

## UTHM SUKSIS Fingerprint Voting System

**Adibah Anisa Zulkifli<sup>1</sup>, Ruhaya Ab. Aziz<sup>1\*</sup>**

<sup>1</sup>Fakulti Sains Komputer dan Teknologi Maklumat,  
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2023.04.02.072>

Received 22 June 2023; Accepted 25 October 2023; Available online 30 November 2023

**Abstract:** UTHM SUKSIS Fingerprint Voting System is an application that will be used by SUKSIS trainees on election day to form the new highest council. It was developed because the election method used now is less systematic. There are three objectives of this system which are to analyze and design the UTHM SUKSIS Fingerprint Voting System, to develop a UTHM SUKSIS Fingerprint Voting System, and to test the UTHM SUKSIS Fingerprint Voting System. The waterfall model is used to ensure that the system is thoroughly analyzed before development begins. The result of this system is a system that has full transparency and has reliable processes.

**Keywords:** Voting, Waterfall Model, Fingerprint

### 1. Introduction

UTHM SUKSIS also known as Kor Sukarelawan Siswa Siswi for Tun Hussein Onn Malaysia University was established on July 17, 2006, with the aim of training trainees from university's male and female students. Since its inception, SUKSIS has developed and progressed in line with the needs of the country. It is estimated that a total of 60 students consisting of 30 male students and 30 female students have passed the Physical Test. Since the 17th century, elections have been the primary method used to carry out representative democracy in the modern world. Every year, SUKSIS will hold an election day to constitute the new highest council. It is very important for the progress of trainees in carrying out their responsibilities. However, it is inconvenient for SUKSIS to elect as a consequence of the existing election process. Therefore, voting system needs to be developed to ease the problem stated. This project focuses on increasing the effectiveness of the voting system and data accuracy.

### 2. Related Work

The system incorporates the theory of secure authentication mechanisms. The concept revolves around the implementation of encryption techniques and multi-factor authentication to safeguard the voting system from unauthorized access. This theory acknowledges the significance of protecting

sensitive voting data, maintaining the confidentiality and integrity of the system and upholding the principles of privacy and security.

## 2.1 Fingerprint System

Fingerprints consist of a succession of ridges and furrows on the surface of the finger and designs such as swirls, loops or arches encircling the core which make them distinctly different for each person[1]. In the process of matching, all properties of fingerprints are compared, not focusing on individual points exclusively. The features can contain sub-areas of special importance such as ridge thickness, curvature, or density. The fingerprint's small parts and their distances from one another are distinguished during the enrolment procedure.

## 2.2 Information Technology

Information technology is utilized in the UTHM SUKSIS Fingerprint Voting System to create a secure and effective voting process. It incorporates biometric fingerprint recognition to verify voters' identities using device tokens, eliminating the need for traditional identification methods. The system employs a database management system to store and manage voter information and voting data, ensuring accurate and organized data. Secure authentication mechanisms protect against unauthorized access, maintaining data integrity. This mobile application development allows voters to conveniently cast their votes using smartphones, enhancing user experience.

## 2.3 Comparison with Existing Systems

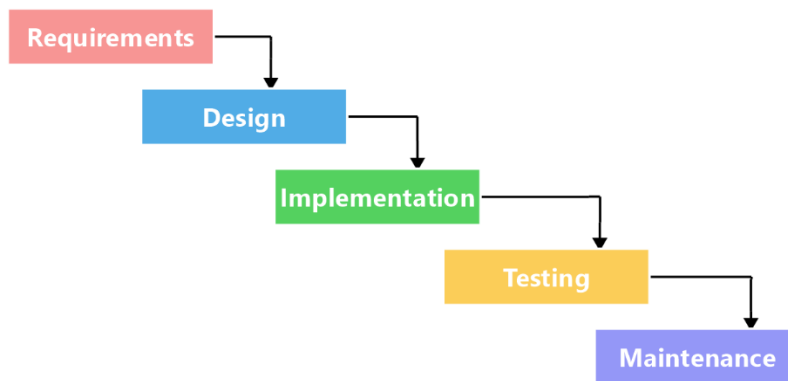
The evaluation of the existing system was conducted by referencing the three systems that have already been developed. These systems were chosen as benchmarks to enhance the quality of UTHM SUKSIS Fingerprint Voting System. The selected systems include Voting System Application, Voting4Schools, and Digital Democracy Application. A comparative analysis was conducted, evaluating the features of these three systems in relation to the proposed system. Table 1 below shows the comparison of Existing System and Proposed System.

**Table 1: Comparison of Existing System and Proposed System**

Feature/System	Voting System Application	Voting4Schools	Digital Democracy Application	UTHM SUKSIS Fingerprint Voting System
Login	Yes	Yes	Yes	Yes
Manage Voter Information	No	Yes	No	Yes
Manage Candidate Information	No	Yes	Yes	Yes
Manage Election	Yes	Yes	Yes	Yes
Calculate Result	Not available	Not available	Available	Available
View Result	Yes	Yes	Yes	Yes
Generate Report	No	No	No	Yes
Platform	Android	IOS	IOS	Android
Authentication	Email	Email	Email	Fingerprint and Email

### 3. Methodology

The Waterfall Process Model is a well-established and widely used software development method that follows a sequential and linear progression through various phases which are requirement analysis, system design, implementation, testing and maintenance. It is distinguished by its structured nature, in which each phase is completed before proceeding to the next phase, ensures a systematic and well-organized development process[5]. This approach emphasizes the importance of thorough planning and documenting at the early stages to reduce risks and modifications later in the project lifecycle. However, it is important to note that the Waterfall Process Model has been subject to criticism for being rigid and possibly having trouble adapting changing requirements and stakeholder feedback[6]. Despite its flaws, the Waterfall Process Model is nevertheless useful for projects with clearly defined and consistent needs since it may set clear milestones and allow for efficient resource allocation.



**Figure 1: Waterfall Process Model**

#### 3.1 System Development Workflow

Table 2 shows the tasks carried out during each phase of the Waterfall model for UTHM SUKISIS Fingerprint Voting System.

