Mobile client for the E-cclesia electronic voting protocol

Sraddheya Gurung

4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh

2023

Abstract

The usability of traditional and modern voting systems have been proven to be below acceptable levels as a result of focusing mainly on security and overlooking the large number of usability challenges involved[49]. E-cclesia[23] is a new voting system that has been proven to be more secure than others currently available, but accordingly, is more complex and has more steps than voting systems the general population will be familiar with. To avoid repeating the same mistakes of some existing systems and creating a voting interface that is very secure but unusable, this paper explored the specific usability challenges of E-cclesia and implemented an application of the voter system that should overcome them.

Through a final evaluation of the implemented interface, the usability of the interface was found to be at least "Good" and trustworthy, but too many failures occurred in the joining stage which needs to be resolved. These results are useful, but was performed on an incomplete version of the voter application, an ideal evaluation would have incorporated the full real E-cclesia system. Overall, this dissertation highlights the importance of considering usability in voting systems, and provides valuable insights into the usability of the E-cclesia voting protocol.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics committee.
Ethics application number: 617552
Date when approval was obtained: 2022-10-20

The participants' information sheet and a consent form are included in the Appendix A and B.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Sraddheya Gurung*)

Acknowledgements

I would like to thank all the participants who took part in the different studies in this paper, especially the Human Computer Interaction experts who took the time to take part even though they were very busy and gave me advice on my dissertation. I would also like to thank my supervisor, Dr. Myrto Arapinis, for the consistent support and feedback. Finally, I would like to thank my friends and family for their support and taking the time to proof read this paper.

Table of Contents

Chapter 1

Introduction

1.1 Motivations

Voting has been integral to democratic societies for centuries [79]. The study of secure electronic voting systems has been a topic of research in the literature for over 30 years, beginning with David Chaum's paper[32]. With the advancement of modern technology, the security of voting systems has continued to develop significantly. However, with these developments, other factors that impact elections have been overlooked, namely usability.

The usability of traditional voting systems has been proven to be below acceptable levels[30], potentially leading to low voter confidence and even inability to vote at all. These issues are concerning and can have great repercussions depending on the stakeholders involved and the impact of the outcome of the election. Electronic voting systems often come with even more usability issues because they are unfamiliar and more complex, yet usability of these systems is still frequently overlooked and consequently at unacceptable levels[36].

The E-cclesia voting protocol is a new voting system developed by Arapinis et al. that has been proven to be more secure than others currently available[23]. Given its strong security measures, it would be ideal for organisations to begin using this voting system, however this will only be possible if users have good experiences with it and it can be proven that the system's usability is not a liability to its security. To achieve this, a comprehensive study into the usability of E-cclesia needs to be performed and also make comparisons to other similar systems. The complete E-cclesia system would comprise of an admin panel for admin users to organise elections and a voter implementation for voters to cast ballots. The main focus of this paper will be the extra step of verification present in voter implementation, as it has been proven to be one of the biggest challenges of more complex voting systems[20][55].

1.2 Research objectives

The main goals of this project in relation to the E-cclesia system are:

1. Gather the functional requirements for the admin panel and voter application.
2. Design and implement the voter application.
3. Evaluate the usability of the implemented voter application.

1.3 Contributions

My main contributions presented in this project are:

1. Identification of the functional requirements of the E-cclesia system and corresponding user flow diagrams.
2. Review of related research and summaries of related design requirements for electronic voting systems.
3. Development and implementation of a suitable design for the E-cclesia system voter application.
4. Designing and conducting a usability study on the voter application with participants from the University of Edinburgh.

1.4 Dissertation structure

*Chapter 2*2 introduces the context and background information for the problem being investigated and explains why this this project is important and necessary. In particular, I will be explaining what E-cclesia is, why it is more secure than existing voting systems and the challenges of implementing it.

*Chapter 3*3 outlines and justifies the methods and procedures used to conduct the research and demonstrates how each stage of the project is a progression from the previous stage.

*Chapter 4*4 outlines the data collection techniques used to identify and document the requirements that helped me design and create the mobile implementation of E-cclesia.

*Chapter 5*5 presents the first feedback round using the initial E-cclesia design and a high fidelity prototype.

*Chapter 6*6 presents the second feedback round using the improved E-cclesia design and a high fidelity prototype.

*Chapter 7*7 outlines and justifies the design decisions for the application and the technology used.

*Chapter 8*8 presents the final evaluation using the final E-cclesia design and the implemented application.

*Chapter 9*9 summarises the findings of the evaluation, limitations of the project and what future work could be done to improve and extend the project.

Chapter 2

Background

2.1 Electronic voting (e-voting)

Electronic voting (e-voting) refers to any voting system in which some electronic means is involved in any stage of the voting process such as recording, casting or counting votes[35]. It has been rapidly replacing traditional voting systems all over the world because of its promise of greater efficiency, enhanced security and greater transparency which makes it more desirable [50] [43]. However, advances in technology have proven that most e-voting systems have failed to meet the minimum security standard for voting [59][63]. E-voting systems can be split into two categories; centralised voting and decentralised voting.

2.1.1 Common roles

The common roles involved in most modern voting systems, both traditional and electronic, are:

- *(Eligible) voter:* Most elections only allow people who satisfy a certain characteristic to cast a vote. For example in national elections, this is usually only people of that nationality above a certain age. An eligible voter is a person who has been added to the electoral roll for the election because they satisfy this characteristic and thus are allowed to cast a vote in the election. Throughout this report, the terms 'voter' and 'eligible voter' will be used interchangeably unless specified as an 'ineligible voter'.
- *Election authority:* The election authority is the body in charge of overseeing the election. This usually involves the setup and adding voters to the electoral roll. Throughout this report, the terms 'Election authority' and 'organisation' will be used interchangeably unless specified otherwise.
- *Tallier:* Once the votes have been cast, they need to be counted/tallied to determine the final result. The tallier checks that the authenticity of the vote and, if it is valid, counts it towards the final result.

2.1.2 Security properties

There are a variation of the security properties that different groups believe should be satisfied by e-voting. The properties this project is concerned with are[23]:

- *Correctness:* Every honestly cast vote will correctly be counted in the final tally, and this final result is the same for all parties involved.
- *Eligibility:* Only eligible voters will be able to vote.
- *Fairness:* No partial results can be learned until everyone has cast a vote.
- *Voter privacy:* Voters' identities cannot be linked to their vote.
- *One voter-one vote:* Only one vote from every eligible voter will be counted in the final tally.
- *Verifiability:* Any party can be certain that all valid votes have been counted in the final tally.

2.1.3 Centralised voting

Centralised voting is the most common type of e-voting currently in place. In this system, voters submit their vote via some electronic means and a single group (talliers) is responsible for collecting and counting all the votes (tallying). The security of this voting system relies on the assumption that talliers will follow all the guidelines and not attempt to breach privacy. In practice, this system is unrealistic and acts more as a central point of failure than a strength[25]. In the case that a subset of talliers has been bribed or dishonest, ineligible votes can be included in the tally set, thus violating correctness and/or eligibility [82][75]. In the case that a subset of talliers collude to find out voters' preferences, the identity of voters can be uncovered, thus violating their privacy. Finally, in the case that voters need to post their votes to a publicly accessible bulletin board, partial results can be leaked under full tallier collusion, thus violating fairness.

Even if the talliers do not breach privacy, other issues can arise. Voters may lack confidence that collusion has not occurred or that their vote has been fairly counted because there is no way for them to verify the votes themselves [16][45]. The talliers can also unknowingly fall victim to human error and miscount and/or have a dispute between themselves [83][17], at which point a third party would not be able to resolve the issue. Specific centralised e-voting systems have additional issues like the Diebold AccuVote-TS 4.3.1 DRE system [59] but the issues that talliers present is common across all centralised systems.

2.1.4 Decentralised voting

In decentralised voting systems, the tally phase can be carried out by any interested parties, including voters. Thus, no single authority is responsible for tallying the votes and any third party can validate the results [57], making this method more secure than centralised voting.

Unfortunately, such attempts at decentralised voting systems currently in existence still suffer from security issues and so do not meet all the security properties listed in Section 2.1.2[56][23], making them impractical for wide spread use. As all voters should be able to tally the votes themselves, the last voter will be able to calculate the current tally before they submit their vote, violating fairness. In the case that the last voter then abstains from voting all together, either to maliciously disrupt the election or because they do not want to complete the process, all active eligible votes cannot be tallied and the election must be aborted. In some cases, attackers can manipulate elections by casting ballots on an eligible voters' behalf or submitting false ballots that prevent the tally from being decrypted [31]. In other systems where specific issues have not been identified, they may lack the analysis to be considered adequately secure.

