer's account data is stored in the organizer table in the database. When an organizer registers, a prepared statement with an SQL INSERT command is used to add the organizer’s details to the database. This ensures that the data is securely and efficiently stored. The code uses PHP's mysqli functions to connect to the database and execute these queries, handling potential errors and ensuring data integrity.

**2. Event Creation and Management:**





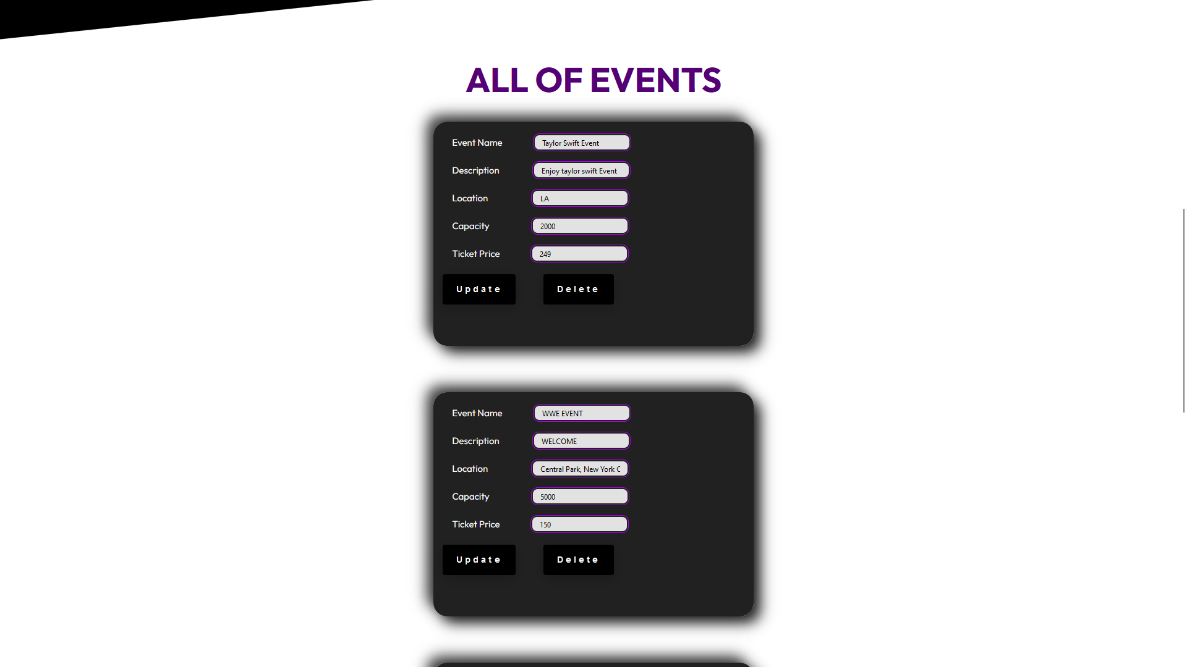
The event creation and management functionalities are handled in the Organizer.php script. When an organizer logs in, they can access a dashboard that provides options to add, view, edit, and delete events. To add a new event, the organizer fills out a form with event details such as the name, description, type, location, capacity, and ticket price. When the form is submitted, the data is processed by the script, and a prepared statement is used to insert the new event into the event table in the database. The script ensures that all required fields are filled out and validates the data before inserting it into the database.

The organizer can also view a list of their events using the DisplayEventOrganizer.php script, which retrieves event data from the database and displays it in a tabular format. They can edit or delete events as needed. The edit functionality retrieves the current event details, allowing the organizer to update the information and save the changes back to the database using an update query. The delete functionality removes the event from the database. These operations ensure that the organizer can efficiently manage their events, keeping the information up-to-date and accurate.

In the Organizer.php script, organizers can add new events by entering event details such as name, description, type, location, capacity, and ticket price. The form submits these details to the same page, where the code processes the input. Using a session variable that stores the organizer's ID, the system ensures that the event is associated with the correct organizer. The code then uses a prepared statement to insert the event details into the database.