**Table 2: Activity in each of the phase**

Phase	Task	Output
Requirement Analysis	<ul style="list-style-type: none"> <li>Proposed project title.</li> <li>Identified System requirement.</li> </ul>	<ul style="list-style-type: none"> <li>Project proposal.</li> <li>Develop Gantt Chart.</li> </ul>
System Design	<ul style="list-style-type: none"> <li>Design Prototype.</li> <li>Identify the software and hardware requirement.</li> </ul>	<ul style="list-style-type: none"> <li>Simple prototype to meet the requirement.</li> </ul>
Implementation	<ul style="list-style-type: none"> <li>Develop interface for the system.</li> <li>Develop database of the system.</li> </ul>	<ul style="list-style-type: none"> <li>Interface of the system is developed with functional features.</li> <li>Database of the system is developed.</li> </ul>
Testing	<ul style="list-style-type: none"> <li>Detect defect or bug</li> </ul>	<ul style="list-style-type: none"> <li>Functional system.</li> </ul>
Maintenance	<ul style="list-style-type: none"> <li>Documentation of the project.</li> </ul>	<ul style="list-style-type: none"> <li>Complete project report.</li> <li>Complete functional system.</li> </ul>

### 3.2 Requirements Phase

The requirement phase is a vital stage where the project requirements are obtained, examined, and recorded. Understanding the requirements and expectations of the stakeholders, including UTHM SUKSIS trainees and trainers is the main goal of this phase. In order to identify and prioritize the functional and non-functional requirements of the software system, interviews and discussions are conducted. Requirements are then documented in a clear and concise manner to make sure they are quantifiable and traceable.

These are the specifications that the system must meet to satisfy the end user's basic needs. As a requirement of the contract, all these functionalities must be built into the system. A functional requirement defined what has been done by identifying the necessary activity, task, or action that must be accomplished [4]. Table 3 shows the functional requirement and its functionalities.

**Table 3: Functional requirements**

Modules	Functionalities
Login	<ul style="list-style-type: none"> <li>• The application can verify the users.</li> <li>• The application should redirect registered users to the respective homepage.</li> <li>• The application will display error message when users enter wrong email and password.</li> <li>• The application shall save user device token in the database.</li> </ul>
Manage voter information	<ul style="list-style-type: none"> <li>• The application shall display the voter information.</li> <li>• The application shall allow the administrator to add voter information.</li> <li>• The application shall allow the administrator to remove voter information.</li> </ul>
Manage candidate information	<ul style="list-style-type: none"> <li>• The application shall display the candidate information.</li> <li>• The application shall allow the administrator to add candidate information.</li> <li>• The application shall allow the administrator to candidate voter information.</li> </ul>
Manage election	<ul style="list-style-type: none"> <li>• The application shall allow the user to scan their fingerprint biometric.</li> <li>• The application shall compare the voter device token and votes device token.</li> <li>• The application shall allow the user who has the same device token access to the election page.</li> <li>• The application shall allow the user to vote.</li> <li>• The application shall save the vote data in the database.</li> </ul>

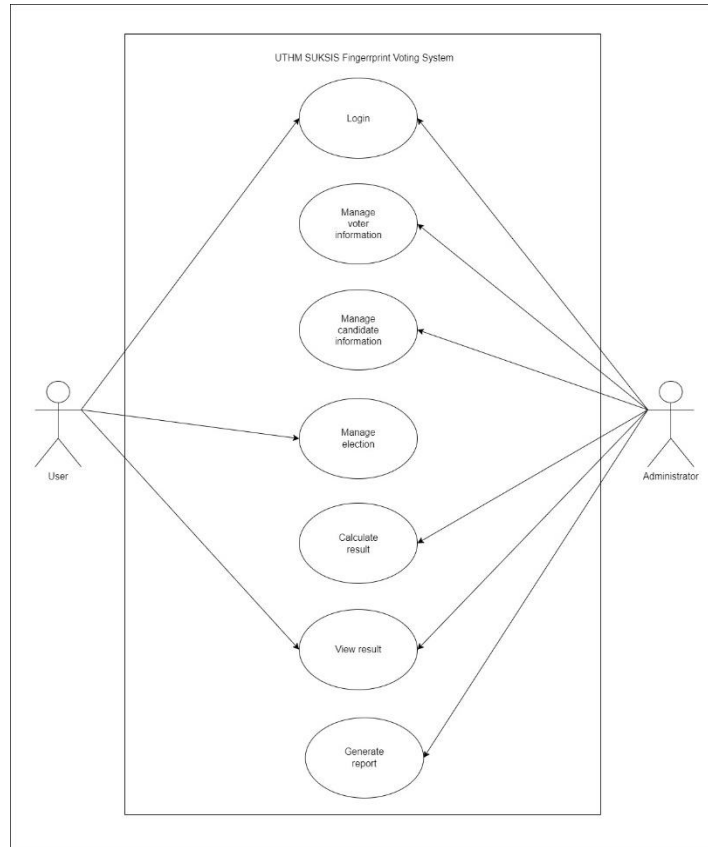
Modules	Functionalities
	<ul style="list-style-type: none"> <li>• The application shall pop up the confirmation box to confirm the vote.</li> <li>• The application shall show already voted box if user voted for more than one candidate in a position.</li> </ul>
Calculate result	<ul style="list-style-type: none"> <li>• The application shall allow the administrator to view the vote percentage.</li> <li>• The application shall allow the administrator to upload the winner's photo.</li> </ul>
View result	<ul style="list-style-type: none"> <li>• The application shall display the photos of new highest council.</li> </ul>
Generate report	<ul style="list-style-type: none"> <li>• The application shall display the photos of new highest council.</li> </ul>

Non-functional requirements specify the criteria that can be used to judge the operation of a system, rather than specific behaviors[4]. A non-functional requirement is important to ensure the overall mobile application usability and effectiveness. Table 4 shows the non-functional requirements and its description.