2.1.5 E-cclesia

E-cclesia [23] is a decentralised voting protocol introduced by researchers at Edinburgh University. By combining the self-tallying elections paradigm [57], special cryptographic tools and blockchain technology, E-cclesia has been shown to overcome the issues faced by other systems that have attempted to achieve the decentralised status and is provably secure. The paper by Arapinis et al. [23] proves that E-cclesia satisfies all the security properties in section 2.1.2 through various methods:

- *Correctness* is achieved using cryptographic tools when generating credentials and by ensuring all the credentials and authenticated ballots are delivered in chronological order.
- *Eligibility* is achieved by using unforgeable signing and verification keys and by ensuring voters only store the credentials of eligible voters.
- *Fairness* is achieved by implementing time lock encryptions for the casting phase. Since no vote can be decrypted before the end of the cast phase, no one can learn the partial results.
- *Voter privacy* is achieved by using an anonymous broadcast channel and zero knowledge properties.
- *One voter-one vote:* is achieved by making sure that multiple votes from the same credentials are not counted.
- *Verifiability:* is achieved using the combination of several cryptography tools proven to be secure.

This voting protocol had two roles; the election authority and the eligible voters. No talliers are required as any party interested can take part in tallying the votes, as in agreement with the definition of decentralisation. The sequence of the protocol is as follows:

1. **Setting parameters:** Through some secure broadcast channel, the election authority will have sent out a link for voters to anonymously register for the election. Those who register will then be added to the electoral roll and become eligible voters. The election authority will also input other parameters needed to create

a valid election. This is all then posted on the bulletin board in the blockchain available for everyone to see.

2. **Credential generation:** Each eligible voter will randomly generate their private credentials for the election using commitments, and posts their signed credentials on the bulletin board on the blockchain.
3. **Cast:** Each eligible voter generates a ballot of their vote which consists of their selected option, their serial number and proof that they are an eligible voter. A zero knowledge proof is used to protect the identity of the voter.
4. **Tally:** Each eligible voter computes the tally that corresponds to the set of ballots they received from the eligible voters. Multiple votes from the same credentials are discarded and tallying can only commence when the tally phase starts.

2.2 Existing systems

Currently, there exists several e-voting systems that satisfy a majority of the security properties but fail to meet the required usability expectations. In this project, I focused mainly on e-voting systems that are self-tallying, that is, any interested party can take part in the tallying phase [57] but cannot be considered completely decentralised as they do not satisfy all the required security properties in section 2.1.2.

- **Helios** is an open source web-based voting system developed by researchers at the University of California, Berkeley [21] [8]. It uses a ballot tracker to allow eligible voters to verify that their ballot was received and tallied properly. It also makes sure to encrypt votes inside the browser before it is sent to the server.
- **Belenios** is a web-based voting system that was originally designed for voting in the public sector in the Benelux region [27]. It is based off of the Helios voting systems and so includes many of the same features, but also supports multiple languages, more complex voting scenarios and larger scale elections. In fact it has been used in a number of high-profile elections in the Benelux region, including the election of the Dutch parliament in 2012 and the Belgian municipal elections in 2018 [CITE].
- **Condorcet Internet Voting Service (CIVS)** [5] is an online voting system that uses the Condorcet method of voting (a ranking voting system), and was developed by the Center for Range Voting, a non-profit organization dedicated to promoting voting methods that better reflect the preferences of voters.
- **Civitas** [34] is a voting system developed by researchers at the University of Surrey that uses blockchain technology and homomorphic encryption to ensure security. It also allows voters to verify that their vote has been correctly recorded using a cryptograhic key.

The e-voting systems listed above are some of the most largely discussed and within the voting research community and most of them have been used in real elections before, however, some other systems that are notable and have been mentioned in this project or related literature includes *Pret-a-voter*, *Scantegrity*, and *DRE*.

2.3 Usability in voting

In democratic societies, voting is an important and fundamental right that gives individuals a say in who they want to represent them. The security of these voting systems is obviously important as it ensures that the election process is fair and trustworthy. However, recently it has been argued that the usability of these voting systems should also be considered at least as serious as security [49][26][41][61].

Though the basic task of voting, selecting an item from a list of options, seems simple, the unique challenges of voting systems can greatly impact voter confidence and affect the results of the election[74]. Usability issues in voting systems can cause overvotes (voting more times than is accepted or allowed), unintentional undervotes (not casting a vote in an election) or voting for the wrong or even opposing option. This is particularly troublesome in close elections where these errors can be difficult to spot and amend. Even if a voter has been able to cast as intended, if their experience with the system was unsatisfactory because it took too long or was confusing, this may make them doubt if their vote was cast as intended or even recorded at all. In turn, they will doubt the outcome of the election and may not vote again. If the voting experience was particularly difficult, a voter may not even finish casting their vote because they have lost motivation.

The most famous consequence of ignoring usability can be seen in the poorly designed ”butterfly ballots” of the Florida 2000 presidential elections which caused major controversy and led to recounts and mass public distrust in the electoral process.[77] [64]. Ultimately, a user needs to trust the voting system they interact with to use it frequently and successfully, and this trust is established through security *and* usability. Therefore, usability should be considered as integral as security in voting systems. This sentiment can be reflected in the inclusion of usability testing in voting systems in the Voluntary Voting System Guidelines (VVSG) created by the United States Election Assistance Commission (EAC)[73].

2.3.1 Usability challenges of voting

- *Large diversity:* The system must be usable for all individuals over a certain age (usually 18). This is challenging as this encompasses a user pool of very different ages, levels of education, socioeconomic statuses, races, gender and abilities. Whereas, most systems are targeted for a specific group of people.[36]
- *No training:* The system must be usable for voters who have not undergone any training as this is usually not provided and in cases where it is, it is unlikely that a large population will have seen or remembered their training.[26]
- *Infrequent task:* Even if the voter is not using the system for the first time, they will likely only use the voting system a few times a year at most and so they will still be novices. Therefore the system cannot rely on users becoming familiar with it.[36]
- *No assistance:* Vote choices can be controversial and should remain private, thus the system should be able to be used without any external assistance.[26]

- *Language:* The terminology of the system should be understood by everyone, even if the language chosen is not the voters' native tongue.[36]
- *Pressure:* The outcomes of elections are often significant and so the situation and setting may cause voters lots of pressure, stress and anxiety which can affect how they interact with the system.[26]
- *Perception:* To have confidence in the outcome of the election, voters must have confidence that their vote was cast and recorded as intended. [42][41]

2.3.2 Usability challenges of the E-cclesia protocol

- *Transparency:* In his study of transparency and trust [58], Kizilcec states that transparency may promote or erode users' trust in a system by changing beliefs about its trustworthiness. Therefore the E-cclesia interface should be transparent about how the protocol works and why it is secure. However, balancing the display of information between providing user confidence in the system and ensuring that they understand the concept can be very challenging. This is especially true where the technology is very complex as in the E-cclesia protocol. Unfortunately, there is another obstacle that even if the explanations included are sufficient enough to give users a positive attitude towards the system, this does not necessarily guarantee trust [38].
- *Unfamiliar steps:* Given that users often base their mental models of a system off of previous experience or knowledge, their ability to successfully participate in new systems like E-cclesia can be affected. In particular, verification is an extra step that is integral to E-cclesia to ensure that their vote will be counted in the final result. However, communicating the importance of this step can be difficult to users if they have never needed to do this before. This is apparent in the study by Acemyan et al. in which 53% of the time participants could not/did not verify their vote [20]. Even if users know they need to verify their vote, ensuring that they do so successfully is still a challenge because this process itself will be unfamiliar to them.
- *Unnatural steps:* In the unlikely scenario that users have used existing self-tallying voting systems like Helios[21] which have separate verification steps, E-cclesia still has the added obstacle that the user needs to wait, possibly for several hours, to verify a vote because of the involvement of blockchain technology. This waiting time is very unnatural and can easily lead to issues like forgetting to verify or recasting a vote. Even if verification is performed successfully, it is likely that the waiting time will create skepticism in the system because all other systems appear to function just as well without this obstacle.
- *Language:* In addition to the difficult standard terminology used in voting, the E-cclesia protocol involves even more complicated terms related to the security of the system that even native speakers may struggle with. Although some of the more complex mathematical proofs behind the protocol do not need to be understood, unfamiliar terminology like blockchains and encryptions will need to be used to communicate the basic security behind the protocol to users and

motivate the users to complete unfamiliar steps. This was demonstrated in the study by Weber and Hengartner in which over half of the participants would not complete their vote, primarily because they did not have the information or motivation to verify their ballot[84]. The use of this terminology cannot be avoided or else adequate transparency will not be achieved and consequently trust will not be gained.

