NUVote: The National University Voting Platform

David, Neo M.
College of Computing and Information Technology
INF-224
Pasay, Philippines
davidnm@students.national-u.edu.ph

Catapang, Gabriel M.
College of Computing and Information Technology
INF-224
Manila, Philippines
catapanggm@students.national-u.edu.ph

Abstract – NUVote is a voting system specifically designed to cater to student government elections held within National University-Manila. Like other online voting platforms, NUVote can be used to create and manage elections. However, it uniquely incorporates a more robust security system, ensuring that only university students can access it, thus safeguarding the integrity of the election and addressing other related issues. The system includes data structures like hashing and arrays, and for the user interface design, Canva was employed.

Keywords – Election, Voting System, Student Government, Hashing, Arrays

I. INTRODUCTION

Within educational institutions such as universities, student governments are indeed a vital organization that serves as the link between the student body and the school administration. It offers students the opportunity to engage in the decision-making process which affects their learning environment, which is why holding elections with utmost honesty and integrity is crucial for an effective leadership and communication process.

NUVote is a Java application designed specifically for holding elections, with a strong emphasis on ensuring the security and integrity of the entire process. NUVote, like other applications, is a powerful tool for conducting a transparent and legitimate electoral process, primarily for university use. It was thoroughly customized to meet the specific needs of academic institutions, providing a secure and user-friendly platform for administering elections, with a primary focus on student government elections open to the entire student body of the university.

A. OBJECTIVES

The aim of the project is to provide a platform which allows a seamless and secure administration of the electoral process inside the university setting with a particular emphasis on the student government elections. To accomplish this goal, the project outlines the following objectives in particular:

- Hide the identity of the voter and protect their privacy, allowing them to express their preferences without the fear of their choice being revealed.
- Enable users to vote for the candidates they wish to elect and display the positions available for the election and the candidates running for each of those positions.
- Provide a user-friendly interface for an easy navigation which also promotes an increase of student participation in the electoral process.

- ♣ Utilize students' unique university-issued student numbers as the basis for creating their accounts, effectively filtering out external access and ensuring an equitable election process for all participating candidates.

B. SCOPE AND LIMITATIONS

As per the proposal, the system aims to enhance the user experience while conforming to certain restrictions by offering specific functionalities. The system will comprise an easy-to-use login interface for both administrators and users, with the option to create accounts using unique student numbers. Administrators will be able to manage jobs and candidates within the system. To secure the confidentiality of users, the system will not collect basic personal information. However, the system will have some limitations, including the absence of features like personal data storage, password reset functionality, and the usage of only a limited set of university-issued student numbers. Additionally, administrators will have predefined accounts. This set of inclusions and exclusions establishes the framework for the system's operation.

C. REVIEW OF RELATED LITERATURE

Student governments, often referred to as the student council plays a vital role in the ecosystem of an educational institution as they serve as the voice of the student body. According to a study by Rogaleva (2011, as cited in Dorozhkin et al, 2016), the overall atmosphere and character of the university's learning environment are greatly influenced by the existence of the student government. Most of the planning and execution of various programs are made by the student government with the cooperation of faculty and administrators. Also, according to (Dorozhkin et al, 2016), the establishment and expansion of student government bodies benefits students' professional development by creating an environment in which students can further develop their skills and competencies in preparation for their future careers. That is why elections are conducted each academic year, not only to facilitate a change in those currently holding positions but also to provide an opportunity for others to gain experience and showcase their leadership skills.

In terms of conducting an election, the process would either be a manual voting or an automated one. Conducting a manual election is not recommended, especially for elections with a high number of anticipated voters such as elections held inside an academic institution. Which is why most institutions rely on an automated election. According to (Ofuri-Dwumfuo & Paatey, 2011), When compared to the traditional manual voting process, online voting provides greater convenience to voters and significantly streamlines the responsibilities of election administrators, allowing for a faster release of election results. However, with this method, election administrators may encounter certain challenges, including concerns about voter privacy and the overall integrity of the election.

II. METHODOLOGY

The foundation of the system was built through the two data structures: HashMaps and ArrayLists in Java. These data structures are essential in efficiently implementing the overall functionality of the program: putting user credentials in a map and accessing candidates information though an arraylist.

Figure 1: Voting System Flowchart

This flowchart represents the included processes in NUvote.