**Table 4: Non-Functional requirements**

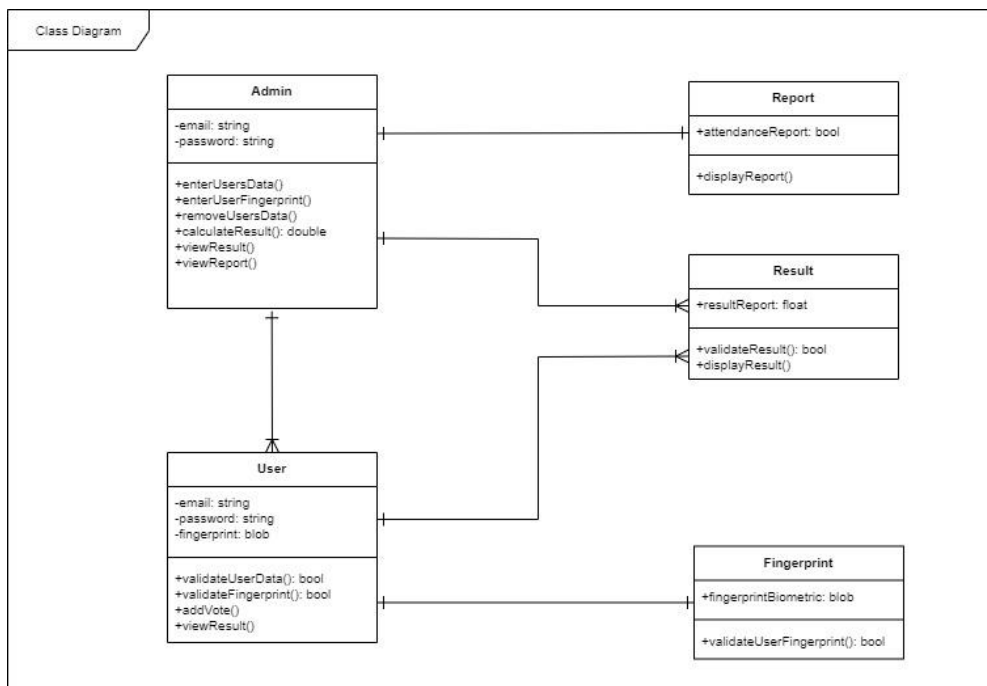
Requirements	Description
Security and privacy	<ul style="list-style-type: none"> <li>• The system can be protected from an authorized party.</li> </ul>
Reliability	<ul style="list-style-type: none"> <li>• The system shall never crash or hang, other than as the result of an operating system error.</li> </ul>
Integrity	<ul style="list-style-type: none"> <li>• The database of the application will be kept properly and secured by the system from any corruption and non-readable.</li> </ul>
Availability	<ul style="list-style-type: none"> <li>• The system is sure readily can be operated and available to use anytime.</li> </ul>

The use case diagram was created as part of the analysis to show the overall purpose and components of the system. It demonstrates the procedure to identify, clarify and organize system requirements of mobile application for UTHM SUKSIS Fingerprint Voting System.



**Figure 3: Use Case Diagram for UTHM SUKSIS Fingerprint Voting System**

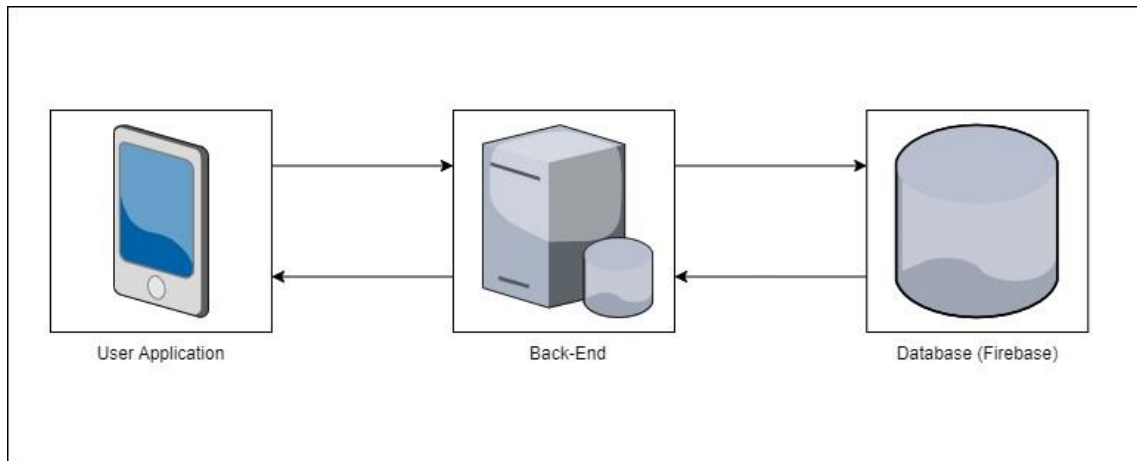
Class diagram is defined as another type of UML Specification diagram which relates to the database more than other UML diagrams. There are a total of five classes identified in the proposed application which are Admin, User, Report, Result, and Fingerprint. Each class has its own attributes and operations or methods to be performed and run.



**Figure 3: Class Diagram for UTHM SUKSIS Fingerprint Voting System development**

### 3.3 System Design Phase

Several activities are undertaken to define the architecture and structure of the software system. The main goal is to produce a thorough design defining the system's development and operation. First, a thorough system design is created by analyzing and translating the requirements acquired during earlier steps. This entails figuring out the essential elements, modules, and connections between them. In the design phase, the right frameworks, tools, and technologies must be chosen in accordance with the project requirements. A crucial aspect is user interface design, which involves designing intuitive and approachable interfaces that enhance the entire user experience. To ensure effective data storage and retrieval, database design and data modelling are also carried out. Figure 4 below shows the overall system architecture for the development of UTHM SUKSIS Fingerprint Voting System.



**Figure 4: Overall system architecture**

Database schema is defined as a text form of the tables in the database.

