THOMPSON RIVERS UNIVERSITY

SENG 3210– Applied Software Engineering

VoxChoice

Eric Jacob (T00691963)

Felix Beauchemin-Berthelot (T00684599)

2024-03-04

Table of Contents

1 Introduction.. 6
2 Design Problem.. 7
2.1 Problem Definition.. 7
2.2 Design Requirements.. 7
2.2.1 Functions... 7
2.2.2 Objectives.. 7
2.2.3 Constraints.. 7
3 Solution.. 8
3.1 Solution 1.. 8
3.2 Solution 2.. 8
3.3 Solution 3.. 8
3.4 Final Solution.. 8
3.4.1 Features.. 8
3.4.2 Environmental, Societal, Safety, and Economic Considerations...................... 9
3.4.3 Limitations.. 9
4 Team Work... 9
4.1 Meeting 1.. 9
4.2 Meeting 2.. 9
4.3 Meeting 3.. 10
4.4 Meeting 4.. 10
5 Project Management.. 11
6 Conclusion and Future Work.. 12
7 Self Reflection.. 12
8 References.. 13
9 Appendix... 14

1 Introduction

The VoxChoice project has the goal of making voting and decision making within a group more practical. The project is based on an online polling system that will allow the users to express their opinions remotely and privately on a variety of different topics. Furthermore the system will allow users to create polls which they can edit the enter, edit prompts and invite people to participate in the survey.

Background Information:

There is a strong need for the ability to gather opinions and reach a democratic consensus in educational settings (SENG 3210 classroom), but also in other settings like corporate boardrooms, family, and social circles. Currently, existing polling services like Google Forms and SurveyMonkey are dedicated to a browser-based format, and there is a need for a mobile application version to accommodate the large mobile user base. VoxChoice is envisioned to provide an alternative for Android that allows for more than just the *Study Tour Decision testcase* but would also extend to many more applications.

Rationale:

The rationale behind VoxChoice originates from the gap in polling applications for the mobile user base. The project seeks to develop a solution tailored to the Android platform. VoxChoice aims to improve voting or polling-based decision-making by offering a convenient and accessible platform designed for users to express their opinions through participation in the voting process. The envisioned solution aims to allow users to participate as both an administrator who could create polls, prompts, invite users, and see results, but also as a user to participate in surveys. This is meant to be used beyond the educational setting and is envisioned to accommodate a larger domain of possibilities.

Overview of Following Sections:

The following sections of this document will delve into the different aspects of the development and documentation of the VoxChoice project.

1) Project Overview: Provides detailed description of the project scope, objectives and the requirements that were considered in the making of the project.
2) Project Learning Outcomes: Describes the skills that were acquired and developed through the creation, designing, developing and implementing the project.
3) Project Requirements: Provides details of the key features and functionalities of the solution, along with non-functional requirements and constraints.

4) Summary of Deliverables: Provides details for the expected project outcomes, including design, implementation, testing, documentation, project management, presentation, and the project poster.

2 Design Problem

2.1 Problem Definition

The problem at hand originates from the necessity of a decision making mechanism within our group in SENG 3210 with many diverse opinions. With the upcoming summer study tour, reaching a consensus can be challenging. The proposed solution, VoxChoice, aims to facilitate a democratic and inclusive decision-making process by enabling remote voting on various topics, starting with the selection of the study tour destination. Moreover, the problem extends beyond simple decision-making. The project also addresses the need for a system that can be easily used by individuals with limited access. VoxChoice aims to transform the way decisions are made within groups making the process more inclusive, efficient, and secure. The ultimate goal is to create a platform that not only serves the immediate needs of the SENG 3210 class but also sets a precedent for future collaborative decision making scenarios. This solution seeks to bridge the gap between different opinions, providing a platform for every user's voice to be heard and considered equally, thus enhancing the decision making process within educational and potentially other organizational settings.

2.2 Design Requirements

2.2.1 Functions

- User registration and authentication to the users to ensure security and uniqueness of the profiles.
- Secure login with usernames and passwords through a built in authentication system
- remote access on android platform
- Administrators have the ability to create new polls and edit existing polls
- Provides users with the ability to vote in the topic of their choice
- Real-time dashboard displaying comprehensive statistics on active polls.