The event data is stored in the event table. When the form is submitted, the INSERT SQL statement is executed using a prepared statement to securely insert the event details into the database. The organizer's ID, stored in the session, is also inserted as a foreign key to link the event to the organizer. This relational setup ensures that each event can be accurately attributed to its organizer, facilitating easy retrieval and management.

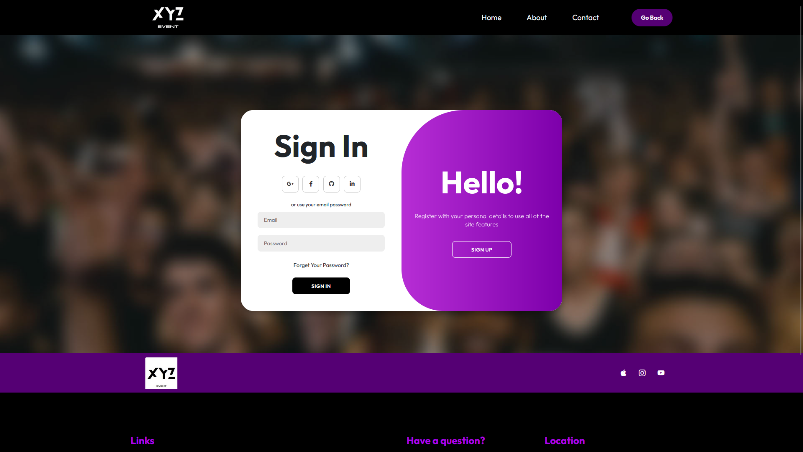
3. Event Viewing and Editing:

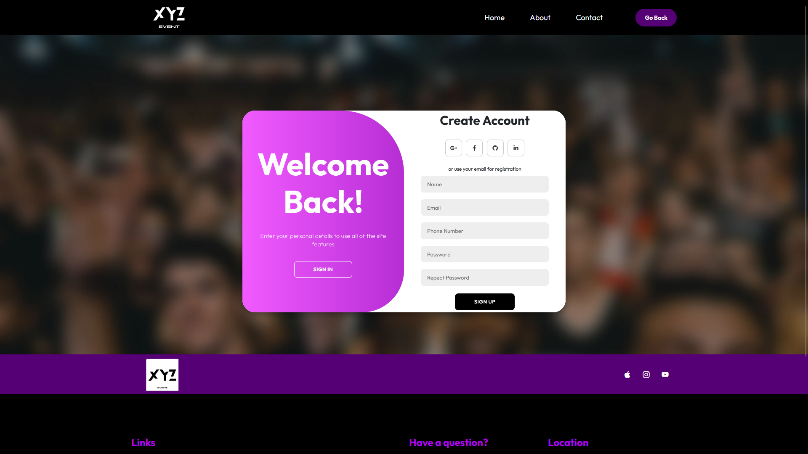


The DisplayEventOrganizer.php script allows organizers to view, edit, and delete their events. The script fetches the list of events associated with the logged-in organizer by using the organizer's ID stored in the session. It displays the events in a table format, where each event entry has options for editing or deleting. Editing an event involves pre-filling a form with the existing data, which can then be updated and submitted.

Fetching events involves executing a SELECT SQL query using the organizer's ID to retrieve relevant events from the event table. For updating events, the code uses an UPDATE SQL statement, and for deleting events, it uses a DELETE SQL statement. These operations ensure that the event data remains up-to-date and that the database accurately reflects the organizer’s event portfolio.

