**COURSE: INFORMATION SYSTEM ANALYSIS AND DESIGN**

**GROUP: 1**

**NAME: LEWIS KARIUKI REG NO: AIIM/00477/2021**

**NAME: PHILIP MUSUMBA REG NO: AIIM/00770/2022**

**NAME: CONSOLATA ADIPO REG NO: AIIM/00778/2022.**

**NAME: VICTOR MURIGI REG NO: AIIM/00751/2022**

**ASSIGNMENT: DEVELOPING AN E-BANKING SYSTEM**

1. **Requirements Analysis Process**

The requirements analysis process is important for understanding the needs and functionalities of the e-banking system involving steps like:

* Stakeholder Identification: which involves the identification of all possible entities that stand to benefit or be negatively impacted by the system this includes customers, bank employees, IT staff and regulatory boards. By asking questions among this diverse group of people, the researcher is able to meet all the needs by conducting surveys, interviews, focus groups, and workshops.
* Functional Requirements: which are the processes of identifying and documenting functional requirements Non-Functional Requirements which is the process of identifying and documenting nonfunctional requirements. Based on Kirsten’s work these are as follows; Outputs-lists the forms of information that the system has to generate such as account details, welcome page, login page, fund transfer page, bill payment page, and transaction history. Functional requirements concern aspects that do not directly contribute to system functionality, as they describe parameters associated with quality of service and performance including security, usability, and conformity with certain standards.
* Requirements Documentation: A Software Requirements Specification (SRS) document is generated for documenting all the requirements acquired. This document employs the use of simple language and icons as well as simple diagrams such as use case diagrams to describe the requirements.
* Requirements Validation: The documented requirements are discussed with the stakeholders to gain confirmation of accuracy and also completeness. It may involve the ability to show the proposal in an architectural or modelling form which may include a mockup of the system. It is the final process by which an official confirmation from various stakeholders is sought in order to caution myself that every stakeholder is on the same page with regards to how the developed system works.
* Requirements Management: An own change management process is created based on the fact of handling modifications of the requirements during the development lifecycle. This includes the identification of changes and the evaluation as well as reporting of these changes to interested parties.

**2. Systems Analysis and Design Methodology**

**Agile methodology** is chosen as the preferred approach for developing the e-banking system.

**3.Reasons and Benefits of Agile Methodology**

* Flexibility and Adaptability: One advantage of using Agile approach is that changes can be made even in the later phases which is very crucial because the banking environment is ever Lab: Changing requirements can occur at any time in a project; this option is crucial because the banking industry is constantly evolving.
* Customer Collaboration: Stakeholder feedback is provided often, to confirm that the system is meeting the need of its users.
* Incremental Delivery: Development is incremental, which means that if there is an issue with a software, it is identified early and the positive features in a software are released faster.
* Improved Quality: So, to be improving quality and avoiding defects, reviews and tests are performed daily and carried out at the end of each iteration.
* Rapid Adaptation to Change: The banking industry is rapidly changing, so it requires a methodology that will be able to adapt to the new technologies and the development of laws and regulations. Agile makes this possible.

**4. Testing Techniques**

Several testing techniques are employed to ensure the robustness, security, and usability of the e-banking system:

* Unit Testing: A part-by-part test is conducted to ensure that each of the components will function correctly. End of unit testing is to write an Automated unit test to check every portion of the code while they are being coded.
* Integration Testing: The communications between different modules or services are verified as appropriate to predict their correct behaviors and cooperative interactions. For example, checking the connectivity between payment processing and account management confirms reduction and recording of proper amount.
* User Acceptance Testing (UAT): Actual and intended customers asses the capability to ensure that it is sufficient to address their needs. Actual scenarios are used to capture our audience’s feedback regarding its usability and functionality.
* Security Testing: Vulnerability testing and compliance to security standards are important issues in the context of e-banking service. Risk management procedures such as penetration testing, code review and vulnerability scan entails the discovery of the risks.

**6. UML Technique Illustration: Use Case Diagram**

A Use Case Diagram effectively illustrates the system's functionalities and user interactions. Here's how to create a Use Case Diagram for an e-banking system:

**Actors:**

* Customer: Performs banking transactions.
* Bank Employee: Manages customer accounts and assists with transactions.
* Admin: Oversees the system and performs maintenance tasks.

**Use Cases:**

* Login: Customer logs into the system.
* View Account Balance: Customer checks their balance.
* Transfer Funds: Customer moves money between accounts.
* Pay Bills: Customer pays bills online.
* Deposit Funds: Customer adds money to their account.
* Account Management: Bank employee handles customer accounts.
* Generate Reports: Admin creates system usage reports.

**Relationships:**

* Includes: "Login" is a prerequisite for all other use cases.
* Extends: "Transfer Funds" might optionally include "Schedule Transfer".