2.2.2 Objectives

- Ensure quick response time and communication time when using the app
- Promote equity, diversity, and inclusion within the platform
- enable users to access the platform remotely, allowing them to participate from any location
- incorporate user feedback constantly and improve platform features accordingly
- The platform must be user-friendly and accessible, ensuring a seamless experience for all users, regardless of their proficiency.
- Secure data is protected and users have the right to have their privacy maintained
- The platform must encourage people to vote through an engaging environment

- Must be aesthetically pleasing with ease of use in mind when developing UI
- Must be reliable with little crashes
- provide administrators with the ability to manage discussion topics effectively
- administrators should be able to easily navigate through different polls, manage participant responses.

2.2.3 Constraints

- Must be developed by March 25th 2024
- Limited to Android Studio
- Regulatory laws of Canada related to voting platforms in Canada.
- Funding and resources to maintain and improve the platform
- Software exportation policies and regulatory constraints

3 Solution

3.1 Solution 1

For our first solution, the team decided that trying to make the initial prototype as simple as possible was preferable. We considered using only local storage for storing the application's data. Furthermore, we did manual variable testing which was time consuming and left us not being able to be certain of the ability of our software to produce the correct results consistently.

The user interface was simple, it was split into 3 pages for the users and 4 pages for the administrators. Firstly, there was a simple login page with a username textbox and a password textbox as well as an extra embedded page to create the account. That page had text fields for their email, username, name, password. We ensured that all fields such as email and username could only be used once per user, and no duplicate accounts could be created. Second page allowed users to see the polls they are currently active for and make their decisions. Third page was a searching tool to allow the user to search potential polls. The fourth page for the administrator was the poll creation page, it had fields to name the poll and create questions and potential answers.

This solution was inadequate because it did not permit for the communication between devices and the implementation for a real-time database as mentioned in the requirements for the project. Furthermore, this solution lacked the automated testing methods using Espresso to test the stability and reliability of our application as well as not using Gradle for automated testing, which leaves us not confident about the reliability of our application.

3.2 Solution 2

Building upon the second solution, the team decided to implement the real-time database through the implementation of Firebase databases. This improved the limitations seen with local storage. In this solution there still was very limited testing through the manual testing methods in Android studio and there was no use of the testing tools Gradle and Espresso.

The user interface was improved, it was still split into 3 pages for the users and 4 pages for the administrators. Firstly, we added a terms of service page to the previously mentioned simple login page. The login page still had a username textbox and a password textbox as well as an extra embedded page to create the account. That page had text fields for their email, username, name, password. We ensured that all fields such as email and username could only be used once per user, and no duplicate accounts could be created. Second page allowed users to see the polls they are currently active for and make their decisions. We improved the visual appeal of this page by adding pictures and changing the background color. Third page was a searching tool to allow the user to search potential polls. The fourth page for the administrator was the poll creation page, it had fields to name the poll and create questions and potential answers. All the other pages were improved by changing their background and improving the visual appeal of the application.

Similar to Solution 1, this version of the software was not using the optimal tools for testing the design and although it matched the requirements of using the real-time database, it did not fulfill the expectations for rigorous testing.

3.3 Solution 3

Solution 3 incorporated the design of solution 2 and integrated various testing methods notably with Espresso and Gradle. The solution used Espresso for testing the user-interface (UI) covering the main flow of the use cases which covers testing user authentication, voting process, analytics checking for administrators and error handling.

Using the Gradle's test suite, we could automate the tests of Espresso for testing data accuracy and ensuring proper behaviour of our system. This version ensured to test the components of the UI and the performance of our application which ensured the proper functionality of our design.

3.4 Final Solution

Table I Decision matrix chart for the considered alternatives

		Solutions					
		Solution 1		**Solution 2**		**Final Solution**	
Criteria	**Weight**	**Score**	**Partial Score**	**Score**	**Partial Score**	**Score**	**Partial Score**
Usability	0.20	5/10	0.10	8/10	0.160	8/10	0.160
Functionality	0.40	4/10	0.160	8/10	0.320	8/10	0.320
Reliability	0.30	4/10	0.120	5/10	0.150	9/10	0.270
Visual appeal	0.10	4/10	0.040	6/10	0.060	8/10	0.080
Sum	1.00		0.420		0.690		0.830

For the final solution we selected Solution 3 due to its highest performance on the Decision Matrix (Table 1).