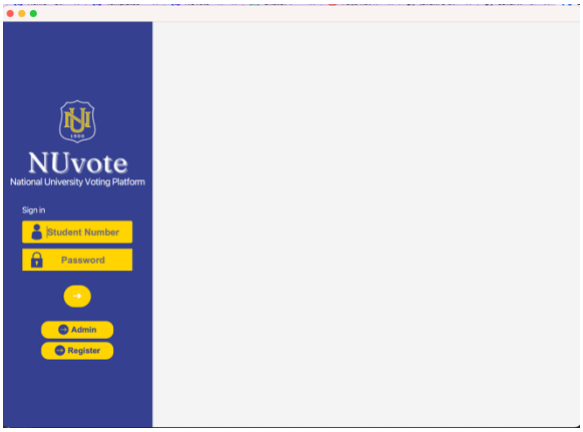

Figure 2: Landing Page

On this page, the user will identify whether they are an admin or a student. If they are a student and have already registered their student number and email before, they can log-in through their registered credentials.

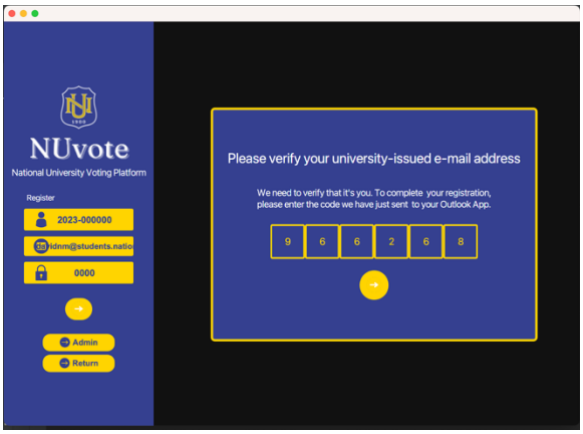

Figure 3: Registration and Verification of Account

The program has a predefined data of the listed student numbers paired with their school emails. The two data, student number and school email, are stored in HashMap data structure in Java. The student number will act as the key on the map, and the school email will be the value.

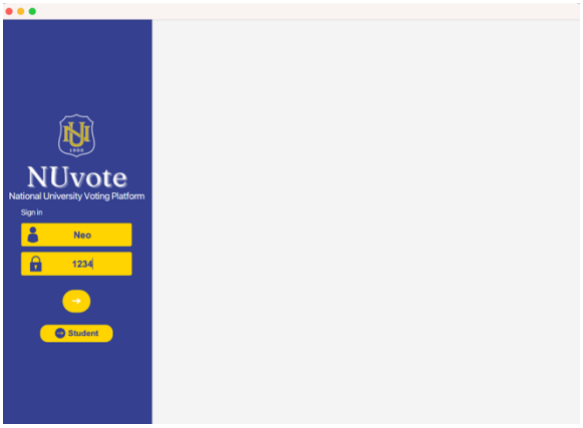

Figure 4: Admin Log-in

The administrator side has predefined accounts as they will be the one to create elections.

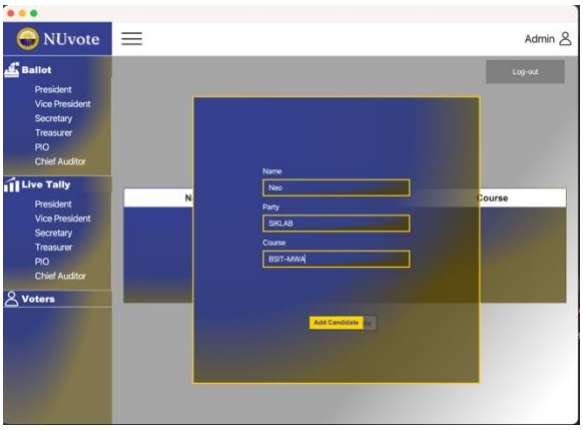

Figure 5: Adding a Candidate

The admin can add a candidate by clicking the ballot option on the sidebar where the candidate positions are also listed. Name, party, and course are the required information in adding a candidate.

Figure 6: Added candidates

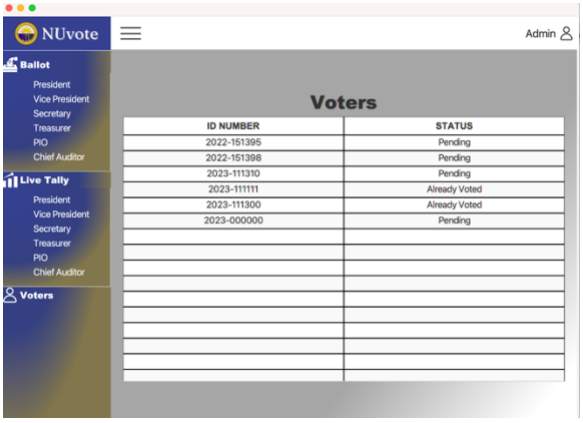

Figure 7: Registered Voters

The admin can also view the registered voters in the sidebar, ID Number and the voting status are the listed information in the table. These information are stored using a hashmap.

