SOFTWARE ENGINEERING

GROUP ASSIGNMENT

CYNTHIA WANJIRU SCT221-C004-0530/2022

ABDULLAHI ABASS SCT221-C004-0516/2022

KHALIF NOOR SCT221-C004-0114/2023

QUESTION ONE

**Read carefully and understand the case study below to answer the two questions. Be as innovative as possible.**

 The railway reservation system functions as follows;

The passenger is required to fill in a reservation form giving detail of his journey. The counter clerk ensures whether the place is available. If so, entries are made in a register, tickets are prepared, amount is computed and cash is accepted.

A booking statement is prepared in triplicate format from the reservation register. One copy is retained as office copy, the other is pasted to the compartment and third is passed on to the train conductor. Besides booking statement, cash statement is prepared at the end of each shift.

Prepare System Require Specification and system specification for above system problem.

System Requirements Specification:

1. Introduction: The railway reservation system is designed to automate the process of ticket booking and cash management. It aims to provide a convenient and efficient way for passengers to reserve their seats and for the railway authorities to manage the reservations and cash transactions.

2. Functional Requirements:

a. User Registration: The system should allow users to register their details, including name, contact information, and any other necessary information.

b. Reservation Form: The system should provide a reservation form for passengers to fill in their journey details, including the source, destination, date, and class of travel.

c. Seat Availability Check: The system should check the availability of seats on the specified route and date.

d. Ticket Generation: Once the seat availability is confirmed, the system should generate tickets with a unique ticket number and passenger details.

e. Amount Calculation: The system should calculate the fare based on the class of travel and any applicable discounts.

f. Cash Acceptance: The system should accept cash payments from passengers for ticket booking.

g. Booking Statement: The system should generate a booking statement in triplicate format, with copies for office use, pasting to the compartment, and for the train conductor.

h. Cash Statement: The system should generate a cash statement at the end of each shift, summarizing the cash received.

3. Non-functional Requirements:

a. Security: The system should ensure the privacy and security of user and transaction data.

b. Reliability: The system should be highly reliable and available for users at all times.

c. Performance: The system should be able to handle a large number of concurrent users and process transactions quickly.

d. User-Friendly Interface: The system should have an intuitive and easy-to-use interface for both passengers and counter clerks.

e. Scalability: The system should be scalable to accommodate future growth and changes in the railway system.

System Specification:

1. System Architecture: The railway reservation system will be built as a client-server application. The server will handle the processing of reservation requests, seat availability checks, ticket generation, and cash management. The client application will provide a user interface for passengers to fill in reservation details and make payments. The system will use a database to store user information, reservation data, and cash transaction records.

2. Technology Stack: The system will be developed using a combination of programming languages and technologies. The server-side will be developed using a language like Java or C#, along with a framework like Spring or ASP.NET. The client-side will be developed using HTML, CSS, and JavaScript, with a responsive design for compatibility across different devices. The database will be implemented using a relational database management system like MySQL or PostgreSQL.

3. Testing and Deployment: The system should undergo rigorous testing to ensure its functionality, reliability, and performance. Different types of testing, such as unit testing, integration testing, and user acceptance testing, should be performed. Once the testing is complete, the system can be deployed on a reliable hosting platform or on-premises infrastructure.

Design a prototype for the case study

The prototype of the railway reservation system will be a computerized system that will allow passengers to fill in a reservation form with details of their journey. The system will then check whether the desired destination is available, and if so, make the appropriate entries in a register, prepare tickets, compute the fare, and accept payment. Finally, the system will generate a booking statement in triplicate form, with one copy retained as an office copy, one copy pasted to the compartment, and one copy given to the train conductor. In addition, the system will prepare a cash statement at the end of each shift.

QUESTION TWO

Discuss the following software testing techniques (walkthroughs, reviews and inspections, dynamic testing, Traceability matrices. Debugging environments)