**Attendee Requirements**



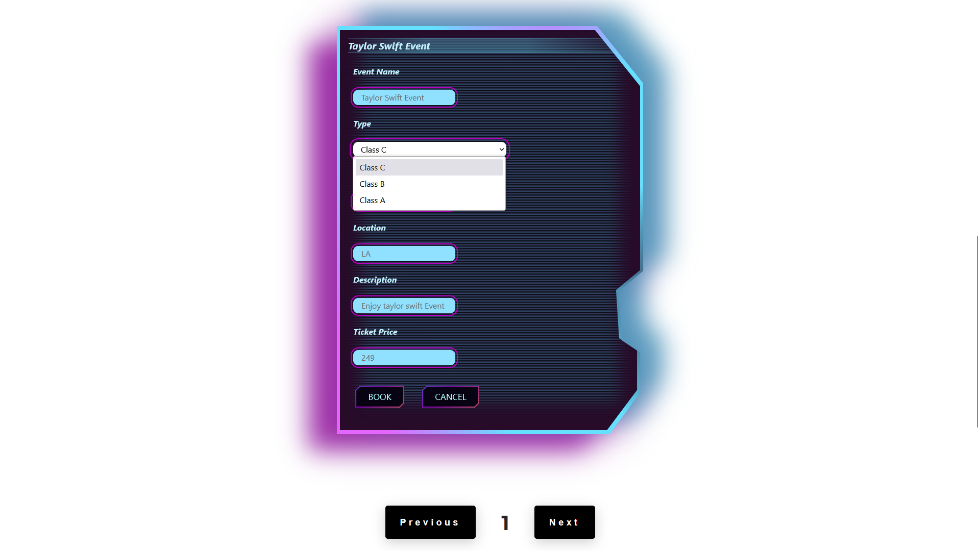


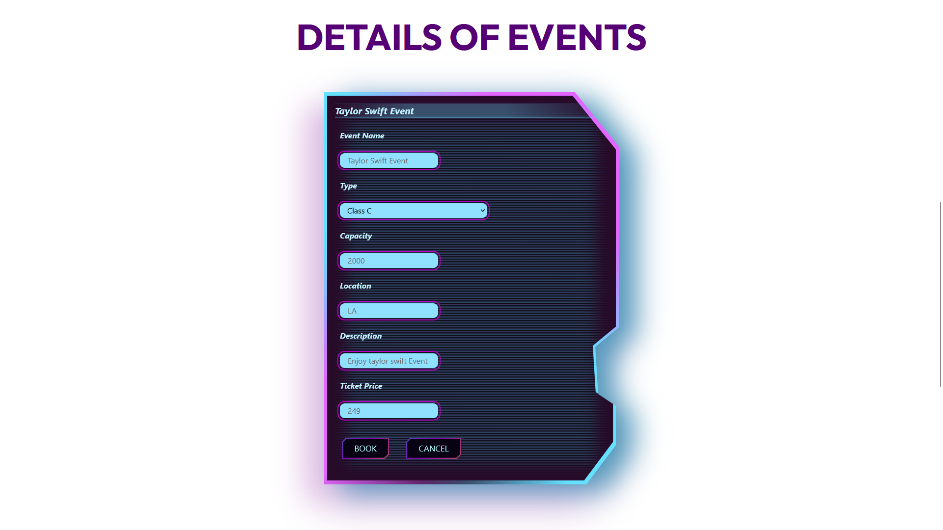
Attendee registration is handled in the attendee-register.php script. When an attendee fills out the registration form with their name, email, phone number, and password, the system first checks if an account with the same credentials already exists. This is done using a prepared statement to query the database for existing records in the attendee table. If no existing account is found, the new attendee's details are inserted into the database, ensuring that each attendee has a unique account.

The attendee's registration data is stored in the attendee table. The code uses a prepared statement with an INSERT SQL command to securely add the new attendee’s details to the database. The use of prepared statements ensures that the data is inserted safely, preventing SQL injection attacks and maintaining data integrity.

The registration process uses SQL prepared statements to prevent SQL injection attacks and ensure secure handling of attendee data. Once the registration is successful, the attendee is redirected to the login page, where they can log in using their email and password. Upon successful login, a session is initiated, and the attendee's ID and name are stored in the session variables, enabling personalized access to their account and events they may register for.