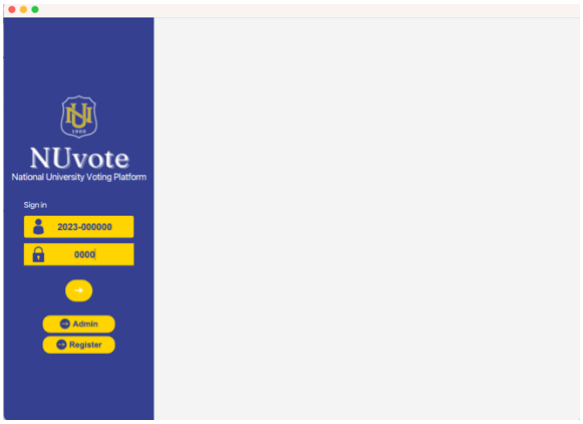

Figure 8: Student Log-in

After a student's registration, they can now input their credentials on this page.

The purpose of outlook verification is to provide security and integrity on the election of NU.

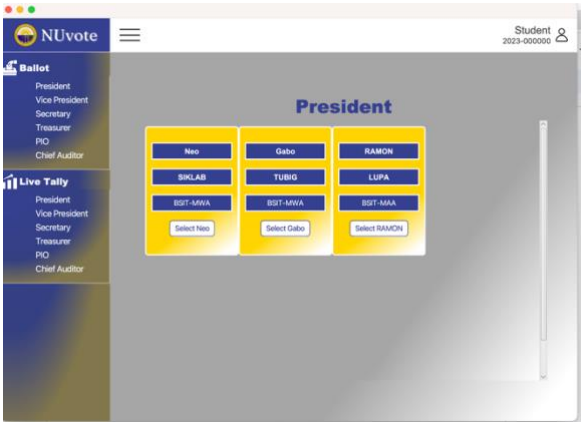

Figure 9: Candidate Voting

After logging in, the student can now vote for the added candidates earlier. The student number can also be seen on the top right of the application. Voting is dong by accessing candidate information in an arraylist.

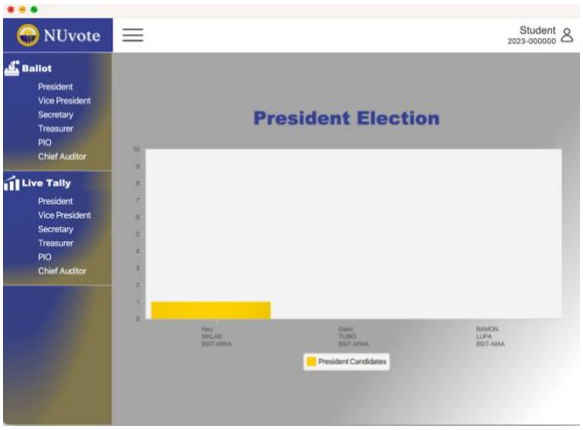

Figure 10: Live Tally

Both students and admins have interfaces for the live tally. The live tallies are created through an ArrayList in Java where the program is able to access the candidates and their information directly such as their vote counts.

III. RESULTS AND DICUSSIONS

The result of the system has shown the importance of implementing the correct data structure for a problem. The system was successful as it was able to exhibit the voting functionality with minimal lags. The difficultness of the system has made us learn a lot in terms of programming and data structures. The system was intrinsically dynamic as it is a voting system where multiple components adjust to minimal events. Overall, creating the system piqued our interest on planning the appropriate data structure and gave us ideas about dynamic software.

IV. CONCLUSION

On this project, we created a successful voting system that implements appropriate data structures. The system was able to implement an additional security for a voting platform in the school, store candidates in the program, and access and vote for the stored candidates. We can also conclude that there is a need for a decent knowledge in data structure to implement dynamic programs.

V. ACKNOWLEDGEMENT

We would like to express gratitude to Professor Elsie V. Isip for effectively teaching us about this subject, Data Structures and Algorithms. Our professor's guidance along with the online resources have been a significant factor on conducting this problem-solving. On top of that, we would also like to acknowledge the individuals who contributed on finishing this project: ourselves.

REFERENCES

Dorozhkin, E. M., Zaitseva, E. V., & Tatarskikh, B. Y. (2016). Impact of student government bodies on students' professional development. International Electronic Journal of Mathematics Education, 11(7), 2666-2677.

Ofori-Dwumfuo, G. O., & Paatey, E. (2011). The design of an electronic voting system. Research Journal of Information Technology, 3(2), 91-98.