2.3.3 Evaluation methods

Think aloud

Assesses the usability of a system by asking evaluators to articulate what they are doing and thinking as they complete a set of tasks that typical users of the system would complete [47] . This can help identify usability issues that may not be found through other methods because it provides direct insight into where users may get stuck, what they might find frustrating and what they are confused about. A think aloud can be concurrent (when they articulate their thoughts during the task), or retrospective (when the participant comments on the process after they complete the task). Notably think alouds only need 5 participants to discover the majority of the usability issues within a system [67].

Cognitive walk through

Assesses the usability of a system by examining how easily users can complete tasks using the system. During a cognitive walk through session, evaluators work through a set of scenarios where users of the system would typically perform and answer a set of questions that help identify potential usability issues.

The System Usability Scale (SUS)

A set of 10 standard questions designed to quickly evaluate the systems usability in a standardised way [28]. Participants answer the questions with a response ranging from "Strongly Agree" to "Strongly disagree". The SUS score of the system is then equal to 2.5(X+Y) where:

$$X = (\text{the sum of points in odd numbered questions}) - 5$$

$$Y = 1 - (\text{the sum of points in even numbered questions})$$

Though the scale is from 0-100, the resulting score can be considered a normalised score rather than a percentage of how usable the system is. The SUS has become an industry standard referenced in over 1300 articles. Conveniently, a comparison of the SUS scores and their acceptability levels has been produced by Bangor et al. in 2009 [24]and can be found in Appendix C.

Chapter 3

Methodology

3.1 User Centred Design (UCD)

The primary research method I used for this project is the User Centred Design (UCD) approach. UCD, coined by Don Norman in 1986[69], is an iterative design process that focuses on the users of the system and their specific needs throughout each design phase to help gain a deeper understanding of who will be using the system. I chose to follow this research method because it allowed me to engage with users and understand their needs effectively within a short period of time.

Though there is no formally agreed upon definition of UCD, some of its key principles include considering the user's needs to guide early development, early and continuous prototyping, and involving usability experts early in the development lifecycle [46]. Similarly, though the phases of the UCD process have not been explicitly defined, it has a general process[70]. This process has been used to inspire the 3 main phases of this study; informing the design, creating design solutions, and evaluating the final design.

3.2 Phase 1: Informing the design

In this section, I aimed to identify the people who will use the application and their requirements. To identify the users I created personas; fictional characters that represent the target audience for a product [11]. In the design process, personas help make informed decisions by identifying which features are most important to each user type. For studies, personas can help develop realistic scenarios for participants to interact with the system as if they were in that context.

To identify the functional requirements of the admin and voter systems, I referred to the requirements found by Oshima[40] and conducted a series of interviews with the E-cclesia development team. I decided to do interviews in addition to the existing requirements as I felt that they were not sufficient to capture all user goals and because the system had been further developed since Oshima's project was written. I chose to conduct interviews to gather this information as I felt this method would allow me to get answers quickly and discuss points of misunderstanding. These interviews were unstructured as I had limited understanding of the E-cclesia system at the time and needed to be able to follow up on points within the conversation that were particularly confusing.

Using the information gathered from these interviews, I was also able to visualise user flows for the admin system and voter system [76]. It was important to finalise the user flows with the E-clessia team to ensure that our mental models aligned and to make sure navigation would be as clear as possible for users. Note that these user flows specifically follow the "Happy path", paths users are expected to take to achieve a particular goal without any troubles[81]. I focused on this path to maintain the clarity of the diagram.

To identify the design requirements of the admin and voter systems, I conducted a short literature review of past usability studies on e-voting systems. Typically, design requirements are determined through questionnaires or interviews that draw on user's experience with similar systems. In this instance, as E-cclesia is a unique system due to its electronic and decentralised nature, finding participants with such desired experience would be unfeasible and so a literature review was the most favourable alternative. In addition to e-voting systems, I included studies on e-commerce systems and mHealth systems because they also aim to gain user's trust and communicate the high security of the system without compromising on usability.

3.3 Phase 2: Creating design solutions

After gathering information for both the admin and voter system, I decided to focus on designing solutions only for a mobile implementation of the voting system as focusing on one system would result in a higher quality and comprehensive design. I chose to focus on the voting system because the issue of voter verification had been identified as one of the most critical yet weakest areas of existing e-voting systems as it is a combination of all of the challenges of the E-ccelsia system outlined in section 2.3.2. It requires a good level of transparency and understandable language to motivate users to verify a vote. Furthermore, the process is unfamiliar and unnatural which can prevent even the most motivated users from verifying their vote successfully. This is reflected in the poor success rate of verification across the three different voting systems studied by Acemyan et al. [20].

To decide upon a final system that is usable and meets the requirements set in Phase 1, iterative design and feedback phases, known as 'fast feedback' session's were conducted in a think aloud style. For these sessions, I recruited experts in the field of Human Computer Interaction (HCI) because they would be able to provide the most informed feedback. The designs in this phase were high-fidelity prototypes designed on Figma[6], a popular design tool used for user interface and user experience designing. I chose to use Figma over other design tools because a majority of HCI experts would be familiar with the software and so inability to use the software would not affect the feedback of the interface itself.

I chose to follow a think aloud style for these sessions because it would allow me to get detailed information on how users will approach the system, even from a small sample size. The scripts for these sessions were semi-structured to ensure that I covered the key points but also allowed me to follow up on areas I thought were more interesting or relevant. The feedback in each session was then analysed using the critical incident technique as this was seen to be effective in the evaluation conducted by Acemyan et al. [19]. This analysis technique focuses only on the most serious user errors observed, rather than attempting to address every possible usability struggle, slip, or mistake that could occur because this list of issues could be very long.

Given the limited availability of the participants in the fast feedback sessions, each session focused on different user goals. The first fast feedback session focused on goals that users would be familiar with; vote casting and viewing the results of the election. The second fast feedback session focused on goals that the users were likely to be unfamiliar with; joining an organisation and election and vote verification. I felt that structuring the sessions in this way would be most time efficient because the feedback could be more detailed when focused on fewer specific goals.

3.4 Phase 3: Implementation and evaluation

To assess that the final implementation is usable and meets the requirements set in Phase 1, I used the feedback and designs from phase 2 to implement an Android[1] application and conducted user studies.

These user studies were conducted over the whole day to simulate a real election and allow separate evaluations on the voting stage and the verification stage. I chose to evaluate these stages separately because they are two distinct stages of voting - each come with their own usability issues therefore, users may have different opinions of each stage. As verification is such a significant usability challenge in self tallying election and the E-cclesia system, it was also important to evaluate the usability of this step separately. Several other studies on self-tallying voting systems have also chosen to perform separate analysis on the voting and verification stages because it was the most logical way to evaluate them [20][19][55].

In addition, I wanted to identify if the level of detail in the explanations in the application were transparent and clear enough to motivate frequent user verification as these were also some of the key challenges of the E-cclesia voting protocol. To do this, half the participants in the study only had the basic information that was part of the application, and the other half of the participants were given more information about the need for verification and then their usability results were compared.

Similar to the fast feedback sessions in Phase 2, this evaluation was run in a think aloud style with a semi-structured script to get more detailed feedback and allow for extra questions. The tasks in this think aloud were accompanied with an email from the organisation setting up the election and rather than explaining the task, the email would have some instructions on what the user should do next. This simulates a real election as I would usually not be telling each user what the tasks are, their only guidance would be from the emails. The feedback from the think aloud sessions was then examined using the critical analysis technique similar to Phase 2.

3.4.1 Evaluation metrics

In addition, to the qualitative feedback that was gathered from the think aloud, further evaluation metrics were used to gather quantitative data on the usability of the application and compare it the data previously reported of other e-voting systems.