Solution 3, implemented the improvements done to the UI from the Solution 2, incorporated the real-time database (Firebase) and implemented automated testing through the integrating the Espresso in the Gradle's automated testing suite. Furthermore, more attention was put into the user experience and the usability of the application. This resulted in solution 3 having a strong performance in all categories in the decision matrix.

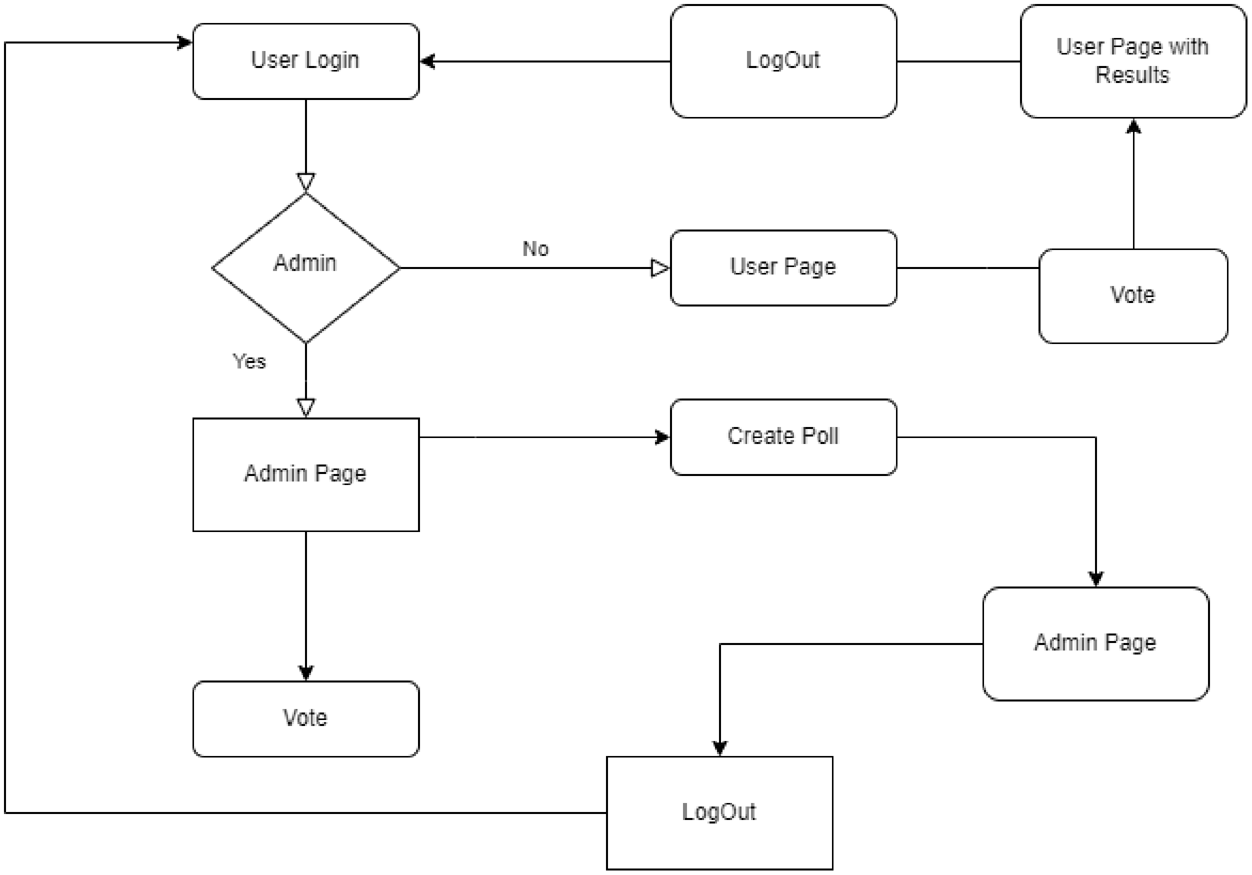
3.4.1 Features And Software Architecture:

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).