**Event Selection and Ticketing:**





The event selection and ticketing process for attendees are implemented in the bookTheEventPage.php script. Once logged in, attendees can browse a list of available events retrieved from the event table in the database. The script displays these events in a user-friendly format, allowing attendees to view details such as the event name, date, location, ticket type and ticket price.

Event data is fetched using a SELECT SQL query, which retrieves all available events from the event table. When an attendee books a ticket, their booking is recorded in the attendee\_booking table, which manages the many-to-many relationship between attendees and events. An INSERT SQL statement is used to add the booking details, ensuring that each booking is uniquely identified and correctly linked to the respective attendee and event.

When an attendee decides to book a ticket for an event, they select the event and submit the booking form. The system checks for duplicate bookings to ensure that the same attendee does not book the same event multiple times. This is done by querying the Attendee\_Booking table, which maintains a record of all bookings, linking attendees to events. If no duplicate is found, the booking is recorded in the Attendee\_Booking table by inserting the attendee's ID and the event ID as foreign keys. This ensures that the booking data is accurately stored and can be retrieved for future reference. The system then confirms the booking to the attendee, completing the ticketing process.

**2. Solution:**

I will analyze the variables that can influence the project's performance and assess their impact on the different stages of implementation, including design, development, and testing phases for the "EMS System" application. Additionally, I will explore how the risks identified in task number 1.2 also affected this implementation. I will enumerate the factors and provide a comprehensive analysis of each topic. The factors encompass:

* Resources

Impact on Design: The availability of resources is crucial during the design phase of the EMS system for XYZ Events. If the necessary tools and resources are not available, it can hinder the creation of detailed and accurate system designs. Tools for creating wireframes, flowcharts, and database schematics are essential for visualizing the system architecture. Without these tools, developers may struggle to produce comprehensive designs, leading to potential misunderstandings about system requirements and functionalities. The absence of proper design tools can result in incomplete or inaccurate system blueprints, which could lead to significant issues in later stages of development.

Impact on Implementation: During the implementation phase, the lack of resources can severely affect the development process. Essential resources include programming environments, libraries, and frameworks necessary for coding the EMS system. If developers lack access to these resources, it can delay the development process and increase the risk of bugs and errors. Additionally, the EMS system relies heavily on database management; without the proper tools and resources to set up and manage databases, critical functionalities may fail. This can lead to incomplete features, compromised system performance, and potential project failure.

Impact on Testing: In the testing phase, the availability of testing tools and resources is paramount. These tools help identify and fix bugs and ensure that the system functions as intended. If the EMS lacks adequate testing resources, developers may miss critical issues, leading to system vulnerabilities and errors post-deployment. Comprehensive testing tools are necessary to simulate various user scenarios and stress-test the system to ensure it can handle real-world usage. Without these resources, the EMS system may be released with significant flaws, affecting user satisfaction and system reliability.

* Scope

Impact on Design: The project scope directly influences the design phase of the EMS system. An unclear or poorly defined scope can lead to multiple design revisions and updates, which can cause inconsistencies and misalignments in the system architecture. As the scope expands, the complexity of the designs increases, requiring more time and effort to ensure accuracy and coherence. This can distract developers from focusing on core requirements and lead to delays and design flaws. Clear and well-defined scope helps create stable and comprehensive designs that align with the project's goals.

Impact on Implementation: During implementation, scope creep can cause significant disruptions. As new requirements are added, developers must continuously update and modify the code, which can lead to delays and increased complexity. This constant change can introduce bugs and reduce system stability. Additionally, expanding scope can strain resources, as developers must allocate more time and effort to accommodate new features. Effective scope management is crucial to ensure that the project remains on track and that the EMS system meets its deadlines and quality standards.

Impact on Testing: In the testing phase, an expanding project scope can extend the testing duration and complicate the process. New features and functionalities require additional test cases and scenarios, increasing the testing workload. This can lead to rushed or incomplete testing if the project deadline is not adjusted accordingly. Moreover, frequent changes to the scope can result in overlooked bugs and inconsistencies, as testers may not have sufficient time to thoroughly test all aspects of the system. Proper scope management ensures that the testing phase is comprehensive and that the system is robust and reliable.