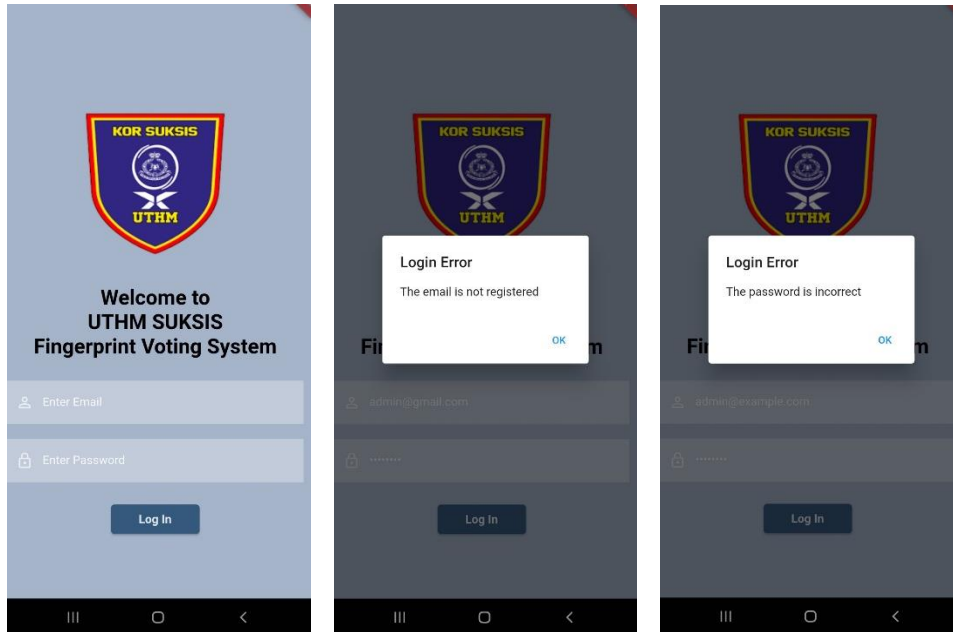
- I. **Admin** (email, password)
- II. **User** (email, password)
- III. **Fingerprint** (fingerprintData, voterId, timeStamp)
- IV. **Result** (candiadateName, voteCount)
- V. **Report** (voterId, status, timeStamp)

### 3.4 Implementation Phase

The implementation phase is a critical stage in the software development lifecycle, where the software design is transformed into a fully functional system. It involves writing actual code from the theoretical concepts and specifications into actual code and deploying the software in intended setting. This process requires attention to detail, adherence to coding standards and best practices, and effective project management.

#### 3.4.1 Implementation of Login Module

The user will need to enter their registered email and password to access into the system. When user which is SUKSIS's trainee logs in to the system, their device token will be saved in the database.



**Figure 5: Interface of Login Module**

Figure 5 above shows UTHM SUKSIS Fingerprint Voting System login interface which SUKSIS trainees will need to enter their registered email and password. The pop-up notification is also displayed on this page if user entered either wrong email or password.

```

1  const SizedBox(height: 30),
2      reusableTextField(
3      "Enter Email",
4      Icons.person_outline,
5      false,
6      _emailTextController,
7      ),
8      const SizedBox(height: 20),
9      reusableTextField(
10     "Enter Password",
11     Icons.lock_outline,
12     true,
13     _passwordTextController,
14     ),
15     const SizedBox(height: 20),
16     SizedBox(
17     width: 120,
18     height: 70,
19     child: firebaseUIButton(context, "Log In", checkAndLogin),
20     ),

```

```

1  Future<void> signInWithEmailAndPassword(
2      String email, String password) async {
3      try {
4          await FirebaseAuth.instance.signInWithEmailAndPassword(
5              email: email,
6              password: password,
7          );
8          await storeDeviceToken();
9          print('Login successful!');
10     } catch (e) {
11         // Handle login errors
12         print('Login error: $e');
13     }
14 }

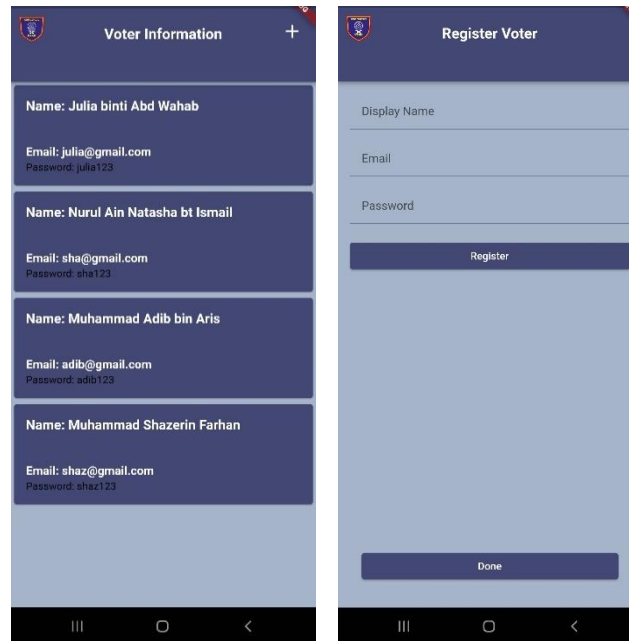
```

**Figure 6: Code segment of Login Module**

Figure 6 above shows the code segment for login module.

### 3.4.2 Implementation of Manage Voter Information Module

The manage voter information module function as the page for admin to register voter by entering their name, email, and password.



**Figure 7: Interface of Manage Voter Information Module**

Figure 7 above shows the interface of manage voter information module.

```

1 void signUp() async {
2   String displayName = displayNameController.text.trim();
3   String email = emailController.text.trim();
4   String password = passwordController.text.trim();
5
6   print('displayName: $displayName');
7   print('email: $email');
8   print('password: $password');
9
10  if (displayName.isEmpty || email.isEmpty || password.isEmpty) {
11    ScaffoldMessenger.of(context).showSnackBar(
12      SnackBar(content: Text('Please enter name, email, and password')),
13    );
14    return;
15  }