Features of the final solution:

- User registration page
- User login page
- Creation of polls for administrators
- Voting for polls for the users

Component Diagram for the Login Page:



•

3.4.2 The system interfaces

Temporal Events:

Periodic Data Refresh: The system checks for any new updates to polls from the Firebase database.

Response: The Application retrieves updates from new polls from the database and refreshes the UI elements to display the new survey every 10 seconds.

Signal Events:

User Login/Registration: A user tries to login to the application.

Response: The Login/Registration component checks the database to confirm that the user actually exists, and also validates the account through checking the password of the registration. This will either result in a failed registration in case of invalid information, or it will allow the user to login.

User Votes On Polls: The user interacts with the UI and votes on one of the polls.

Response: The Poll component will update the data in the database corresponding to the poll based on the vote from the user.

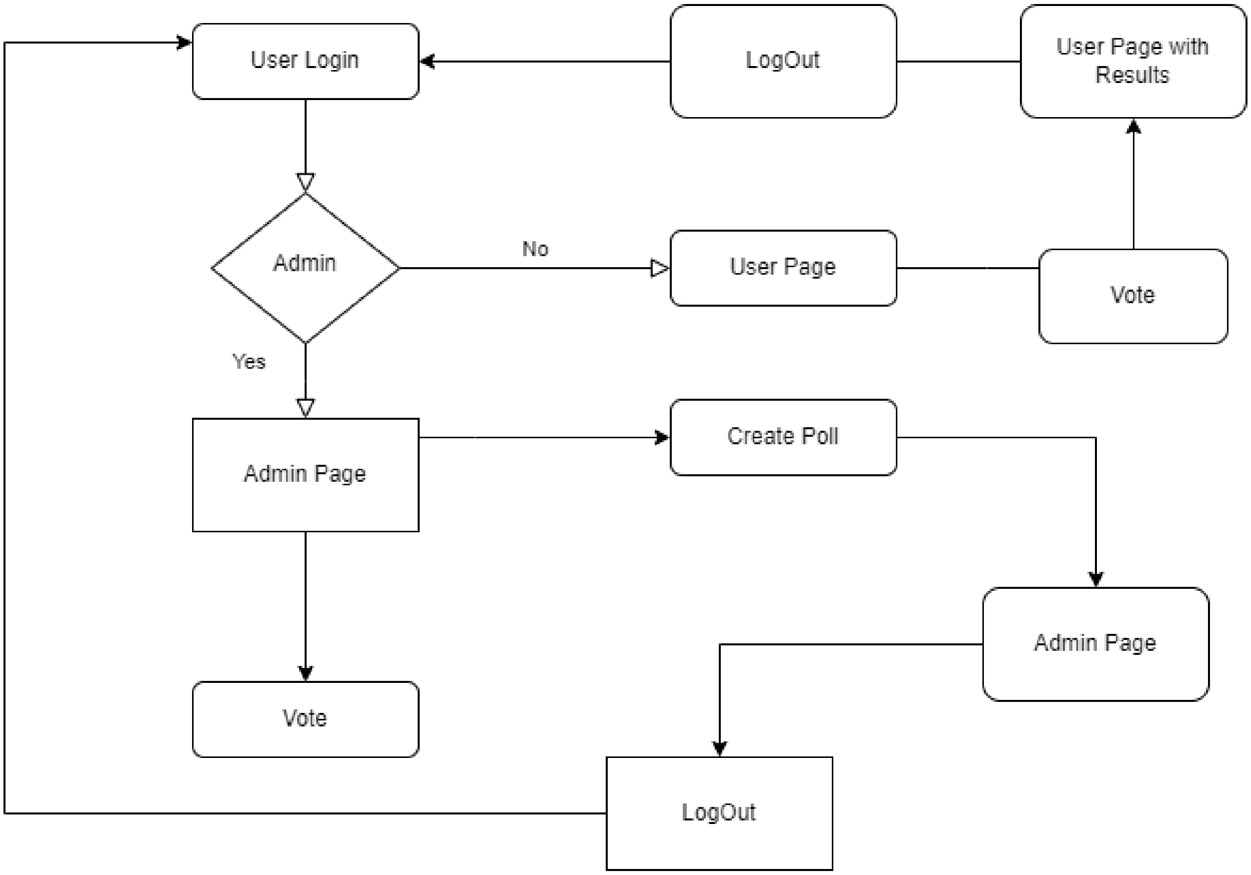
Admin Creates Polls: The admin creates a poll by inputting a question and the various different answers possible.