Usability

The National Institute of Standards and Technology (NIST)[68] is a non-regulatory agency of the US Department of Commerce that is responsible for developing and maintaining a range of standards in various areas of technology including voting systems. To measure the usability of voting systems, NIST recommends using the three metrics defined by the ISO 9241-11 standard[61][51]:

- Effectiveness: The accuracy and completeness that users can achieve specified goals. This will be measured by the number of tasks completed successfully without assistance.
- Efficiency: The resources for users to achieve their goals accurately and completely. This is usually measured by the time taken to complete the task but in this case, as a think aloud requires participants to articulate their thoughts at their leisure, this would affect their performance and so will not be measured. The decision to omit this metric was also made in the Weber and Hengartner's usability study of Helios [84].
- Satisfaction: The user's subjective response to using the system. This will be measured using the SUS outlined in section 2.3.3. Though the SUS has standard questions, I will be using a slightly modified set of questions that have been shown to have no impact on the scales' readability but can be easier to understand [20].

Since these metrics are widely utilized in other usability studies[20][19][41][30], the data collected here can be compared to previous studies for insights into how this system compares to existing systems.

Verification and trust

In addition to the SUS questions, each participant was asked additional questions using a scale of "Strongly agree" to "Strongly disagree" to find out more about their opinion of the usability of the application. These questions (found in Appendix M.0.1 and M.0.2) primarily focused on how user's trust in the system was changed after the verification step and were based on the questions in the study conducted by Karayumak et al[54]. Ensuring that the verification process does not affect the user's trust in the system is crucial because it is an important step in decentralised voting systems and in E-cclesia. However, as outlined in Section 2.3.2, it can be very difficult for users to complete this task successfully because it is a new . If the verification process affects the user's trust, they may not use the system again and may not perceive it as secure.

Chapter 4

Informing the design

	Admin: Alex Smith	Voter: Pat Jones	Voter: Jordan Hwang
Background	- Studies BSc Computer Science and mathematics - Final year of study - Has only voted in a school election before - Is sceptical about systems that promise high levels of security	- Studies BA History - Third year of study - Has voted in a national election using a DRE system - Is indifferent to the security of systems	- Studies BA Social anthropology - First year of study - Has never voted before - Usually trusts the security of systems that claim to be secure
Pains	- Does not have time to learn a complex system - Does not have time to answer other peoples questions about an election	- Encouraged to vote because of a friend but is not very interested - Does not have time to learn a complex system	- Is nervous about voting in elections
Goals	- Wants to organise an election where the results are not contested	- Wants to finish voting as fast as possible	- Wants to vote as intended

Figure 4.1: Personas that characteristics and opinions that some of the users may have. Alex Smith is a persona for the admin system and Pat Jones and Jordan Hwang are both personas of the voter system.

4.1 Personas

To help understand a user's goals, behaviors, and preferences, I created three personas - one admin user and two eligible voters, as can be seen in figure 4.1. Typically, personas are designed from user research, however, given the unique challenge where anyone in the population could be a user, I created personas focusing on three characteristics that could affect a user's opinion and fluency of the application:

- *Background/knowledge of computer security:* One persona has advanced technological knowledge through studying informatics. Therefore, the systems should use language that is simple enough for all users to understand but technical enough to satisfy those with more computing knowledge.
- *Trust in e-voting:* The three personas cover three different opinions of E-voting - trusting, sceptical and indifferent. Therefore the system should aim to gain at least some trust from all the users by the end of their interaction with the system.
- *Experience in voting:* Two of the three personas have some experience with some method of voting. The system should be easy to navigate regardless of a user's prior voting experience.

4.2 Functional requirements and User flows

To ensure that users can successfully achieve their goals using this system, I identified its functional requirements. This was accomplished using the requirements found by Oshima [40] and a series of 3 interviews with some members of the E-cclesia development team; Dr. Myrto Arapinis, Dr. Thomas Zacharias, and Pavlos Georgiou. At the beginning of each interview, I presented a series of low-fidelity wireframes (found in Appendix D) to convey my understanding of the E-cclesia system at the time and its requirements which would then be discussed and corrected by the team. Low-fidelity wireframes are used in early project stages to verify content accuracy and do not need to include design elements like color and font which allowed for quick feedback. These sessions took longer than expected as the team was still developing the protocol at the same time and so some questions had not been considered or did not have a definitive answer yet.

4.2.1 Admin system

The main goals I identified for admin users were 1) to create a list of voters, 2) to set up an election and 3) to send emails to lists of voters. The functional requirements that need to be met for these goals to be realised are as follows:

1. An election authority is able to register an election by specifying details such as: the title, question, description, options to select from (and the description of each option), casting start time, casting end time, and a list of verified voters. Once created, these details are immutable. [1]
2. The election authority is able to view the status of the election at various points in the protocol. The different states of the election from an admin's perspective are as follows:
    - *Drafted:* The process of setting up an election has begun howeve,r the details can still be edited as they have not been sent to the blockchain yet.

[1]The question and option description are not currently parameters for the E-cclesia protocol but would ideally be added for further clarity for voters.

- *Created:* The election has been set up and its details are now immutable as the smart contract is officially being set up on the blockchain. At this point eligible voters will be joining the election, thus the system should present the number of eligible voters that have registered at the time.
- *Active:* The casting period of the election has started. The system should present the number of votes that have been cast at this time.
- *Closed:* The casting period of the election has ended. The system should present the total number of votes cast and, if the tallying has finished, the results of the election.

3. An election authority is able to send an email to selected verified voters. If this email is the invitation to register for an election, a template (with the automatically generated QR code to join) will be provided but can be edited. This includes re-sending emails to eligible voters who have already received the same email, as well as eligible voters who have not received the email yet.
4. An election authority can create new lists of voters or edit an existing list of voters through a CSV file, raw text, or by individually selecting a voter.
5. The system must send a reminder to the election authority every 6 months to update the details of the list of voters.

The user flow of these goals and requirements can be found in Appendix E.

4.2.2 Voter system

The main goals I identified for the voter were 1) to join an organisation, 2) to join an election, 3) to vote in an election, 4) to verify their vote has been recorded in the final results of the election and 5) to view the results in an election. The functional requirements that need to be met for these goals to be realised are as follows:

- An eligible voter is able to join an organisation and election using a personalized QR code sent by the election authority[2]. When this action takes place, the verification key and ID of the eligible voter is passed to the election authority.
- Voters who are not eligible for an election cannot add their verification key and ID to the eligibility List.
- An eligible voter is able to view the status of the election at various points in the protocol. The different states of the election from a voters' perspective are as follows:
  - Joined: The eligible voter has joined the election and their verification key and ID has been passed to the election authority.
  - Voting started: The eligible voter can begin casting their vote for the election.
  - Vote cast: The eligible voter has cast their vote. This vote is now immutable and in the process of being recorded on the blockchain.

[2]The current secure broadcast channel chosen by the E-cclesia team to send this information is email

  - Vote recorded: The vote cast by the eligible voter has now been recorded to the blockchain.
  - Results calculated: The result for the election has been tallied.
- An eligible voter is able to view the results of the election as long as they have joined the election. This applies to voters who have voted and those who have abstained [3].
- An eligible voter is able to view all the organisations they are on the electoral register for and all the elections they have joined.
- It was mentioned that implementing notifications into the voting system may be troublesome because it would require keeping the application running consistently in the background for many hours. This could consume a large amount of power from the users' device and may affect satisfaction. Therefore, though not a requirement, this point should be acknowledged.

The user flow of these goals and requirements can be found in Appendix E.

4.3 Design requirements and principles

To help identify and evaluate potential usability issues in the design, I will be following Nielsons 10 Usability Heuristics[66], guidelines chosen because they are the standard used by most UI designers, are backed by research, and promote a user-centered approach to design. Given the unique usability challenges of this project presented in section 2.3, I decided to identify further design requirements through a literature review of usability studies primarily on e-voting systems but also some papers on e-commerce systems, and mHealth systems. Using NVivo[62], a qualitative data analysis tool, I identified the 3 main themes for the design requirements; language, process, and interface.

4.3.1 Language and communication

Due to the uncommon terminology of decentralised voting and issues with user's attentiveness or memory, the language used as well as the way the information is communicated needs careful consideration. Too much technical language can overwhelm users and unclear communication can be confusing and make them doubt the reliability of the system.