```

```

1 Future<void> saveVoterDataToFirestore(
2   String displayName,
3   String email,
4   String password,
5 ) async {
6   try {
7     String collectionPath = 'voters';
8     User? currentUser = FirebaseAuth.instance.currentUser;
9
10    if (currentUser != null) {
11      String uid = currentUser.uid;
12
13      await FirebaseFirestore.instance.collection(collectionPath).doc(email).set({
14        'displayName': displayName,
15        'email': email,
16        'password': password,
17      });
18
19      displayNameController.clear();
20      emailController.clear();
21      passwordController.clear();
22
23      ScaffoldMessenger.of(context).showSnackBar(
24        SnackBar(content: Text('Data saved successfully')),
25      );
26    }
27    catch (e) {
28      print('Error saving voter data: $e');
29    }
30  }
31 }

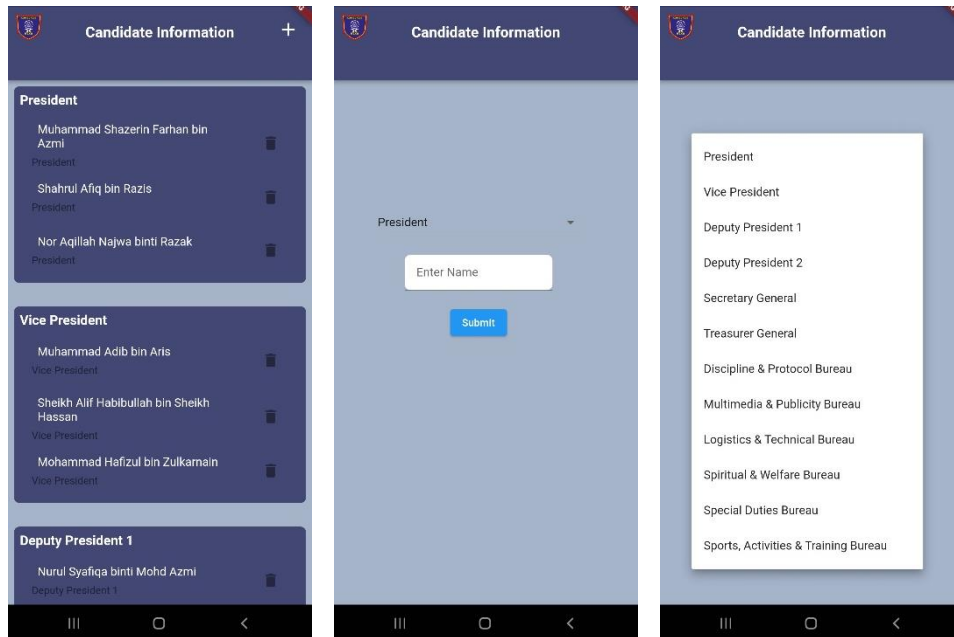
```

**Figure 8: Code segment of Manage Voter Information Module**

Figure 8 above shows the code segment for manage voter information module.

### 3.4.3 Implementation of Manage Candidate Information Module

The manage candidate profile functions as the page for the administrator to register candidate by entering selecting their position and entering their name.



**Figure 9: Interface of Manage Candidate Information Module**

Figure 9 above shows the interface for manage candidate information module.

```

1 children: [
2     DropdownButton<String>(
3         value: selectedPosition,
4         onChanged: (String? newValue) {
5             setState() {
6                 selectedPosition = newValue!;
7             });
8         },
9         dropdownColor: Colors.white,
10        items: <String>[
11            'President',
12            'Vice President',
13            'Deputy President 1',
14            'Deputy President 2',
15            'Secretary General',
16            'Treasurer General',
17            'Discipline & Protocol Bureau',
18            'Multimedia & Publicity Bureau',
19            'Logistics & Technical Bureau',
20            'Spiritual & Welfare Bureau',
21            'Special Duties Bureau',
22            'Sports, Activities & Training Bureau',
23        ].map<DropdownMenuItem<String>>((String value) {
24            return DropdownMenuItem<String>(
25                value: value,
26                child: Text(value),
27            );
28        }).toList(),
29    ),

```

```

1 void saveDataToFirestore(
2     String name,
3     String position,
4 ) async {
5     String collectionPath = 'candidates';
6
7     await FirebaseFirestore.instance.collection(collectionPath).doc(name).set({
8         'name': name,
9         'position': position,
10    });
11
12    nameController.clear();
13
14    ScaffoldMessenger.of(context).showSnackBar(
15        SnackBar(content: Text('Data saved successfully')),
16    );
17 }

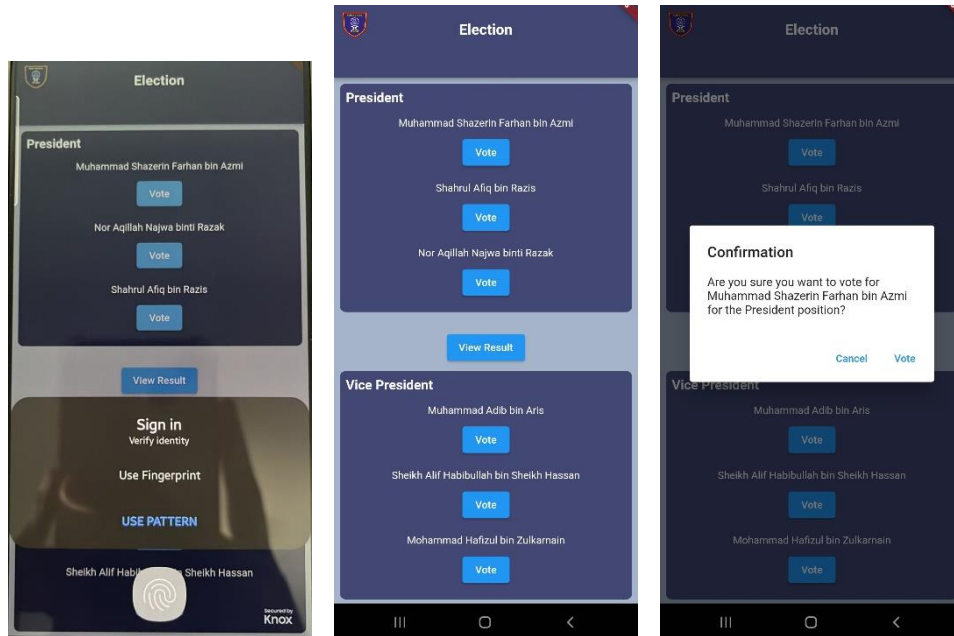
```

**Figure 10: Code segment of Manage Candidate Information Module**

Figure 10 above shows the code segment for manage voter information module.

### 3.4.4 Implementation of Manage Election Module

The manage election page is a secure page where the fingerprint verification will be activated when user accessing the page. The user needs to scan their fingerprint to initiate the voting process. Additionally, when the user selects the “Vote” button, a confirmation box will appear, requesting their confirmation to cast the vote. It is important to note that only one candidate can be selected per position.