* Experience

Impact on Design: The experience level of the development team significantly affects the design phase. Experienced developers can create more efficient and user-friendly designs, understanding the nuances of system architecture and user requirements. They can foresee potential issues and design the system to mitigate these risks. In contrast, less experienced developers may produce designs that are not optimized for performance or usability, leading to complications in later stages. Ensuring that the design team has the necessary experience and skills is crucial for the project's success.

Impact on Implementation: During implementation, the experience of the developers plays a crucial role in the quality and efficiency of the code. Experienced developers can write optimized and maintainable code, reducing the likelihood of bugs and performance issues. They are also better equipped to handle unexpected challenges and integrate new features smoothly. Inexperienced developers may struggle with complex coding tasks, leading to suboptimal code and increased debugging efforts. Investing in a skilled development team can enhance the implementation phase and ensure a higher quality EMS system.

Impact on Testing: In the testing phase, experienced testers can identify and resolve issues more effectively. They are familiar with best practices and can design comprehensive test cases that cover various scenarios. Their expertise allows them to detect subtle bugs and performance bottlenecks that less experienced testers might miss. Additionally, experienced testers can provide valuable feedback for improving the system. Ensuring that the testing team has the necessary skills and experience is vital for delivering a robust and reliable EMS system.

#### **Communication**

Impact on Design: Effective communication is essential during the design phase to ensure that all stakeholders have a clear understanding of the system requirements and functionalities. Poor communication can lead to misunderstandings and misaligned expectations, resulting in designs that do not meet the client's needs. Regular meetings and clear documentation help ensure that everyone is on the same page. Effective communication facilitates collaboration and helps create a design that aligns with the project's goals and user requirements.

Impact on Implementation: During implementation, communication issues can lead to delays and errors. If developers do not have a clear understanding of the requirements or changes, they may implement features incorrectly or overlook critical functionalities. Regular updates and feedback sessions help ensure that the development process stays aligned with the project's goals. Effective communication between team members and stakeholders helps identify and address issues promptly, ensuring a smooth implementation phase.

Impact on Testing: In the testing phase, communication is crucial for coordinating test efforts and ensuring that all aspects of the system are thoroughly tested. Testers need clear and detailed information about the system's functionalities and expected behaviors to design effective test cases. Communication issues can result in incomplete or redundant testing, leading to undetected bugs and performance issues. Regular communication between testers, developers, and stakeholders helps ensure that the testing process is comprehensive and that the system meets the required standards.

* Technology Risk

Impact on Design: Technology risks, such as security breaches and lack of resources, can significantly affect the design phase. Security considerations must be integrated into the system architecture to protect sensitive data and ensure compliance with regulations. If the design does not account for potential security threats, the system may be vulnerable to attacks. Additionally, inadequate resources can hinder the creation of comprehensive and accurate designs, leading to potential issues in later stages. Mitigating technology risks involves thorough planning and incorporating security best practices into the design.

Impact on Implementation: During implementation, technology risks can cause significant disruptions. Security breaches can compromise sensitive data and damage the system's integrity. Developers must implement robust security measures to protect the system and user data. Additionally, resource constraints can delay the development process and affect the quality of the code. Ensuring that the development team has the necessary resources and tools helps mitigate these risks and ensures a smooth implementation phase.

Impact on Testing: In the testing phase, technology risks such as security vulnerabilities and inadequate testing resources can affect the system's reliability and performance. Comprehensive testing tools and methods are necessary to identify and address potential security issues and performance bottlenecks. If these resources are lacking, critical bugs and vulnerabilities may go undetected, compromising the system's integrity and user experience. Mitigating technology risks involves thorough testing and ensuring that the testing team has access to the necessary tools and resources.