Response: The poll component will update the poll database with the new poll and with the specified fields imputed by the admin.

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. Project Title SENG 3120 8

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.

CreatePollActivity: Logic to let admins create the poll
Login: Logic for users to login
Poll: Data structure
PollAdapter: Logic to communicate with the Firebase databases
SignUp: Logic for



•

3.4.2 Environmental, Societal, Safety, and Economic Considerations

Our engineering design project for the voting app takes into account several environmental, societal, safety, and economic considerations. To make sure that our project does not only fulfill the requirements but also contributes positively to the world around us.

Environmental Considerations:

Our design tries to minimize the use of physical resources like dedicated servers and computing resources by minimizing the data storage requirements and using cloud-based solutions (Firebase real-time database). Furthermore, by using cloud storage we can reduce the amount of on-device storage which can contribute to minimizing the energy consumption of the devices. In addition, using cloud based services reduces the need for infrastructure for the typical server infrastructure resulting in a diminished energy consumption.

Societal Considerations:

During the conception of our design, we made sure to prioritize equity, diversity, and inclusion (EDI). Some of the considerations that were considered in the creation and brain-storming of the project:

Equity: Our design aims to promote equity by ensuring that all people have an equal opportunity to access and use the various functionalities of our application regardless of socio-economic status, location, or other factors. We prioritize all users to participate with our application and for further development we aim to integrate more language options to remove the language barriers between users and improve the equality of opportunity.

Diversity: In the development team, we embraced diversity by including members with very different backgrounds and experiences. Through this we could use our different vantage points to develop the application to ensure that it is inclusive to our different communities.

Inclusion: Throughout the project, inclusion of a varied user base was a priority to our group. We plan on making this application accessible to earlier versions of Android, increasing the diversity of the user base and including poorer demographics. As well as implementing language options in subsequent releases to increase our inclusivity regardless of the person's background or abilities.

Safety Considerations:

Privacy and Data Protection: We prioritized the safety of our users by doing everything in our power to upkeep their privacy and protect their data. This involves making sure that only the required fields are taken from the real-time database, and minimizing gathering compromising information about our users during the signup stage. Furthermore we abided by the laws and regulations in our province for standards for data transmission.

Error Handling: The app has error handling methods through exception handling to manage and react in case of unexpected situations and to prevent security vulnerabilities, such as validating the users input information during the signup stage and during the voting polls.

Security Audits: The app will be continuously tested through its release and further development to ensure that the app behaves properly and upkeeps the safety and privacy of all users in all cases.

Economic Considerations:

Cost-Effectiveness: Our design will adopt Firebase's pay-as-you-go pricing model, which will allow for scalability of our project through the release and will permit us to deliver the prototype early without a significant upfront investment. In case of needing to scale further to much bigger audiences, other cloud database services will be considered.

3.4.3 Limitations

The design project seeks to fulfill several requirements but it’s important to understand its limitations that might affect the functionality, implementation or deployment of the application. Here are some of the limitations of the prototype:

Internet Dependency: The voting app relies on an Internet connection for data transmission and communication with the cloud database. This means that users without access to a stable internet connection will be limited in their ability to use the application. This becomes a liability in case of network outages.

Device Compatibility: Our prototype is constructed for Android Version 12.0 and later. Older and less powerful devices may not have the ability to use the application. Outdated devices or outdated operating systems may result in performance issues or the inability to use the application.

Scalability: Our prototype will be using Firebase’s pay-as-you-go system for cloud databases. This results in scalability issues due to potentially high prices if traffic becomes high with our application. Furthermore, our ability to scale is entirely dependent on Firebase’s ability to accommodate our service.

Security Considerations: Although our prototype was thoroughly tested, there may still be existing vulnerabilities and security risks. Again, we are also dependent on Firebase which means that any vulnerability they encounter, we will also be affected.

