

 اونیورسیتی ملیسیا فہق السلطان عبداللہ UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH	FAKULTI KOMPUTERAN			MARKAH: / 50
	MATAPELAJARAN: SOFTWARE TESTING			
	TOPIK:		KOD: BCS3233	
	PENILAIAN: Assignment 1- Individual		BIL: 1 MASA: 2 Weeks (12 Hours SLT)	

NO MATRIK : CB21159

SECTION : - 01

Instructions:

1. This is an **Individual Assignment** that will carry **10% (50 Marks)** of the assessment marks. Details mark distribution as follows:

CO Number	Domain	Marks	Percentage
CO1- Knowledge (C4)	Cognitive	35	7%
CO3-Critical Thinking & Problem Solving (C3)	Cognitive	15	3%

2. This assignment will cover chapters from Week 1 to Week 5 of the lectures session.
3. The submission through KALAM is **MUST** and the Hardcopy of the submission depends on your lecturers' section. Please confirm it with your respective lecturer.
4. The **submission date** for this Individual Assignment is **20th November 2023; 5.00PM via KALAM**. Please submit it by your lecture section accordingly.

Case Study:

Read the given Case Study (**The Omninet Requirements Document**) as provided in **Appendix 1**. This case study must be referred to in answering all the questions for this assignment.

Questions:

1. Suggest the best testing model to be implemented in the given case study. Give TWO reasons to justify your choice to implement the suggested model for the Omninet Project.
(CO3-5 Marks)

Considering the requirements outlined in the Omninet Requirements Document, the best testing model to implement for the Omninet Project would be **the V-Model (Verification and Validation Model)**. Here are two reasons to justify this choice:

Traceability between Requirements and Testing:

The V-Model is known for its clear and direct mapping between each development phase and its corresponding testing phase. As the Omninet project involves specific and well-documented requirements, the V-Model ensures that each requirement is associated with a specific test case or test scenario.

Given the detailed nature of the Omninet project requirements, the V-Model allows for easy traceability. Each stage of development (such as design, coding, and testing) has a corresponding stage in testing, ensuring that the delivered product meets the specified requirements.

Early Detection of Defects:

The V-Model emphasizes testing activities in parallel with the development phases, starting with the verification phase. As the development progresses, testing activities are initiated, allowing for early detection of defects and issues.

In the context of the Omninet project, where there are multiple components such as kiosk software, payment processing, and call center functionalities, early defect identification is crucial. The V-Model's parallel testing approach ensures that issues are identified and addressed promptly, reducing the likelihood of costly corrections later in the development lifecycle.

2. If you were managing all the testing for Omninet Project, which levels of testing would you plan? Briefly explain your plan for how to be executed for each level of testing you propose and why it needs to be executed.

(CO3-10 Marks)

As the manager overseeing testing for the Omninet Project, I would plan multiple levels of testing to ensure the robustness and quality of the system. The proposed testing levels include:

Unit Testing:

Execution Plan: Developers will conduct unit testing for individual components, such as the kiosk software, payment processing module, and communication server.

Rationale: Unit testing ensures that each component functions correctly in isolation. It helps identify and fix defects at an early stage, reducing the likelihood of integration issues later in the development process.

Integration Testing:

Execution Plan: After successful unit testing, the integration testing phase will begin. This involves testing interactions between different components, such as the communication between the kiosk and the payment processing system.

Rationale: Integration testing validates that the various components work together as intended. It identifies interface issues, data flow problems, and communication errors, ensuring the seamless operation of the integrated system.

System Testing:

Execution Plan: Once integration testing is complete, system testing will be conducted to evaluate the entire Omninet system. This involves testing end-to-end scenarios, including user interactions with the kiosk, payment processing, and communication with the call center.

Rationale: System testing verifies that the entire system meets the specified requirements. It focuses on functional and non-functional aspects, including usability, performance, security, and compliance with the defined standards.

Acceptance Testing:

Execution Plan: Acceptance testing will be performed by stakeholders, including potential users and project sponsors, to validate that the Omninet system meets business requirements and expectations.