* Communication Risk

Impact on Design: Communication risks can lead to misunderstandings and misaligned expectations during the design phase. If the development team does not have a clear understanding of the client's requirements, the system design may not meet the user's needs. Regular communication and feedback sessions help ensure that the design aligns with the project's goals and user expectations. Mitigating communication risks involves clear and concise documentation and effective collaboration between all stakeholders.

Impact on Implementation: During implementation, communication risks can cause delays and errors. Misunderstandings about requirements or changes can lead to incorrect or incomplete features. Regular updates and feedback sessions help ensure that the development process stays aligned with the project's goals. Effective communication between team members and stakeholders helps identify and address issues promptly, ensuring a smooth implementation phase.

Impact on Testing: In the testing phase, communication risks can result in incomplete or redundant testing. Testers need clear and detailed information about the system's functionalities and expected behaviors to design effective test cases. Communication issues can lead to undetected bugs and performance issues. Regular communication between testers, developers, and stakeholders helps ensure that the testing process is comprehensive and that the system meets the required standards.

* Scope Creep Risk

Impact on Design: Scope creep can cause significant disruptions during the design phase. As new requirements are added, the design must be continuously updated and modified, which can lead to inconsistencies and misalignments. This can also distract developers from focusing on core requirements and lead to delays and design flaws. Clear and well-defined scope helps create stable and comprehensive designs that align with the project's goals.

Impact on Implementation: During implementation, scope creep can cause significant disruptions. As new requirements are added, developers must continuously update and modify the code, which can lead to delays and increased complexity. This constant change can introduce bugs and reduce system stability. Effective scope management is crucial to ensure that the project remains on track and that the EMS system meets its deadlines and quality standards.

Impact on Testing: In the testing phase, an expanding project scope can extend the testing duration and complicate the process. New features and functionalities require additional test cases and scenarios, increasing the testing workload. This can lead to rushed or incomplete testing if the project deadline is not adjusted accordingly. Proper scope management ensures that the testing phase is comprehensive and that the system is robust and reliable.

* Cost Risk

Impact on Design: Cost risks can affect the design phase by limiting the resources available for creating comprehensive and accurate system designs. Budget constraints may restrict access to necessary tools and technologies, leading to suboptimal designs. Proper budgeting and financial planning help ensure that the design phase has the resources it needs to produce high-quality system blueprints.

Impact on Implementation: During implementation, cost risks can cause significant disruptions. Budget constraints may limit access to essential development tools and technologies, affecting the quality and efficiency of the code. Developers may have to make compromises to stay within budget, leading to potential system vulnerabilities and performance issues. Effective financial planning and resource management help mitigate cost risks and ensure a smooth implementation phase.

Impact on Testing: In the testing phase, cost risks can limit access to necessary testing tools and resources, affecting the system's reliability and performance. Comprehensive testing tools and methods are necessary to identify and address potential bugs and vulnerabilities. Budget constraints may lead to incomplete or rushed testing, compromising the system's integrity and user experience. Proper financial planning and resource allocation help ensure that the testing phase is thorough and effective.

* Operational Risk

Impact on Design: Operational risks, such as disruptions in essential processes and operations, can affect the design phase by delaying or hindering the creation of comprehensive and accurate system designs. Ensuring that the design team has the necessary resources and support helps mitigate operational risks and ensures a smooth design phase.

Impact on Implementation: During implementation, operational risks can cause significant disruptions. Issues such as human errors, natural disasters, and cyberattacks can affect the development process and compromise the system's integrity. Effective risk management and contingency planning help mitigate operational risks and ensure a smooth implementation phase.

Impact on Testing: In the testing phase, operational risks can affect the system's reliability and performance. Issues such as human errors, natural disasters, and cyberattacks can compromise the testing process and affect the system's integrity. Effective risk management and contingency planning help mitigate operational risks and ensure a thorough and effective testing phase.

* Skills Resource Risk

Impact on Design: Skills resource risk can affect the design phase if the development team lacks the necessary expertise and experience to create comprehensive and accurate system designs. Ensuring that the design team has the necessary skills and support helps mitigate skills resource risk and ensures a smooth design phase.