* **Walkthroughs**: This is a technique where a group of people, usually the developers, testers, and users, review a software product or document and provide feedback. The purpose of walkthroughs is to identify errors, improve quality, and ensure that the product meets the requirements and expectations. Walkthroughs are informal and flexible, and they can be done at any stage of the software development life cycle.
* **Reviews and inspections**: This is a technique where a group of people, usually the peers of the author, examine a software product or document and check for defects, compliance, and improvement opportunities. The purpose of reviews and inspections is to ensure that the product conforms to the standards and specifications, and to prevent errors from propagating to later stages. Reviews and inspections are formal and structured, and they require planning, preparation, execution, and follow-up.
* **Dynamic testing**: This is a technique where the software product is executed with various inputs and outputs to verify its functionality, performance, reliability, security, and usability. The purpose of dynamic testing is to detect faults, measure quality attributes, and evaluate user satisfaction. Dynamic testing can be done manually or automatically, and it can be classified into different levels (unit, integration, system, acceptance) and types (functional, non-functional, regression) depending on the scope and objective.
* **Traceability matrices**: This is a technique where a table or matrix is created to show the relationships between different elements of the software product or process. For example, a traceability matrix can link the requirements to the design, code, test cases, and defects. The purpose of traceability matrices is to ensure that the product meets all the requirements, to facilitate change management and impact analysis, and to provide evidence of quality assurance.
* **Debugging environments**: This is a technique where a set of tools and facilities are used to locate and fix errors in the software product. Debugging environments can include features such as breakpoints, watch variables, step execution, call stack, error messages, log files, etc. The purpose of debugging environments is to help the developers identify the root cause of the errors, correct them quickly and efficiently, and prevent them from recurring.

QUESTION THREE

Discuss Software Quality Evaluation **–** Problems, Software standards, Certification, Software Tools support for Systems Engineering

* **Problems**: Evaluating software quality can be challenging due to the subjective nature of quality attributes, the complexity of software products, the variability of user needs and environments, and the lack of universal standards and metrics. Problems can arise in defining what quality means, measuring how well the software meets the quality criteria, predicting how the software will perform in the future, and improving the software to achieve higher quality.
* **Software Standards**: Standards are established guidelines or specifications used to ensure that a product, process, or service meets certain criteria for performance, safety, and quality. Software standards can cover various aspects such as coding practices, documentation formats, testing methods, project management processes, and more. They help in maintaining consistency, improving interoperability, reducing errors, and enhancing quality.
* **Certification**: Certification is a formal process where a third party (usually a standard organization or a certification body) assesses the software product or process and confirms that it complies with certain standards or criteria. Software certification can provide assurance of quality, reliability, security, and other attributes. It can also increase customer confidence, market acceptance, and competitive advantage.
* **Software Tools Support for Systems Engineering**: There are various tools available that support different aspects of systems engineering including requirements management, design modeling, simulation and analysis, configuration management, project management, quality assurance, etc. These tools can automate tedious tasks, prevent errors, improve collaboration, enhance visibility and control, and ultimately increase productivity and quality.

QUESTION FOUR

DiscussCASE tools as used in Web engineering techniques and process, standards and guidelines.

* **Design and Modeling**: CASE tools can be used to create graphical models of the web application’s architecture, data structures, user interfaces, and other components. These models can help in understanding and communicating the design, identifying potential issues, and making informed decisions.
* **Coding and Testing**: CASE tools can provide features such as code generation, code analysis, unit testing, integration testing, performance testing, security testing, etc. These features can help in reducing coding errors, ensuring code quality, verifying functionality, and assessing non-functional attributes.
* **Project Management**: CASE tools can support project management activities such as planning, scheduling, tracking, reporting, risk management, etc. These activities can help in coordinating the team’s efforts, controlling the project’s progress, managing the project’s risks, and ensuring the project’s success.
* **Configuration Management**: CASE tools can manage different versions of the web application’s artifacts (code, documents, data, etc.), control changes to these artifacts, and maintain traceability between these artifacts. These capabilities can help in handling parallel development efforts, accommodating evolving requirements, and preserving historical information.

QUESTION FIVE

Discuss Process Improvement (PI):- Quality and process standards and guidelines

* **Quality Standards**: These are predefined specifications set by recognized standard bodies like ISO (International Organization for Standardization), IEEE (Institute of Electrical and Electronics Engineers), etc. They define the criteria that a product, service, or process must meet to be considered as ‘quality’. For example, ISO 9001 is a standard that sets out the criteria for a quality management system. Adhering to these standards during PI ensures that the improved processes result in high-quality outcomes.
* **Process Standards**: These are specific to the processes followed in an organization. They provide a defined framework within which tasks must be performed. For example, CMMI (Capability Maturity Model Integration) is a process level improvement training and appraisal program that organizations use to streamline process improvement and encourage productive, efficient behaviors. Adherence to process standards ensures consistency and repeatability in processes.
* **Guidelines**: These are recommendations that act as a reference point for the implementation of standards. They provide practical advice on how standards can be implemented and are often more flexible than standards. Guidelines can be organization-specific or industry-specific.