L1 The language used in the design should be familiar to all users and still be technically correct. [55][65][33][84]

L2 The terminology should be consistent throughout the system. [55][48][26]

L3 The system's communication should be scannable, simple and clearly worded. [55][19][39][54]

[3] In the current implementation of the E-cclesia protocol, abstaining voters cannot view the results however this should be amended for improved usability satisfaction

L4 The systems feedback should be relevant to the situation. [48][49][55][65][72][26]

L5 Specific and clear instructions and information should be provided at critical junctions throughout the system. [20][19][48][49][55][65][58]

L6 The design should clearly communicate when an error has occurred, why it has happened and how to resolve it. [19][55][65][26]

4.3.2 Process and navigation

Due to the unfamiliarity: both the steps and security of the E-cclesia voting system; making the process as similar to other more familiar systems; and explaining new steps, should be prioritised. Poor navigation can introduce opportunities for user errors and if a system requires too much effort from the user, they may not have the motivation to complete a step.

P1 The process a user needs to take to achieve their goals using the system should be intuitive and familiar. [20][19][39][55][72][78]

P2 The number of steps in the process a user needs to take to complete their goals should be minimal. Where possible, these steps should be automated. [19][39][55][54][20][84][49]

P3 The design should always communicate the current step in the overall process to the user. These steps should be distinct and easily identifiable. [55]

P4 The design should clearly communicate the need for each step to the user unless it is highly intuitive. [20][39][55][72][58][84]

P5 The system should present users with the option to learn more about the technical elements of the process to verify the security of the process if desired. [39][58]

4.3.3 Interface and design

Due to the large skepticism surrounding voting and new systems of voting in particular, the interface should give users the impression of a trustworthy system through good design.

I1 Features of the interface should be clearly labelled or intuitive. [19][49][55]

I2 The interface design should be consistent throughout the system. [55][65]

I3 The interface should be of high perceived quality, primarily through simplicity and following design standards. [65][72][37][41]

Chapter 5

Fast feedback 1

5.1 Aims

The aim of this study was to get feedback on the first Figma mock-up of the UI for the mobile application of the E-cclesia voting system and to understand how usable it is. In particular, this mock-up focused on the tasks users would be most familiar with; casting a vote and viewing their results. Some of the unique design elements of the system include:

- **Election colour:** To show the status using visual cues, each election can be a range of three colours/shades. Regular blue indicates an active election where a user can perform an action. A translucent shade of blue indicates an active election where the user needs to wait for a process to be finished before they can perform an action. Grey indicates that the election is closed.
- **Election status bar:** Additionally, each election has status bars at the top to indicate the status of the election using colours (green and grey) and text ("Election not started", "Voting open", "Tallying votes", "Results ready").
- **Election timeline:** Each election has a timeline at the top of the election page indicating the current step and how many steps are left to cast a vote. These three steps correspond to joining an election, selecting options and finally casting the vote.
- **Button instructions:** Although the button label should be clear, some buttons have added text below them to give further explanation on the action that will be produced once the user presses the button.

Examples of these design features can be seen in Figure 5.1. The full Figma mock-up can be seen in Appendix F.

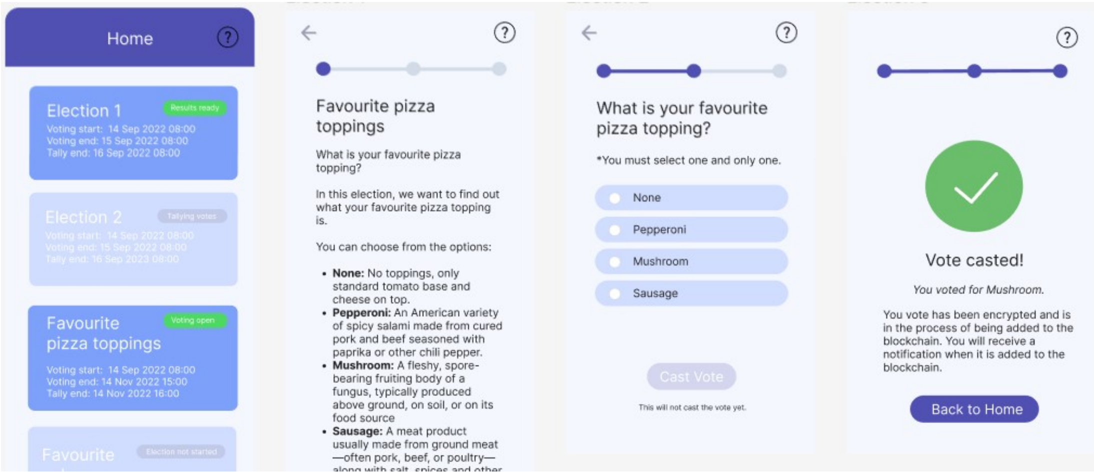

Figure 5.1: Figma mock-up of selected pages of the first iteration of the E-cclesia voting UI design. These pages specifically illustrate the design decisions from Section 5.1 but the full Figma mock-up can be viewed in Appendix F. From left to right: election home page, election information page, vote selection page, and the vote cast confirmation page.

5.2 Participants

This study involved one HCI expert, henceforth referred to as H1, who was a PhD student at the University of Edinburgh Technology Usability Lab in Privacy and Security (TULIPS)[13] at the time. The TULIPS lab is a part of the University of Edinburgh's Security and Privacy group that specializes in improving the usability of security and privacy technologies. Given their knowledge in both HCI and security, I felt that this participant would be able to give me the most informed feedback. I reached out to this participant by recommendation of my supervisor and the list of course contacts on the University of Edinburgh HCI course page[14]. I would have liked to have more HCI experts participate, however, due to their workload at the time of the year, other HCI experts were unavailable.

5.3 Procedure

The session began with the participant giving their consent for the study and for the audio to be recorded so the session could be reviewed in more detail later. A script, found in Appendix G, was used to ensure that all the key points were covered and the session was run efficiently. H1 was given each task one at a time rather than all at once to simulate the wait times expected between the different election states and instructed to verbalise their thoughts. As the participant was an HCI expert, they already had experience participating in think alouds and using Figma and so this did not need to be explained. Due to the limitations of Figma, H1 was instructed to vote for specific options in the elections as the other options were non-functional.

5.4 Results

The participant was able to complete all the tasks successfully but had several recommendations and points where the system did not produce the actions that they expected. Below is a summary of the results categorised by the task. Aspects of the application that could be improved have been labelled with an ”N” for negative and any particularly useful aspects that the participant commented on have been labelled with ”P” for positive.

Vote casting

P1 *Clear election state.* H1 could clearly understand that the “Favourite colour” election had not started yet: “it is quite clear that it hasn’t started yet because it is greyed out”.

P2 *Straightforward design.* The navigation was very clear to H1: ”I like the design, it’s really nice. it is quite straightforward. It has a lot of consistent design patterns that we usually use.”

N1 *No change observed on casting.* H1 was confused by the vote selection page and the page confirming if the user would like to proceed with casting their selection. They both looked the same to them and it seemed as if their action did not result in any change: “when I clicked on the cast vote button, it was very confusing, because the text on the bottom (below the button) was quite small”. Instead, they expected a dialog box stating that they had selected an option to cast.

N2 *Insufficient information after casting.* H1 felt unsatisfied with the information provided by the cast confirmation page: “it would be helpful to have maybe an info button or a link with more information or a link about how this process works.”

N3 *Would prefer push notifications.* Though the states of the elections were clear, H1 mentioned that they would also find it convenient if there was a push notification for when the election started.

Viewing the results

N4 *Non-descriptive language.* H1 felt that the label “Description” of the drop-down box was too vague and provided no information about what information the box contained and what result clicking on the button would produce: “I just clicked on it because I wasn’t sure what it was”.

N5 *Expected to see election information.* H1 understood the results of “Election 2” had not been tallied yet however they did not expect a dialog box: “I expected to be brought to a page like this [the election information page] where you have the description but no outcome yet. And with some description like “currently tallying election’...so that I can go back to the description and see things [the information]”.

N6 *Unfamiliar terminology.* H1 had to ask if the word “tallying” meant that the votes were being counted.

N7 *Confusion between the same election name.* H1 commented that there could be an issue with too much scrolling if a user was participating in lots of elections and if there were elections with the same names but from two different organisations: ”imagine two societies had their AGM and they both called it ’election’, then you don’t know which to vote in”.

5.5 Discussion

Generally, the navigation and design were clear to H1 and aligned with their mental model of the system. This is unsurprising as vote casting and viewing results are tasks that are familiar processes for both the user to enact and myself to design. However, there are still several improvements H1 identified that would make the system status more clear and increase voter confidence.