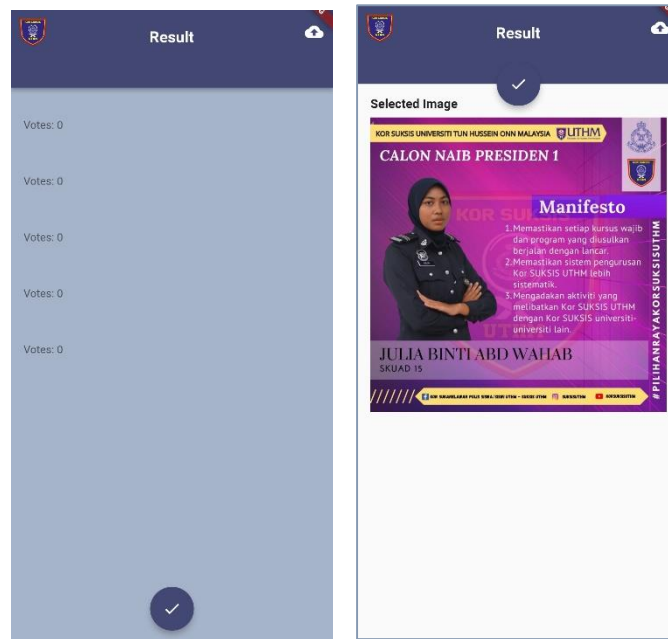


**Figure 11: Interface of Manage Election Module**

Figure 11 above shows the interfaces for manage election module.

### 3.4.5 Implementation of Calculate Result Module

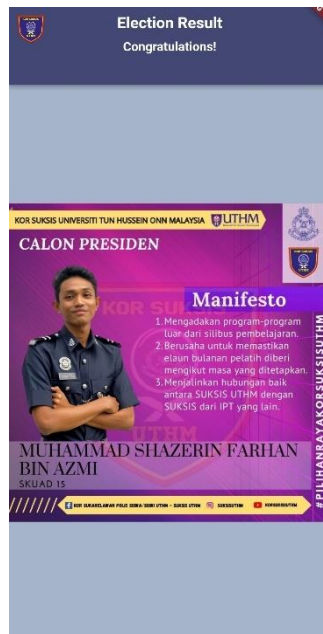
The calculate result page where the vote count from the database will be displayed. The upload image function also requires as the administrator will upload photo of the new highest council to the database.



**Figure 13: Interface of Calculate Result Module**

### 3.4.6 Implementation of View Result Module

The photos of new highest council of UTHM SUKISIS Fingerprint Voting System are retrieved from the database and displayed in this page.



**Figure 15: Interface of View Result Module**

### 3.4.7 Implementation of Generate Report Module

This page displays the attendance of voters participating in the UTHM SUKISIS Fingerprint Voting System. Trainees from UTHM SUKISIS who have not logged into the system or have not cast their votes will be marked as absent. The attendance status serves as an indicator of active participation, ensuring that all trainees' presence and engagement in the voting process are accurately recorded.



**Figure 17: Interface of Generate Report Module**

### 3.5 Testing Phase

A crucial step in the software development process is Functional Testing and User Acceptance Testing (UAT), which ensures that the system meets the expectations and requirements of its target audience. UAT involves exposing the software to real-life situations and enabling end-users or their delegates to assess its usability, functionality, and overall user experience. Functional testing and UAT acts as a final step to establish stakeholder confidence in the software’s readiness for deployment, allowing them to evaluate how well it satisfies user needs and identify any potential problems[7]. Functional testing and User Acceptance Testing gives end users which are UTHM SUKISIS trainees and trainers crucial input and insights that can contribute to the software’s improvement and will ensure a successful user adoption.

#### 3.5.1 Functional Testing

Table 5 below shows the output of functional testing for login module.

**Table 5: Functional Testing for Login Module**

Test ID	Requirement ID	Explanation	Result (Pass/Fail)
TC01-01	FR01-01	<ul style="list-style-type: none"> <li>The application can verify the users.</li> </ul>	Pass
TC01-02	FR01-02	<ul style="list-style-type: none"> <li>The application should redirect registered users to the respective homepage.</li> </ul>	Pass
TC01-03	FR01-03	<ul style="list-style-type: none"> <li>The application will display error message when users enter wrong email and password.</li> </ul>	Pass
TC01-04	FR01-04	<ul style="list-style-type: none"> <li>The application shall save user device token in the database.</li> </ul>	Pass

Table 6 below shows the output of functional testing for managing voter information module.

**Table 6: Functional Testing for Manage Voter Information Module**

Test ID	Requirement ID	Explanation	Result (Pass/Fail)
TC02-01	FR02-01	<ul style="list-style-type: none"> <li>The application shall display the voter information.</li> </ul>	Pass
TC02-02	FR02-02	<ul style="list-style-type: none"> <li>The application shall allow the administrator to add voter information.</li> </ul>	Pass
TC02-03	FR02-03	<ul style="list-style-type: none"> <li>The application shall allow the administrator to remove voter information.</li> </ul>	Pass

Table 7 below shows the output of functional testing for managing candidate information module.

**Table 7: Functional Testing for Manage Candidate Information Module**

Test ID	Requirement ID	Explanation	Result (Pass/Fail)
TC03-01	FR03-01	<ul style="list-style-type: none"> <li>The application shall display the candidate information.</li> </ul>	Pass
TC03-02	FR03-02	<ul style="list-style-type: none"> <li>The application shall allow the administrator to add candidate information.</li> </ul>	Pass
TC03-03	FR03-03	<ul style="list-style-type: none"> <li>The application shall allow the administrator to candidate voter information.</li> </ul>	Pass

Table 8 below shows the output of functional testing for managing election module.

**Table 8: Functional Testing for Manage Election Module**

Test ID	Requirement ID	Explanation	Result (Pass/Fail)
TC04-01	FR04-01	<ul style="list-style-type: none"> <li>The application shall allow the user to scan their fingerprint biometric.</li> </ul>	Pass
TC04-02	FR04-02	<ul style="list-style-type: none"> <li>The application shall compare the voter device token and votes device token.</li> </ul>	Pass
TC04-03	FR04-03	<ul style="list-style-type: none"> <li>The application shall allow the user who has the same device token access to the election page.</li> </ul>	Pass
TC04-04	FR04-04	<ul style="list-style-type: none"> <li>The application shall allow the user to vote.</li> </ul>	Pass
TC04-05	FR04-05	<ul style="list-style-type: none"> <li>The application shall save the vote data in the database.</li> </ul>	Pass
TC04-06	FR04-06	<ul style="list-style-type: none"> <li>The application shall pop up the confirmation box to confirm the vote.</li> </ul>	Pass

TC04-07	FR04-07	<ul style="list-style-type: none"> <li>The application shall show already voted box if user voted for more than one candidate in a position.</li> </ul>	Pass
---------	---------	---	------

Table 9 below shows the output of functional testing for calculate result module.

