THOMPSON RIVERS UNIVERSITY

SENG 3210– Applied Software Engineering

VoxChoice

Remote Polling Application

Damien Hodsall (TRU ID)

Keishi Miya (TRU ID)

Kyle McKee (TRU ID)

Date submitted

Table of Contents

- The table of contents should be automatically generated by selecting “References/ Table of Contents”. Remember that the table of contents should not have an entry of the “Table of Contents” itself.
- Proofread the text for typing and grammar mistakes.
- Follow the IEEE Bibliography style for the references by selecting "References/ Citations & Bibliography/ Style".

List of Figures

List of Tables

1 Introduction

Technology has transformed the way we communicate and connect in the digital age. The convenience and accessibility of virtual environments have made them the status quo, and thus technology is inherently expected to transcend physical boundaries. To empower users with the means to express their opinions from anywhere, they require a platform from which teachers, administrators, and the like can facilitate remote discussions.

The VoxChoice Android app seeks to solve this problem. In the forthcoming sections of this report, we will discuss the rationale behind VoxChoice, elucidate its design requirements, explore potential solutions, and ultimately present the chosen design with a thorough analysis of its features, constraints, and overall impacts.

2 Design Problem

2.1 Problem Definition

The classroom can be perceived as a societal controversy; where some are welcome to the collective environment, others may find it hostile or judgmental. Such issues are most prominent in group discussions and can lead insecure students to shy away or give in to the biases of the majority. In disjointed classrooms, personal interactions can ultimately distort the results of a discussion.

The VoxChoice Android app lets teachers create virtual surveys for their students using a remote polling system; however, it can be used by anyone. The remote nature of the polling system will allow users to create polls, respond to polls, and view the results of existing polls from anywhere. The results of polls can be sent to any group that the user chooses, and the results are anonymous to protect voter privacy. Using remote surveys, VoxChoice promotes accessibility and inclusivity while protecting users' discretion.

2.2 Design Requirements

2.2.1 Functions and Objectives

Functional Requirements

- The system must allow users to create or log into their Google account.
- The system must not disclose any personal information (password or contacts).
- The user shall be able to create polls with dynamic question layouts.
- The user shall be able to send their polls to any other grouping of users.
- The user shall be able to vote on polls created by other users.
- The user shall be able to make changes to their answers before submission.
- The user shall be able to edit or delete their existing polls.
- The system must keep users' votes anonymous.
- The system shall provide users read access to a dashboard with all ongoing and expired polls.
- The system shall allow users to export poll results in PDF format.
- The system must be accessible for users remotely via Android OS.

Objectives

- The system should be ***accessible*** to all types of users from anywhere.
- The system should be ***understandable*** to any experience level of user.
- The system should be ***responsive*** to the users' inputs promptly.
- The system should be ***secure*** from data leaks and file corruption.
- The system should be ***well-documented*** for the sake of maintainability and litigation.

2.2.2 Non-functional requirements and constraints

Non-Functional Requirements

- Maintainability: The application should be easy to maintain and evolve using engineering procedures and quality assurance practices.
- Modifiability: The system architecture and documentation will enhance the application's adaptability to changes at any point of the development cycle.
- Scalability: The number of voters can vary greatly based on the demand of a poll's creator.
- Portability: The application will be developed with a lower-level Android API to be compatible with a broader range of Android devices.
- Security: The application validates the identity of users given their Google accounts; users' information or voting data must be kept confidential.
- Usability: The application will boast a simple GUI for effective user interactions.

Constraints

- The system must adhere to the laws and regulations of its state of operation (Canada/British Columbia).
- The system must prioritize the safety and security of the public and the environment.
- The system must be completed by March 25, 2024.
- The system must be created without a budget.
- The system must be compatible with Android OS.
- The user must possess a valid Google account to access the system.
-

3 Solution

In this section, you will detail various solutions generated during your team's brainstorming sessions for project implementation. Not all solutions may encompass all desired features, and some may not fully meet the constraints. These solutions emerge as your team explores ways to implement all features within the specified constraints. Ultimately, you'll choose a solution that, in your assessment, incorporates all necessary features and adheres to the constraints. It's crucial to bear in mind that the process of engineering design is iterative!