The most critical feedback was that H1 did not realise when they were on the confirmation page that was used to double-check if they were sure they wanted to cast the current selection [N1]. In a real election, users may proceed with this step even though they are not sure about their selection yet because they expect a different confirmation page. As users cannot change their vote once it has been cast, due to the immutable nature of blockchains, this could result in incorrect votes and unhappy users. Fortunately, this issue can be resolved by replacing the text below the button with a dialog box, as this was expected by the user and was noted as more visible than the text below the button.

Another critical potential issue that was overlooked by both myself and the E-cclesia team was the scenario in which two different organisations set up elections with the same name [N7]. Though each election also has accompanying information, there is a chance that they are only filled with generic information or information similar to the other election with the same name. This could cause voter confusion which could affect their ability to cast a vote correctly. In this case, a user should be able to filter the elections according to organisations. In addition, a note should be added in the functional requirements of the admin system in section 4.2.1 that an organisation cannot repeat election titles.

Error	Violated requirement	Suggested improvement
	Vote casting	
N1: No change observed on casting	P3	Confirm cast selection using a dialog box.
N2: Insufficient information after casting	L5, P5	Add link to help page on cast confirmation page with information on how the E-cclesia protocol works.
N3: Would prefer push notifications	I3	Noted, but no notifications will be implemented as explained in Section 4.2.2.
	Viewing the results	
N4: Non-descriptive language	L3	Change "Description" to "Election information".
N5: Expected to see election information	P1	While user cannot perform an action, allow election information page to be viewed but indicate that a process needs to be completed before the user can act.
N6: Unfamiliar terminology	L1	Replace term "tallying" with "counting".
N7: Confusion between same election name	I1	Include option to filter elections by organisation and limit organisations from being able to repeat election names

Table 5.1: Feedback of possible errors/areas of improvement along with the design requirement (from section 4.3) which the error violates and how this error can be resolved. As per the critical incident analysis method outlined in section 3.3, it does not include every incident observed throughout the study.

Chapter 6

Fast feedback 2

6.1 Aims

Like the first feedback session, this study aimed to get feedback on the usability of the second iteration of the Figma mock-ups of the E-cclesia voting system. In particular, this mock-up also focuses on the unfamiliar steps of the E-cclesia protocol; setting up and verifying a vote. Some of the unique design elements of the system include:

- **Instructions for setting up:** Both the organisation home page and the election home page have instructions on how to add an organisation or election when nothing has been added yet. These inline instructions have been included because these extra steps may confuse users.
- **Election timeline:** The election timeline has been improved to clearly indicate all the steps in the election alongside coloured icons (green tick and red cross) for more clarity. Some of the steps also have subtext that provide extra information on the step. An information button has also been included to provide more details about each step.
- **Security focused confirmation page:** The page that the user sees to get confirmation that they have cast their vote includes information about what will happen to their vote, how it is kept secure, and what they need to do next. Though the information is concise, the keywords ”blockchain” and ”encryption” have been used purposefully as technical terms which can help instill more confidence in the quality of the security of E-cclesia. Information on the next step (verification) has been highlighted in bold to draw attention to this text as it is the most important section of this page.

Examples of these design features can be seen in Figure 6.1. The full Figma mock-up can be seen in Appendix H.

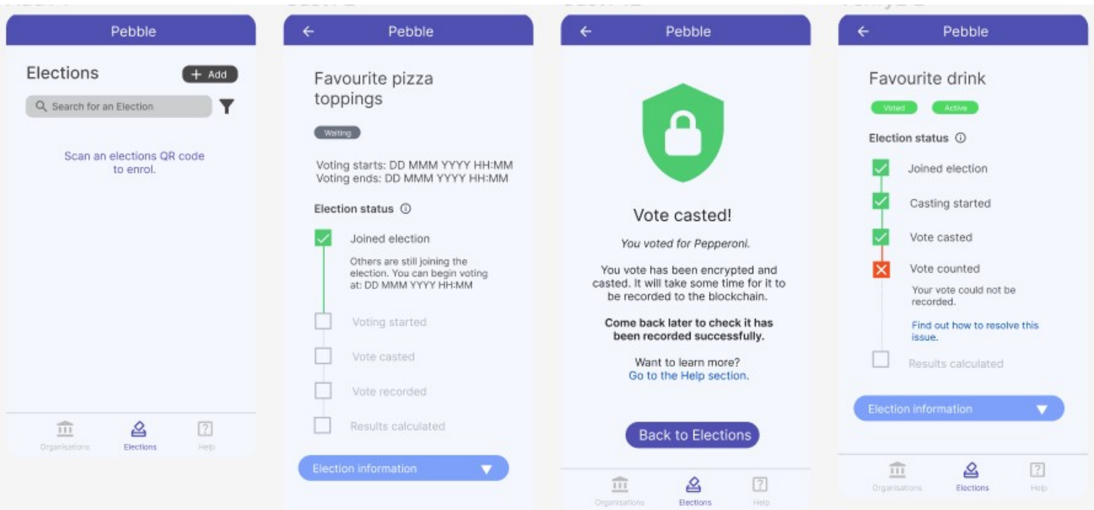

Figure 6.1: Figma mock-up of selected pages of the second iteration of the E-cclesia voting UI design. These pages specifically illustrate the design decisions from section 6.1 but the full Figma mock-up can be viewed in Appendix H. From left to right: empty election home page, election information page when a user has just joined the election, vote cast confirmation page, and the election information page when a vote has been cast but not recorded.

6.2 Participants

This study involved two HCI experts, H1 who was also involved in fast feedback 1, and a new participant, henceforth referred to as H2. Like H1, H2 was a PhD student at the University of Edinburgh TULIPS lab[13] at the time and was selected because of their expertise and knowledge of both usability and security. I reached out to these participants by emailing the list of TULIPS staff and students and by recommendation of H1 and my supervisor. I would have liked to have more HCI experts involved in this stage, however, due to their workload at the time as well as their personal circumstances, other HCI experts were unavailable.

6.3 Procedure

At the beginning of the session, the participants read the Participants Information Sheet (PIS) (found in Appendix A) and gave their consent for the study (the consent form can be found in Appendix B). Again, a script, found in Appendix I was used to help guide the session. This script was more detailed and carefully written as comments from the first feedback session highlighted that greater clarity in the instructions could be achieved. The participants were given one task at a time again and had prior experience participating in think alouds and using Figma through their work in the field of HCI. This session also included some follow-up questions focusing on how this system compared to their experience with other systems and their perception of the security of the application. Given that verification is a new step, I wanted to check that this still made sense to users and was easy to perform.

6.4 Results

Both participants were able to complete all the tasks successfully and within minimal clicks. Similar to the previous fast feedback session, below is a summary of the results categorised by task, where positive feedback has been mark with ”P” and negative feedback/improvements have been marked with ”N”.

Setting up

P1/N1 *Elections joined through a separate page.* The two participants had contrasting opinions about this feature of the application. H1 did not expect to go to another page to join an election: “I would assume I could go through the organisation and then on that page I could add an election.” On the other hand, H2 had no issues joining an election: “[The elections navigation button] is right there [at the bottom] so it’s helpful.” This contrast indicates a difference in mental models of users of the system and can be challenging to address because it is difficult to know which mental model is more common.

N2 *Unclear election organiser.* H1 was unsure if the election they had joined was the correct one because there was no indication which organisation was running the election.

P2 *Liked QR code feature.* Both participants appreciated the fact that you could join an organisation and election using a QR code. They found this more convenient than typing.

N3 *Lack of instructions.* Though H1 had no difficulty joining an election, they would have preferred more guidance via the interface: “I only saw the button saying add because you used the word add in your question. It was not entirely clear to me [that I joined an election through this button], especially for my first time doing this.”

Vote casting

P3 *Clear election status.* Both participants selected the election timeline as one of their favourite features of the system because it clearly communicated the current status of the election and helped them keep track of what stage of the process they were in.

P4 *Clear confirmation dialog box.* Both participants were aware that that their vote would be cast because of the dialog box and appreciated the confirmation message.

P5 *Reassuring security reminder.* Both participants appreciated the message on the vote cast confirmation page as it mentioned terminology that reminded them that the system was more secure than other common platforms for voting. Specifically, H1 liked the mention of “blockchains” and H2 liked the mention of “encryption”.

Vote verification

N4 *Unclear time for verification.* H1 felt uneasy because there was no indication on how long the system would take to verify a vote had been recorded. This is information they wanted to be able to have more confidence in the application.