**Table 9: Functional Testing for Calculate Result Module**

Test ID	Requirement ID	Explanation	Result (Pass/Fail)
TC05-01	FR05-01	<ul style="list-style-type: none"> <li>The application shall allow the administrator to view the vote percentage.</li> </ul>	Pass
TC05-02	FR05-02	<ul style="list-style-type: none"> <li>The application shall allow the administrator to upload the winner's photo.</li> </ul>	Pass

Table 10 shows the output of functional testing for view result module.

**Table 10: Functional Testing for View Result Module**

Test ID	Requirement ID	Explanation	Result (Pass/Fail)
TC06-01	FR06-01	<ul style="list-style-type: none"> <li>The application shall display the photos of new highest council.</li> </ul>	Pass

Table 11 below shows the output of functional testing for generate report module.

**Table 11: Functional Testing for Generate Report Module**

Test ID	Requirement ID	Explanation	Result (Pass/Fail)
TC07-01	FR07-01	<ul style="list-style-type: none"> <li>The application shall display the photos of new highest council.</li> </ul>	Pass

### 3.6 Maintenance Phase

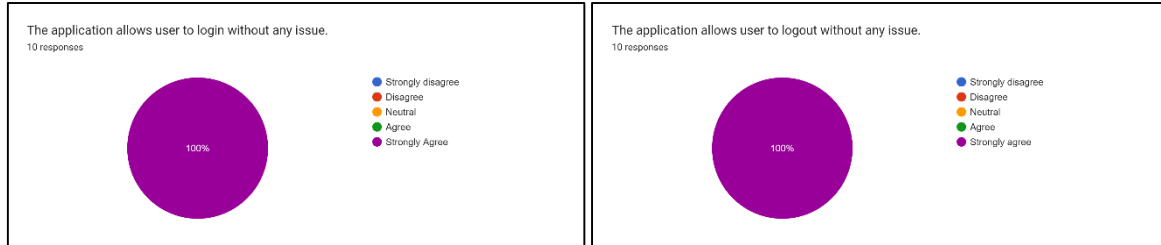
Several tasks are carried out to guarantee the software system's efficient operation and lifespan during the maintenance phase. First, bug fixing becomes a top priority since any problems or flaws found while the system is in use are immediately addressed and fixed. To improve the system's effectiveness and responsiveness, performance optimization measures are also made. To protect the software against any vulnerabilities, security updates and patches are applied on a regular basis. To enhance the user experience and adapt to changing requirements, user input and recommendations are carefully reviewed before being adopted. Additionally, it is essential to regularly assess the system's performance, examine logs and analytics, and spot potential improvement areas. Lastly, complete documentation of all modifications conducted during the maintenance phase is required to make future maintenance tasks easier.

## 4. Result and Discussion

The performance and user experience of the UTHM SUKSIS Fingerprint Voting System are analyzed in this section. User feedback indicated overall satisfaction with the system's ease of use and reliability. The findings obtained through User Acceptance Testing provide valuable insights for future enhancements, including recommendations for improving system performance and addressing user concerns.

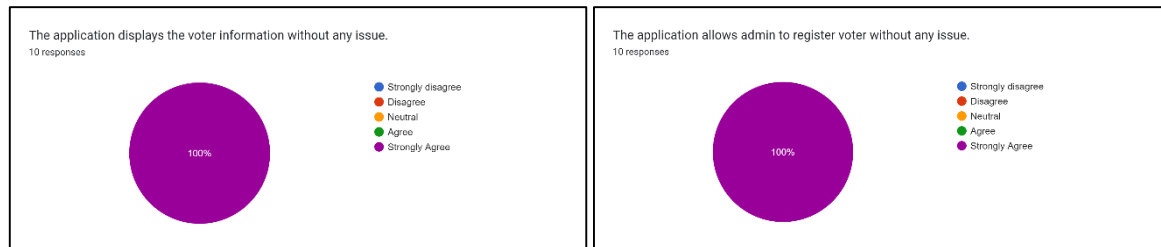
#### 4.1 User Acceptance Testing (UAT)

User Acceptance Testing is conducted to obtain feedback from end users to ensure that the developed system meets their needs and expectations. Through User Acceptance Testing, we can validate the functionality, usability, and performance of the system. There are 10 responders which UTHM SUKISIS are trainees and trainers who gives their feedback via Google form.



**Figure 19: User Acceptance Testing for Login Module**

Figure 19 above shows the evaluation on login module as 10 responders are strongly agreeing the application allows user to login and logout without any issue.



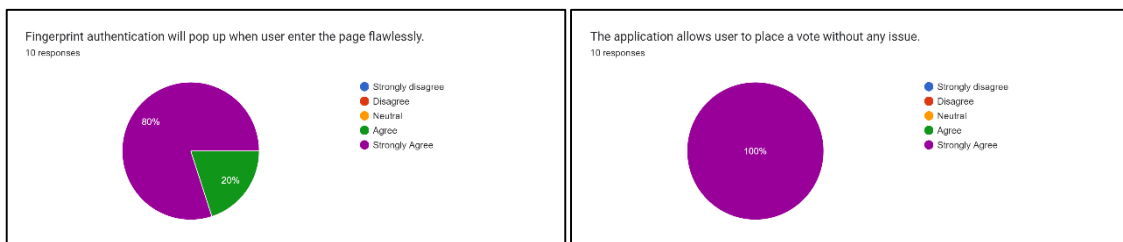
**Figure 20: User Acceptance Testing for Manage Voter Information Module**

Figure 20 above shows the evaluation on manage voter information module as 10 responders are strongly agreeing the application displays the voter information and allows the administrator to register voter without any issue.