Regulatory Compliance: Our prototype during commercialization will have to abide by several provincial and federal regulations everywhere it is deployed. This means that during commercialisation it will be crucial to contact experts on the regulations prior to launching the product in different jurisdictions.

4 Team Work

4.1 Meeting 1

Time: Feb 15, 2023, 5:00 to 8:00 pm

Agenda: Introduction, Design

Team Member	Previous Task	Completion State	Next Task
Eric Jacob	N/A	100%	UI
Felix Beauchemin	N/A	40%	Design

4.2 Meeting 2

Time: Feb 25, 2023, 10:00am to 2:00 pm

Agenda: Design, UI, solutions

Team Member	Previous Task	Completion State	Next Task
Eric Jacob	UI	80%	UI, running application, Solutions
Felix Beauchemin	Solutions	100%	Team Work, Project management, running application, Solutions

4.3 Meeting 3

Time: March 10, 2023, 12:00pm to 5:00 pm

Agenda: UI, running application, Solutions, Team Work, Project Management, running application, Solutions

Team Member	Previous Task	Completion State	Next Task
Eric Jacob	UI, running application, solutions	80%	Final Testings, make the app visually aesthetic
Felix Beauchemin	Team Work, Project management, running application, Solutions	50%	Conclusions

4.4 Meeting 4

Time: March 17, 2023, 9:00am to 1:00 pm

Agenda: Final Testings, make the app visually aesthetic, conclusions

Team Member	Previous Task	Completion State	Next Task
Eric Jacob	UI, running application, solutions, Final Testings, Aesthetics	80%	Presentation, Final Test and demo runs with debugging
Felix Beauchemin	Team Work, Project management, running application, Solutions, conclusions	50%	Presentation, Future Work

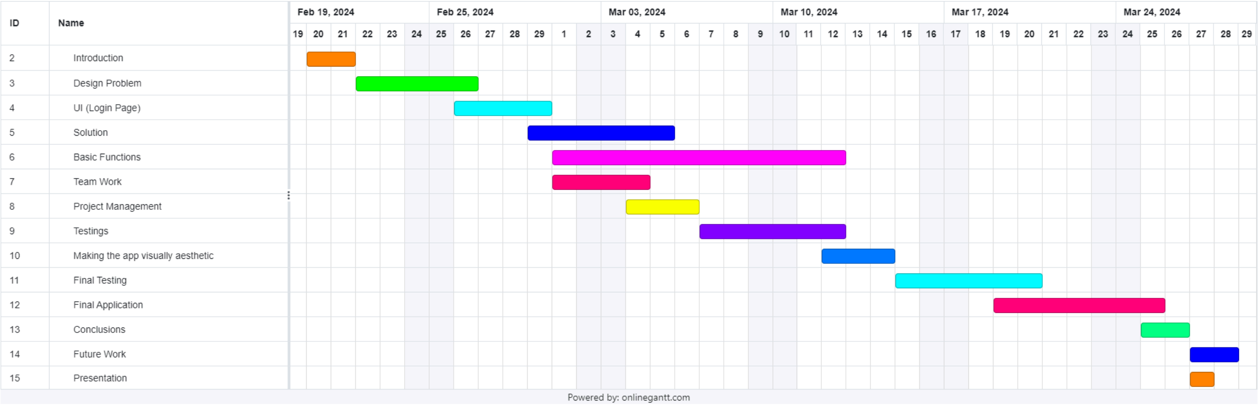
4.5 Meeting 5

Time: March 24, 2023, 9:00pm to 5:00 am

Agenda: Presentation, Future Work, Final Tests and demo runs

Team Member	Previous Task	Completion State	Next Task
Eric Jacob	UI, running application, Presentation, Final Test and demo runs with debugging	100%	N/A
Felix Beauchemin	running application, , Presentation, Future Work	100%	N/A

5 Project Management



ID	Name
2	Introduction
3	Design Problem
4	UI (Login Page)
5	Solution
6	Basic Functions
7	Team Work
8	Project Management
9	Testings
10	Making the app visually aesthetic
11	Final Testing
12	Final Application
13	Conclusions
14	Future Work
15	Presentation

The main tasks are coding for the basic functions of the application, merging the UI with the database through firebase, the final report and the presentation. We started each task early to leave plenty of time to complete each job before the scheduled dates. Also, each task was given a buffer for a few days of slack time for each duty depending on the reasons as it is very hard to predict exactly how long a task will take. The critical path is:

- Introduction
- Design Problem
- UI
- Solutions
- Basic Functions
- Team Work
- Project Management
- Testing
- Making the app visually aesthetic
- Final Testing
- Final Application
- conclusions
- Future Work
- Presentation Prep

6 Conclusion and Future Work

In conclusion, throughout this software engineering design project we had to overcome several hurdles to get to the final solution. Firstly we had to analyze the prompted assignment and decipher the various requirements of our final solution. Then requirement specifications were made and various solutions were considered. This reinforced our learning previously acquired in the Software Engineering Requirements and Specifications course. Furthermore, we had to learn how to develop Android applications using Android Studio and how to apply its various embedded tools like Firebase, Graddle and Espresso. As well we learned how to use Git (Github) for version control, and to share code through an online platform for enhanced collaboration.

Our project fulfilled the following design functions and objectives while meeting the design constraints listed in Section 2.2.3:

- User registration and authentication to the users to ensure security and uniqueness of the profiles.
- Secure login with usernames and passwords through a built in authentication system
- remote access on android platform
- Administrators have the ability to create new polls and edit existing polls
- Provides users with the ability to vote in the topic of their choice

In the future, the following improvements could be done to the design:

- Further testing could be done to prevent the application from crashing during its utilization.
- User interface could be reworked to make it more visually appealing.
- Features could be added to accommodate a large audience, not only students and teachers.
- Dynamic number of options could be displayed during the poll creations.
- Different database schemes could be used for more efficient storing of data. This could be achieved by not using Firebase and using a conventional database.

Overall, the project was a strong introduction to Android development and Java and the project could still be extended by making a way for any user to be able to make polls and have an access code for allowing some specific users to access specific polls.

7 Self Reflection

Felix:

What did I did?

Throughout the process of completing this project, I did several tasks:

- Writing several parts of the technical project report like: Introduction, Section 3, Conclusion and this self reflection.
- I collaborated with my partner (Eric) to make the application, especially integrating the Firebase into the project and different basic tasks of the application.
- I collaborated with Eric to make the project presentation.
- I created the brainstorming document for the first submission.
- I encouraged the group to follow the themes taught in class and to ensure proper management of the project development.

What I enjoyed?

This project was much more of a burden than anything else. I guess the minor enjoyment that I had was succeeding in the proper user login and registering page which I think was well done.

What I found difficult?

I found programming to be the most difficult part in this project. I found it very difficult to debug the code and understand why some seemingly ambiguous bugs were randomly making our app crash.

What really worked?

I think that our group's collaboration really worked, I think that Eric and I put forward our best efforts throughout the project and we did our best.

Next time, I think that several aspects could be improved upon like having more technical experience with the Android Studio Developer and be better equipped with the Java fundamentals for developing for Android.

Eric:

What did I did?

Throughout the process of completing this project, I did several tasks:

- Writing several parts of the technical project report like: Design Problem, Team Management, Project Management, Conclusion and this self reflection.

- Collaborated to create the UI for the login page and Sign in page along with setting up the firebase for user authentication.
- I collaborated to make the project presentation.
- I created the test document.
- I collaborated to create the voting page along with setting up the database.

What I enjoyed?

I enjoyed the challenges we faced, along with the problem solving that was required to complete this project. I had never written any code in Java and was learning everything along the way which made it rewarding to complete each task.

What I found difficult?

I found time management along with coding to be difficult. Our lack of experience in Java along with the limitations of android studios to debug random errorless crashes made it challenging. Also, every other course had a final project due at the same time.

What really worked?

Even though we only had 2 members, I felt it was easy to get work done and decide on the direction to take our project with me and Felix having similar visions for the end results.

Next time, I think having more fundamental experience about Java and time management could help us to work through this project easier.

8 References

[1] https://firebase.google.com/docs/auth

[2] https://firebase.google.com/docs/database

[3] https://www.cs.dartmouth.edu/~sergey/cs65/cheatsheets/Common-Android-Views-Cheat-Sheet.pdf

[4] https://www.scribd.com/document/272173051/Common-Android-Views-Cheat-Sheet