N5 *Clearer indication of errors.* H1 easily identified that their vote in the election was not recorded however they think it could be made more prominent: "It would be nice if it was clearer if something went wrong and if something was successful. Making sure to use the colour red only if it was very important."

N6 *Lack of election journey information.* H1 understood each state of the vote however would have liked more information on how long the verification step took.

P6 *Intuitive system.* Both participants felt they understood the purpose of each step and that the system performed how they would expect. H1 felt that the flow was very similar to what they experienced in past elections and H2 felt that the flow matched her expectations of an election. H1 also mentioned that it was less of a hassle than other polls they had taken part in before on platforms like Slack and found it "very simple for something I would have expected to be quite complicated for a user."

Viewing the results

P7 *Results were easy to read.* Both participants liked the results page and found it easy to understand. H2 particularly liked the use of graphics: "I like the end with the results as it was very clear and visually appealing. It's usually just numbers but I like the colours and visuals."

N7 *Misleading use of colours.* H1 could clearly identify the election closed when the status bar turned red, however this could be misleading: "it would be better if the state for closed was grey and red to indicate that there was an issue".

6.5 Discussion

Generally, there was more positive feedback in this iteration of the design than in the first feedback session and a majority of the changes will be much smaller in comparison to the first feedback. Notably, there are very few improvements pointed out by the HCI experts for the tasks of casting a vote and viewing the results, showing that any issues the experts identified in the first fast feedback have been successfully resolved without consequences in other areas. The most critical observation was the contradicting mental models for joining an election [N1]. Problems here will prevent users from casting votes successfully as it is one of the first steps they need to do and can influence their opinion with their future experience with the system. For the next iteration of the design, I will keep the same design for joining an election because both H2 and I follow this mental model. However, if the majority disagree with this mental model in the next evaluation, this should be changed.

Both participants largely trusted the security of the system, even when their vote had not been recorded for an election. This error did not change their opinion of the security of the application, only that from a usability standpoint, this could be indicated more clearly and should not occur often. Interestingly, H1 mentioned that they would trust the application if an organisation they trusted used it: “In general, I didn’t even think about [the security], I just assumed it was safe. I assume that the organisation picked this application because it was trustworthy, and I would trust the organisation”. H2 also mentioned that their use of the application would depend on how often an organisation uses it. This suggests that good usability of the admin panel will be critical when it is designed as convincing the organisations to trust and use the E-cclesia voting system frequently will influence how often voters use the system and their trust in the system.

Error	Violated requirement	Suggested improvement
	Setting up	
P1/N1: Elections joined through a separate page	P1	Keep design the same for now, but if other users face issues with this step in the final evaluation, the design should be changed.
N2: Unclear election organiser	I1	Include name of organisation that is running the election in the election information page.
N3: Lack of instructions	L5	Edit inline instructions on organisation and election home pages to be more precise and specify the ”Add” button.
	Vote casting	
	No feedback to improve upon was given for this section.	
	Vote verification	
N4: Unclear time for verification	P5	Include information of how long verification will take in the help page and link the help page in the cast confirmation page.
N5: Clearer indication of errors	L6	Change the colour of the election item on the election home page to red when an error occurs.
N6: Lack of election journey information	L5	Include timestamps of when each step was completed in the election timeline.
	Viewing the results	
N7: Misleading use of colours	I3	Use grey to indicate election is closed on status bar. Reserve red for errors.

Table 6.1: Feedback of possible errors/areas of improvement along with the design requirement (from section 4.3) which the error violates and how this error can be resolved. As per the critical incident analysis method outlined in section 3.3, it does not include every incident observed throughout the study.

Chapter 7

Implementation

The final application design has been developed using the results from the fast feedback sessions (from Section 5 and 6) and the information gathered in Section 4. This section details the decisions behind each design element and the implementation of the application. The screenshot of all the pages can be seen in Appendix J.

7.1 Design

Although careful design considerations have gone into each feature of the application, the following features can be considered unique to this application.

Welcome pages: The application begins with a series of welcome pages outlining the sequence of steps a user needs to take to successfully vote in an election. I chose to present this information here because users often get confused by the steps in e-voting systems which can lead to security and usability errors[20]. The design of these pages is very minimal to keep users focused on the information and make sure they do not get demotivated by large amounts of text before they even begin.

Instructions with explanations: Where possible I attempted to add inline instructions such as the when a user has not added organisations or elections yet as mentioned in Section 6.1. These instructions also include reasoning on why this step needs to be taken (although it does not go into much detail because the instructions need to be concise).

Election status indicators: The status of each election is indicated by the colour of the election item itself on the election home page (it turns form blue to red when an error occurs), as well as through the election status bars at the top. The status bars have been improved to become one word only states to keep the design uncluttered but maintain the sentiment: Waiting(grey), Active(green), Closed(grey), Voted(green). This has been taken as inspiration from the status bars on the voting platform Snapshot[18].

Election timeline: Each election has its own timeline that displays all the steps a user needs to take to vote in an election successfully. This timeline has been taken as inspiration from the e-commerce website Amazon as this should be familiar to most users. An information button has also been included to provide more details about each step. Although this information is more about the ballot rather than the election, this terminology could confuse users and so this has been labelled as ”Election status”.

Careful selection of terminology: Throughout the system I made sure to use words that express the same sentiment as that expressed in the original E-cclesia paper [23] but could be easier to understand for users who are not confident in English or voting terminology. This includes ”organisation” instead of ”election authority”, ”count” instead of ”tally” and ”vote” instead of ”ballot”.

Single organisation information: It was suggested in Table 5.1 to filter elections by organisation and in Table 6.1 to include the name of the organisation on the election information page. However, as there was limited time to develop the application and the final evaluation using this implementation will only be conducted using a single organisation, this feature was omitted. In future implementations, this should still be included.

No push notifications: Participants have mentioned their preferences for push notification, as seen in Table 5.1. However, due to the E-cclesia developers suggestion to assess the usability of the application without notification in Section 4.2.2, this feature has been omitted in this implementation.

7.2 Technology

7.2.1 Framework and programming language

The application was developed using Android Studio[3] and the Java programming language[71]. Android studio is the integrated development environment (IDE) designed specifically for applications on the devices using the Android operating system. I chose to develop an application for the Android operating system because it is the most popular phone operating system at the moment and so more users will be able to use this application if it is on Android than on other operating systems[80]. Android studio also has easy to use debugging tools and a built in emulator which are useful for development.

Although Kotlin[53] has recently become the more popular choice for Android application development in the industry [60], I chose to write this application in Java because I was more familiar with it and it has a more extensive ecosystem of libraries and frameworks [44]. It also has a much larger developer community and extensive documentation and so it would be easier for me to learn how to use new techniques in Java over Kotlin.

7.2.2 Libraries

Though other libraries were also used, in this section, these are a few of the more uncommon libraries specific to this application.

Material Design. Material Design[9][10] is a design language developed by Google[7] that aims to provide a consistent and intuitive user experience across different devices and platforms. This library helps to ensure consistency across the interface and has built-in design features that follow the latest practices because it is maintained by Google. Material Design also includes accessibility features like high contrast text that make it easier for users with disabilities to use the application.

AndroidX. AndroidX[4] is a collection of the Android Jetpack[2] libraries and contains a set of libraries that provides developers with a range of components, tools, and features to help them build modern Android applications. For example I used the Recyclerview[22] library. I chose to use this suit of libraries because they allowed me to incorporate more advanced features that would make the interface more usable without coding it myself.

Code Scanner. [29]I used a library to implement the QR code scanner functionality. The source I used was based off of the ZXing scanner[29], an open-source, multi-format barcode scanner library for Android.

Pie Chart. [52]I used a library to create the pie chart for the results page in the application. This is a popular Android chart/graph view library.

7.3 Challenges of implementation

No back-end to work with. Prior to the design and implementation, the expectation was to be able to view a demonstration of the command line version of the E-cclesia protocol being used and to connect the front-end of the application I was creating to the back-end created by the E-cclesia team. Unfortunately the E-cclesia system could not be completed in this time and so I had to proceed with using static data and without any external data sources. This made it very difficult to anticipate the structure of the application and the parameters and classes it may need to support.

Storing data to the phone: As the final evaluation was to take place over the course of an entire day to simulate a real election, users should be able to close the application without affecting the data on it, therefore the data should be saved to the phone. For example, if a user joins an organisation, the information about the organisation joined should be retained on the application even if a user closes the application and opens it again later. I solved this issue by utilising “Shared preferences”[12], a storage mechanism in Android that allows applications to store and retrieve simple data in key-value pairs.