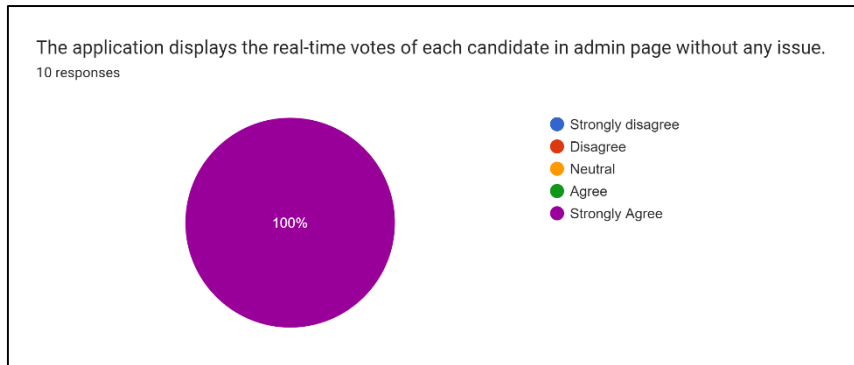
**Figure 21: User Acceptance Testing for Manage Candidate Information Module**

Figure 21 above shows the evaluation on manage candidate information module as all 10 responders vote for strongly agree as both functionality and usability for this module works flawlessly.



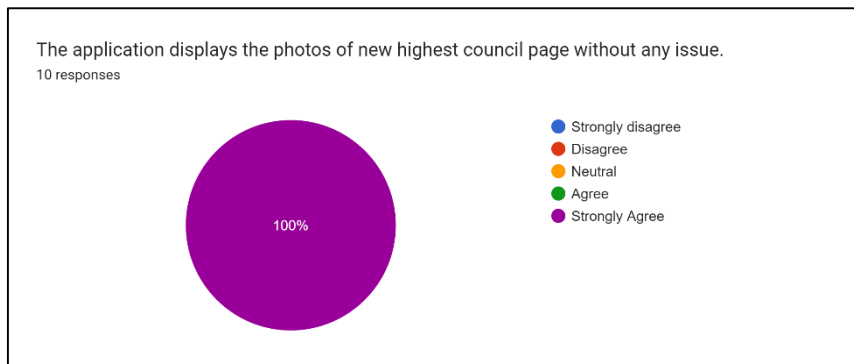
**Figure 22: User Acceptance Testing for Manage Election Module**

Figure 22 above shows the evaluation on manage election module. The results for the fingerprint authentication pop up flawlessly when user enter the page are out of ten responders, one responder voted for agree. Furthermore, the application allows users to place their votes without any issue, as all respondents voted strongly agree.



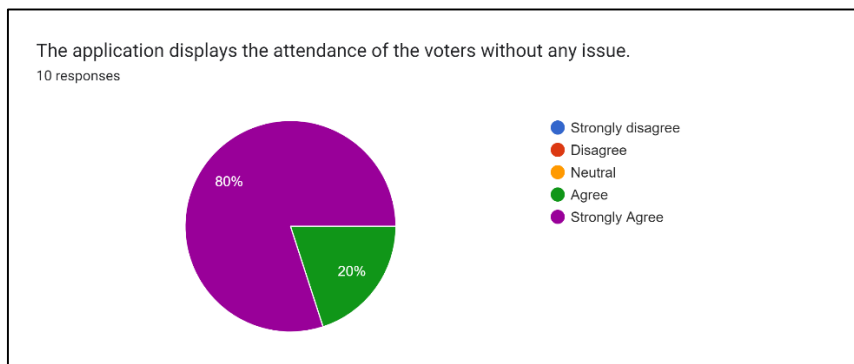
**Figure 23: User Acceptance Testing for Calculate Result Module**

Figure 23 above shows the evaluation on calculate result module as all ten responders strongly agree that the application displays the real-time votes of each candidate in admin page without any issue.



**Figure 24: User Acceptance Testing for View Result Module**

Figure 24 above shows the evaluation on view result module. Ten out of ten responders vote for strongly agree as the application displays the photos of new highest council without any issue.



**Figure 25: User Acceptance Testing for Generate Report Module**

Figure 25 above shows the evaluation on generate report module as all responders strongly agree that the application displays the attendance of the voters without any issue.

## 5. Conclusion

In conclusion, the UTHM SUKSIS Fingerprint Voting System is an essential application developed to address the limitations of the existing election process at SUKSIS, Universiti Tun Hussein Onn Malaysia (UTHM). The system aims to increase the effectiveness and data accuracy of the voting process, ultimately constituting a new highest council for SUKSIS trainees. The Waterfall Process Model was adopted as the methodology for system development, ensuring a systematic and well-organized approach. Through thorough requirement analysis, system design, implementation, testing, and maintenance phases, the system was meticulously developed. The system's features include login functionality, management of voter and candidate information, election management, result calculation and viewing, report generation, and authentication through fingerprints. The system design encompasses an overall architecture, database schema, and user interface design. Extensive testing, including user acceptance testing, was conducted to ensure the system's usability and functionality. In the maintenance phase, bug fixing, performance optimization, security updates, and user feedback incorporation were prioritized. Overall, the UTHM SUKSIS Fingerprint Voting System provides a transparent and reliable platform for the election process, offering future potential for enhancements and improvements.

## Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support

## References

- [1] Y. Faridah and H. N., "Fingerprint Biometric Systems," Trends in Bioinformatics, pp. 52-58, 2016.
- [2] M. Martin, "What is Waterfall Model in SDLC? Advantages and Disadvantages," GURU99, 2022.
- [3] K. Pohl, "Requirements Engineering: Fundamentals, Principles, and Techniques 2010th Edition," 2010.
- [4] W. V. Casteren, "The Waterfall Model and the Agile Methodologies: A comparison by project characteristics - short," ResearchGate, 2017.
- [5] W. W. Royce, "Managing the development of large software systems: Concepts and techniques," in Proceedings of the 9th International Conference on Software Engineering, pp. 328-338, 1970.
- [6] B. W. Boehm, "A spiral model of software development and enhancement," Computer, vol. 21, no. 5, pp. 61-72, 1988.
- [7] I. Sommerville, "Software Engineering," Pearson Education Limited, 2016.