Impact on Implementation: During implementation, skills resource risk can cause significant disruptions. If the development team lacks the necessary expertise and experience, it can lead to suboptimal code and increased debugging efforts. Ensuring that the development team has the necessary skills and support helps mitigate skills resource risk and ensures a smooth implementation phase.

Impact on Testing: In the testing phase, skills resource risk can affect the system's reliability and performance. If the testing team lacks the necessary expertise and experience, it can lead to incomplete or ineffective testing. Ensuring that the testing team has the necessary skills and support helps mitigate skills resource risk and ensures a thorough and effective testing phase.

* Market Risk

Impact on Design: Market risks can affect the design phase by introducing uncertainty and potential changes in user requirements and expectations. Ensuring that the design team stays updated on market trends and user needs helps mitigate market risks and ensures that the system design aligns with market expectations.

Impact on Implementation: During implementation, market risks can cause significant disruptions. Changes in user requirements and expectations can lead to frequent updates and modifications in the system, affecting the development process and system stability. Ensuring that the development team stays updated on market trends and user needs helps mitigate market risks and ensures a smooth implementation phase.

Impact on Testing: In the testing phase, market risks can affect the system's reliability and performance. Changes in user requirements and expectations can lead to frequent updates and modifications in the system, affecting the testing process and system integrity. Ensuring that the testing team stays updated on market trends and user needs helps mitigate market risks and ensures a thorough and effective testing phase.

Examining the factors that influence EMS project performance and evaluating their impact on the design, development, and testing phases highlights the importance of resource availability, scope management, team expertise, and effective communication. Additionally, addressing identified risks, such as technology, communications, scope creep, cost, operational, skill resource, and market risks, ensures that the EMS project stays on track and meets quality standards and deadlines. Proper planning, risk management, and continuous improvement are essential for successful implementation of an EMS system for XYZ events.

**2. Solution:**

**Strengths of the EMS Project**

The Event Management System (EMS) I developed in Activity 3 has several notable strengths, which greatly enhance its usability and functionality for both organizers and attendees. First, EMS allows any user to create an account, ensuring that organizers and attendees have personal profiles within the system. This feature is crucial as it enables users to access and use the system according to their roles, whether managing events or booking tickets. The account creation process is streamlined and secure, ensuring that all user data is handled with care and protected from unauthorized access.

Another strength of EMS is its robust event booking system. Users can browse and book multiple events as they please, with the system keeping track of all bookings. This functionality is essential for attendees who may wish to participate in different events. The system’s ability to manage and record all bookings in the database ensures that event attendance is accurately tracked, which aids in resource planning and event management. The integration of authentication and authorization mechanisms further enhances the security of the system, ensuring that only authorized users can access their dedicated pages and perform actions relevant to their roles.

The event management system also provides a comprehensive event management dashboard for organizers. This dashboard includes advanced features that allow organizers to add, update, delete, and view event data, as well as manage announcements. By abstracting these operations away from regular attendees, the system ensures that only authorized personnel can make important changes to the event data, thus maintaining data integrity and preventing unauthorized modifications. This separation of roles and permissions is essential to keeping the system secure and reliable.

One of the key strengths of the event management system is its robust event management system. Any user, whether an organizer or an attendee, can create an account and have their own profile. The loginOrganizer.php and attendee-register.php scripts facilitate secure account creation by validating user input and preventing duplicate accounts. This ensures that each user has a unique profile, which they can use to interact with the system according to their role.

The system also effectively implements authentication and authorization. Organizers and attendees have different levels of access; organizers can manage events while attendees can book and view events. This is accomplished through session management and role-based access control in the code, ensuring that users only access the functionality relevant to their role. This structured approach enhances security and ease of use, making the system reliable for all users.

EMS excels in event management for organizers. The Organizer.php script allows organizers to add, view, edit, and delete events. The process of creating an event involves filling out a detailed form, and the data is linked to the database through crafted SQL statements. This not only reduces manual errors but also ensures that the data is stored securely and efficiently.