3.1 Solution 1

Write a brief description of your first solution and provide the reasons for not selecting this one.

You can use the component diagram, sequence diagram, and class diagram.

3.2 Solution 2

This solution is an improved solution but might not be the final solution that you select. Give a brief description of this solution here.

You can use the component diagram, sequence diagram, and class diagram.

3.3 Final Solution

This solution is the final solution. Explain why it is better than other solutions. You may use a table for comparison purposes. After providing the reason for selecting this solution, detail it below.

3.3.1 Features and the software architecture

Discuss all the features of your final solution. Describe the functionalities of the top-level components and how they will be used for enabling those features. The product features may be tabulated (with a title) for improved comprehension. Use component diagrams to model the internal structures (i.e., sub-components or second-level components) of two major components. Describe the functionalities of the sub-components and the interactions (e.g., the interfaces) between the sub-components. Explain the interfaces between the top-level architecture and the internal structures (i.e., explaining how the internal structures interact with other top-level components).

3.3.2 The system interfaces

Describe the temporal events (i.e., the time-triggered events) and the signal events (i.e., events received from external components) for the intended application. Describe the expected response of the system to each event.

3.3.3 The user interface design

Design the user interface components. Describe the user interface components, the possible business events, and the responses to the triggered events.

3.3.4 The requirements traceability matrix

List the system's requirements and map the requirements to the corresponding design component, code component (e.g., java class file or XML configuration file), and the required testing scenario.

3.3.5 Environmental, Societal, Safety, and Economic Considerations

Explain how your design project considered environmental, societal, and economic considerations. It may include how your implementation has positive contributions to the environment and society. What type of financial decisions did you make? How did you make sure that the implementation is safe to use?

3.3.5.1 Environmental considerations

Explain how your design project considered environmental considerations.

3.3.5.2 Societal considerations

Explain how your design project considered societal considerations.

3.3.5.3 Safety considerations

Explain how your design project considered safety considerations.

3.3.5.4 Economic considerations

Explain how your design project considered economic considerations.

3.3.6 Limitations

Every product has some limitations, so is the case with your design solution. Highlight some of the limitations of your implementation here.

4 Teamwork

Since this is a group project, you must have a fair distribution of tasks among yourselves. To this end, you must hold meetings to discuss the distribution of tasks and keep track of the project progress.

4.1 Meeting 1

Time: January 26, 2024, 10:00 am to 11:00 am

Agenda: Distribution of Project Tasks

Team Member	Previous Task	Completion State	Next Task
Team member 1 **Team member 2** **Team member 3** **Team member 4**	N/A N/A N/A N/A	N/A N/A N/A N/A	Task 1 Task 2 Task 3 Task 4

4.2 Meeting 2

Time: February 4, 2024, 10:00 am to 11:00 am

Agenda: Review of Individual Progress

Team Member	Previous Task	Completion State	Next Task
Team member 1 **Team member 2** **Team member 3** **Team member 4**	Task 1 Task 2 Task 3 Task 4	80% 50% 100% 75%	Task 1, Task 5 Task 2 Task 6 Task 4, Task 7

4.3 Meeting 3

Provide a similar description here.

4.4 Meeting 4

Provide a similar description here.

5 Conclusion and Future Work

- Provide a concise summary of your accomplishments, outlining the design functions and objectives successfully achieved while adhering to the specified constraints.
- In consideration of the limitations inherent in the application design, offer recommendations for potential enhancements in future iterations of the design.

6 References

- Use the IEEE reference style.
- Do not put any reference if it is not cited in the text.

7 Appendix

If you want to provide any additional information, use this appendix.