**ST PAULS UNIVERSITY**

**LIMURU CAMPUS**

**REGULAR**

**BCS 3106 SOFTWARE ENGINEERING**

**AMOS KILUMA BCISLMR167323**

**STACY MWEREMA BCISLMR142223**

**POWEL OTIENO BCISLMR168723**

**FAITH ONTIERI BCISLMR141423**

**LEC CECILIA NANFUKA**

**8/3/2023**

**SOFTWARE ARCHITECTURE AND DESIGN DOCUMENT**

1. Introduction

The Software architecture and design document (SAD) is crucial for clear communication and a smooth development process.This document outlines the software architecture and design for an online voting platform intended for university elections. The platform aims to increase voter participation, improve accessibility, and ensure secure and transparent elections.

1. System Requirements
   * Voter authentication
   * Candidate nomination
   * Secure ballot creation and voting mechanism
   * Real-time and final result tabulation and visualization
   * User roles and permissions management
   * Audit logging
   * User-friendly interface
2. System Architecture

The platform will be a web-based application accessible through a web browser. It will utilize a three-tier architecture with the following components:

* + *Presentation Layer:* Web interface built with HTML, CSS, and JavaScript for user interaction.
  + *Business Logic Layer*: Handles user requests, interacts with the data layer, and implements voting logic.
  + *Data Layer*: Securely stores user data, election information, and vote records.

1. Technology Stack
   * Front-end: Bootstrap
   * Back-end: Python
   * Security: access control mechanisms
2. Conclusion

This document provides a high-level overview of the software architecture and design for a university online voting platform. The proposed architecture prioritizes security, accessibility, and scalability while meeting the functional requirements of a secure and efficient voting system.

**TEST PLAN SHOWING IMPLEMENTATION OF THE SOFTWARE LIFE CYCLE**

1. Introduction

This document outlines the test plan for an online voting platform developed for university elections. It details the testing approach, activities, and resources required to ensure the platform’s functionality, security, and reliability, adhering to the Software Testing Life Cycle (STLC) phases.

1. Project Overview
   * Project Name: Online Voting Platform – University Elections
   * Client: St Paul’s university
   * Target Users: Registered students
2. STLC Phases and Activities

Requirement Analysis:

* + Reviewing and understanding the functional and non-functional requirements specifications.
  + Identifying testable requirements and defining clear pass/fail criteria.
  + Developing a traceability matrix linking requirements to test cases.

Test Design:

* + Designing various test cases covering positive, negative, and edge-case scenarios.
  + Utilizing equivalence partitioning and boundary value analysis for comprehensive testing.
  + Creating test scripts with detailed steps and expected outcomes for manual testing.
  + Implementing automated testing frameworks for repetitive tasks and regression testing.

Test Environment Setup:

* + Configuring a dedicated test environment mirroring the production environment.
  + Utilizing the test data simulating real-world usage patterns and user diversity
  + Implementing security measures.

Test Execution:

* + Test cases manually and through automated tools based on the testing strategy.
  + Document test results, including screenshots, logs, and defect descriptions.
  + Prioritize critical and major defects for immediate resolution.

Test Reporting:

* + Generating test reports for each phase, summarizing test execution results, identified defects, and progress.
  + Execute.
  + Including effect severity, resolution status, and retesting results.
  + Presenting final test report with overall testing conclusion, recommendations, and closure of all defects.

Test Closure:

* + Reviewing and approving all defects with resolution evidence.
  + Conducting regression testing after defect fixes to ensure stability.
  + Obtaining final approval from stakeholders before production deployment.

**CONCLUSION**

The system allows users to sign up and therefore log in to interact with the system. The system allows the Students, delegates and candidates to interact virtually with each other. The system also enables delegates to do the preliminary voting and even voting online and the results are displayed for all users to view. The system allows every user to like the candidates of their preferences and the most liked candidate is the most popular. The system is able to compute reports for the whole election process.

Online voting offers convenience to the voter and considerable ease to election administrators as they can get election results out more quickly than conventional methods of manual voting. Even though computerized voting systems have a number of advantages over manual ones, there a few challenges that must be overcome in order that their benefits may be fully realized.

*References*

1. Al-Ameen, A.; Talab, S.A., “E-voting systems vulnerabilities,” Information Science and Digital Content Technology (ICIDT), 2012 8th International Conference on , vol.1, no., pp.67,73, 26-28 June 2012 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6269229&isnumber> =6269212
2. Cui Zhe; Dai Xiang, “A practical distributed electronic voting system,” Electric Information and Control Engineering (ICEICE), 2011 International Conference on , vol., no., pp.1095,1099, 15-17April2011doi:10.1109/ICEICE.2011.5777548URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5777548&isnumber =5776798
3. Gritzalis, D.(2002). Secure Electronic voting. Seventh Computer Security Incident Response Team Workshop Syros, Greece.
4. Krejcie, R. & Morgan D. (1970). Determining Sample Size for Research Activities Educational and Psychological Measurement.
5. Melanie Volkamer (2010), ―Electronic Voting in Germany‖, Data Protection in a Profiled World, DOI 10.1007/978904818865910, © Springer Science+Business Media B.V. 2010
6. TahaKh. Ahmed and Mohamed Aborizka (2011), ―Secure Biometric E-Voting Scheme‖, ICICIS 2011, Part I, CCIS 134, pp. 380–388, 2011. © Springer-Verlag Berlin Heidelberg 2011
7. C-DOW]:ITEA2 CREATE ip10020 amended FPP, December 2011.[INT-GSC]:Interfaces of the Generic Substrate Carrier.[KRU41]: The “4+1” view model of software architecture, Philippe Kruchten, November 1995,(<http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4>P1.pdf).
8. [RWAES]S. Meyer, K. Sperner, C. Magerkurth “Towards Real-World Aware Enterprise Systems”, IEEE MASS 2011, Valencia, Spain.[IoT-IRM]Internet-of-Things Architecture IoT-A Project Deliverable D1.2 –Initial Architectural Reference Model for IoT (<http://www.iot-a.eu/public/public> Documents/documents-1/1/1/d1.2/atdownload/file).
9. ]Microsoft.(2009).“Chapter16:QualityAttributes,”inMicrosoftArchitectureApplication Guide,2nded.[Online].Available:http://msdn.microsoft.com/en­us/library/ee658094.aspx
10. Smith, A.D. and J.S. Clark, 2005. Revolutionizing theVoting process through online strategies. OnlineInfor. Rev., 29(5): 513530.
11. Spannaus, E., 2004. Electronic Voting is Threat to theConstitution. Executive Intelligence Review.