Rationale: Acceptance testing ensures that the system aligns with the intended business goals. It provides confidence to stakeholders that the product is ready for deployment and meets the criteria for successful implementation.

Regression Testing:

Execution Plan: Throughout the development lifecycle, a suite of regression tests will be maintained and executed after each code change or system update to ensure that existing functionalities remain unaffected.

Rationale: Regression testing prevents the introduction of new defects as the system evolves. It safeguards against unintended consequences of code changes and ensures that previously validated features continue to work as expected.

Security Testing:

Execution Plan: A dedicated security testing phase will be conducted to identify and address vulnerabilities in the Omninet system, including measures to protect user data, prevent unauthorized access, and secure payment transactions.

Rationale: Given the sensitive nature of user data and financial transactions, security testing is critical to safeguard against potential threats and vulnerabilities, ensuring the integrity and confidentiality of the Omninet system.

3. Choose any appropriate technique in Static Testing to be implemented in the Omninet Project. Give TWO reasons for your choosing technique.

(CO1 – 5 Marks)

An appropriate technique in static testing for the Omninet Project would be Static Code Analysis. Here are two reasons to support the choice of this technique:

Early Identification of Code Issues:

Reason: Static Code Analysis allows for a comprehensive review of the source code without executing it. By analyzing the code before runtime, potential issues such as syntax errors, code smells, and adherence to coding standards can be identified early in the development process.

Benefit: Early identification of code issues reduces the likelihood of these issues evolving into more complex problems during subsequent testing phases. It promotes a proactive approach to code quality, minimizing the time and effort required for debugging and corrective actions later in the development lifecycle.

Consistent Adherence to Coding Standards:

Reason: Static Code Analysis tools can enforce coding standards consistently across the entire codebase. This includes aspects such as naming conventions, indentation, and the proper use of programming constructs.

Benefit: Consistent adherence to coding standards enhances code readability, maintainability, and facilitates collaboration among developers. It ensures that the codebase follows a uniform structure, making it easier for team members to understand and contribute to the project. This is particularly important in a project like Omninet, which involves multiple components and contributors.

4. Plan how you will execute your chosen technique. Review the Omninet Marketing Requirements Document and discuss your findings by using the given template.

(CO1 – 30 Marks)

Section #	Problematic Phrase or Section	Reworded Phrase or Section
1.1	Abbreviations are not consistently used or explained.	Clearly define and consistently use abbreviations throughout the document. For example, explicitly mention that "CC" refers to "Credit Card."

3.1.2	Payment screen elements are vaguely described.	Provide a detailed breakdown of the elements on the Payment screen, specifying how each payment option is presented to the user.
3.1.3	Internet Browser options need clarification.	Explicitly state the criteria or mechanism for presenting the choice of browsers (Netscape, Opera, or Internet Explorer) on Windows kiosks at the Welcome screen.
3.1.5	Localization requirements lack specifics.	Specify the method for configuring each Omninet kiosk in its primary local language, providing clear guidelines for locales with multiple commonly used languages.
3.1.6	Content Control measures are vaguely mentioned.	Clearly outline the mechanisms and technologies in place for implementing site blocking and protecting against malicious code.
3.1.7	Session Termination options need clarification.	Specify the user experience and system behavior during session termination, including any notifications or prompts provided to the user.
3.1.8	Confidentiality measures are vaguely described.	Clearly articulate the steps taken to protect user confidentiality, including the process of clearing cookies, downloaded files, and URL history after session termination.
3.2.1	Software Updates lack specific details.	Provide a more detailed breakdown of the types of updates, such as operating system patches, driver updates, and payment rate table changes, and their implications for the kiosks.

3.2.2	View Kiosks information is incomplete.	Expand the information displayed for each kiosk in the call center, including additional details on crashes, reboots, and hard failures, to provide a more comprehensive view.
3.2.3	View Users details need additional context.	Clarify how call center agents access user information, particularly regarding the availability of previous session details and how paid time remaining is calculated.