Furthermore, the ability to edit and delete events via DisplayEventOrganizer.php adds flexibility to organizers. They can update event details as needed or remove events that are no longer relevant. This dynamic management capability is crucial to maintaining accurate and up-to-date event information, and improving overall event coordination and resource allocation.

For attendees, EMS provides a seamless experience from registration to event booking. The bookTheEventPage.php script allows attendees to easily browse and book events. The system checks for duplicate bookings and ensures that each booking is accurately recorded in the database. This functionality is essential for managing attendee data and preventing overbooking or inconsistent data. The user-friendly interface and organized data management enhance the attendee experience. They can easily find events, book tickets, and view their booking history. This efficient process reduces confusion and increases user satisfaction, making the system more attractive to potential users.

**EMS Project Weaknesses**

Despite the strengths of the system I developed, the EMS system has some weaknesses that need to be addressed. One important issue is the vulnerability in the admin authentication system. If a random user or hacker discovers the admin login credentials, they can access the admin dashboard, which could compromise the entire system. To mitigate this risk, a proposal was made to add an additional layer of security during the admin registration process. This could include requiring a unique code that is only provided to legitimate admins by the system administrators. Implementing this feature would greatly enhance the security of the system by ensuring that only authorized individuals can create admin accounts.

**Evaluation of Enhancements for the EMS Project**

Solution: To mitigate these risks, the system should include an additional layer of security for organizer accounts. One effective solution is to introduce a “code” field during the registration process for organizers. This code will only be provided to organizers authorized by the EMS administrators. By requiring this code, the system ensures that only authorized individuals can create organizer accounts, which greatly enhances security.

Another weakness is the lack of functionality for users to edit their personal information. While the system allows users to view their booking history and personal details, it does not currently provide a way for users to update this information. This limitation can be inconvenient for users who need to update their contact details or other personal information. To address this issue, a feature could be added that allows users to submit requests to change their personal information. These requests would be reviewed and approved by an administrator, ensuring that data integrity is maintained while providing flexibility for users to update their information as needed.

**Evaluation of Enhancements for the EMS Project**

Solution: A viable improvement could be to add an editable user profile section where participants can update their personal information. Implementing this feature would require a secure process for handling data changes, such as sending requests to an administrator for approval before updating the database. This would ensure the accuracy of user data while maintaining the security of the system.

The current booking process requires participants to visit the organizer’s office to complete payment and collect their tickets. This step can be inconvenient and time-consuming for users. To improve this process, the system could integrate an online payment gateway. This feature would allow users to complete their bookings and payments online, providing a more seamless and efficient experience. By integrating a secure payment system, EMS can offer a more modern solution that is in line with current user expectations and technological advancements.

**Evaluation of Enhancements for the EMS Project**

Solution: Integrating a payment gateway will simplify the booking process. By allowing participants to pay online using credit cards or other digital payment methods, the system can automatically confirm bookings and issue e-tickets. This will not only enhance user convenience, but will also reduce administrative workload and improve the overall efficiency of EMS.

**References:**

* Altvater, A. (2024). What is SDLC? Understand the Software Development Life Cycle. [online] Stackify. Available at: https://stackify.com/what-is-sdlc/.
* ‌Michael Page. (2021). The top 7 SDLC methodologies. [online] Available at: https://www.michaelpage.com.au/advice/career-advice/productivity-and-performance/top-7-sdlc-methodologies.
* GeeksforGeeks (2019). Frontend vs Backend - GeeksforGeeks. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/frontend-vs-backend/.
* ‌Club, T.C. (n.d.). 20 Best IDE Software [2023 Guide]. [online] The CTO Club. Available at: https://thectoclub.com/tools/best-ide-software/.

**Plagiarism**

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

**Student Declaration**

|  |
| --- |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.  Student signature: Rashed Hasan Date: 11/6/2024 |