State dependant design features: Multiple design features in the application were dependant on the state of the election or the ballot state. Although the features might be almost identical, a new XML (Extensible Markup Language) file had to be created for each difference. For example, a different XML file[15] has been created for each step of the ballot timeline. In addition, this state of each election had to be consistent throughout each feature in the system which required lots of data to be transferred between features. This was challenging to figure out and also to keep track of because of the large number of design components used.

Chapter 8

Final evaluation

8.1 Aims

This study aimed to measure the usability of the application and the final design of the interface with the average user, not HCI experts. Unlike the fast feedback sessions, in addition to the qualitative data obtained from the think-aloud sessions, I will also be evaluating the usability of the application with quantitative metrics.

8.2 Participants

This study involved 12 participants from the University of Edinburgh, 7 of which were informatics students and 5 studied other subjects, ensuring I would get be able to gather a majority of the usability issues both categories of users would face[67]. A full detail of the participants can be found in Appendix N. These participants were recruited through the School of Informatics mailing list and word of mouth. The main requirement was to have an Android phone and the reward upon completion was £10 compensation. To register interest in participating, students had to fill in a form used to help me recruit participants who would be more representative of the general population, excluding age and educational background as most of the student population are between 19-24 years of age and are currently pursuing higher education. In particular, I wanted to recruit both informatics students and students from other schools to see if this affected their opinion and understanding of the system.

8.3 Procedure

Meetings with each participant were scheduled beforehand in rooms on central campus of the University of Edinburgh to ensure privacy. The script used for each session can be found in Appendix K. The study was split into 3 sections for each participant:

1. **Voting:** At the beginning of the session, the participant would read the Paticipants Information Sheet (PIS), (found in Appendix A) and fill in the consent form (found in Appendix B). The participant would then be given more details about the project, how the study would be conducted, and an explanation of how to participate in a think-aloud. After installing the application onto their phone using Android studio[3] (screenshots found in Appendix J, the think-aloud began and the emails (found in Appendix L), were given to the participant with its corresponding task. Once the participant completed all the tasks, they filled in the questionnaire about their opinion of the voting system of the application (found in Appendix M.0.1), and follow-up questions may be asked. Before participants left, they were told about the next step (the verification step) of the user study.

2. **Verification part 1:** Participants were asked to check the application at their convenience until the next meeting to check that their vote had been recorded. Each time they checked the application, they were instructed to send an email of the elections and their state to me so I could observe their checking habits. If any changes occurred on the application, they were advised to follow the actions. Before leaving, 6 of the participants were given basic information about the need for verification, and 6 of the participants were given a more detailed explanation of its significance.

3. **Verification part 2:** In the evening, another meeting in a private room on the campus of the University of Edinburgh grounds took place so that the participant could complete the second questionnaire about the usability of the verification system (found in Appendix M.0.2). Again, follow-up questions may be asked. The study then concluded with participants receiving the £10 compensation for volunteering for the study.

8.4 Results

8.4.0.1 Think aloud

Similar to the previous fast feedback session, below is a summary of the results categorised by task, where positive feedback has been marked with a "P" and negative feedback/improvements have been marked with an "N".

Setting up

N1 *Unexpected welcome pages.* 5 participants expected to be able to use the application straight away instead of the series of welcome pages. This is possibly because task 1 implies that the user can begin using the application's features immediately.

N2 *Too many welcome pages.* P1 felt that there were too many welcome pages: "I wasn't expecting this many pages before I get to the actual thing. I probably won't remember all of that." Other participants often only read all the first text on the first page and only the large text on the subsequent pages, indicating that they had already lost attention.

P1 *Appreciated instructions before tasks began.* 2 participants mentioned that having some information in the beginning through the welcome pages was reassuring even if they won't remember all of it.

N3 *Expected to use default camera.* 2 participants first tried to use the default phone camera to scan the QR code instead of the button on the applications because this was their regular approach to scanning QR codes. When they found out the application had its own scanner, they liked this feature.

N4 *QR scanning may not always be possible.* Though all participants were able to successfully scan all the QR codes, P2 had a slower phone which took more time than usual to scan the code. Scenarios in which the QR scanner functionality may not work at all are also very possible.

N5 *Unexpected process to join an election.* 7 participants first tried to join an election through the organisation home page or through the organisation's information page because the difference between an organisation and an election was not clear and the order of joining was not clear: (P11 said) "the election and organisation pages look like they are both at the same level but the election is actually dependant on the organisation." Even some participants who understood there was a difference between an organisation and an election expected a different joining process: (P9 said) "so I need to add an election, maybe it's not on this page [organisation home page]? I'm not sure, but I will try to add it through this page".

Vote casting

N6 *Failed to understand need for wait.* P1 did not understand why they needed to wait before a certain time to begin casting: "I didn't think it was that complicated but I didn't understand why I needed to wait for so long."

N7 *Wanted pictures.* P11 suggested including pictures as a visual aid: "In an election, you can have people who are not really familiar with the names of the candidates but do know which party they want to vote for and their logo."

N8 *Wanted selected option on dialog box.* 5 participants stated they would have had more confidence in their actions if the option they had selected was visible in the dialog box.

Vote verification

P2 *Error was clearly indicated.* All the participants agreed that it was very easy to see that there was an error and that they should take an action: (P8 said) "there was a cross and it is in red so I could see that there is a problem."

N9 *Unanswered questions about verification.* Though all the questions participants had about verification were anticipated and already in the help page, most participants did not explore this page unless suggested to. For example P12 asked: "How do I participate in verifying votes? How long will it take to verify a vote?".

N10 *Wanted notifications.* 4 participants mentioned that they would have been more confident if there was a notification when there was an update on the vote verification or at least only when there was an error.

Viewing results

P3 *Clear results shown.* 4 participants mentioned that the results page was very clear and that they liked the contrasting colours: (P3 said) "I like the pie chart and the percentages and I could still see what I voted for which was good."

N11 *Wanted to see exact number of votes.* 2 participants wanted to be able to see the exact number of votes for each option in the election.

8.4.0.2 Effectiveness

In Table 8.1, setting up, and specifically joining the first election was the biggest challenge for participants, which can impact their ability to complete subsequent tasks and their opinion of the system. Participants struggled with understanding the difference between organizations and elections, and even those who understood had issues. All participants joined the second election successfully, indicating that this step is easy to learn, however, given the importance of this step, eliminating the initial confusion is crucial. This can be done by aligning the process to users' mental models, such as allowing users to join elections through the organisations information page.

Task	Success Rate
Setting up	50%
Vote casting	100%
Vote verification	100%
Viewing results	100%
Average	88%

Figure 8.1: Lists of tasks in the final evaluation and their corresponding success rates.

The other point of confusion in setting up was joining an organisation. P1 failed the task because they expected to immediately join an election. This is a result of inaccurate mental models and can also be resolved by restructuring the organisation and election pages. Two other participants failed because they expected to scan the QR code with the normal phone camera instead of through the application. Once they found out the application had its own scanner, they liked this feature however future implementations should support both methods.

Notably, all the participants successfully verified their vote and recast their vote when an error occurred. This is an improvement to the usability of voting system like Helios in which only 43% of participants performed any type of vote verification [20]. In fact this step was described as straightforward: "(P4) Usually you have to click a bunch of times or go to many different places like a maze to verify but this was simple to use.".

8.4.0.3 Satisfaction

The SUS rating (out of 100 possible points) for the voting interface is 82 and for the verification interface, it is 85. Thus, using the SUS ranking system developed by Bangor et al.[24], the usability of the entire application can be considered "Good" and the verification interface is also just in the "Excellent" range. This is significantly higher than any of the e-voting systems evaluated by Acemyan et al.[20] and the traditional voting systems evaluated by Byrne et al[30], suggesting that the users satisfaction with the current application is already at a good level, though of course can be improved further.

The SUS rating for the verification interface for participants who were given no extra information about the process was 84 and for participants who were given extra information, the SUS rating was 85. These almost equivalent scores suggest that the current information presented on the application and the way the interface is designed is enough to motivate and guide users to verify their votes successfully.

As can be seen in Figure 8.2, the SUS scores of the usability of both the voting and verification interfaces are above 85 and so the usability is "excellent"[24] for non-informatics students. For informatics students, the SUS score of the usability of both interfaces is just above 80 and so the usability is "good"[24] for them. This suggests that perhaps the current level of transparency on the application is not satisfactory for users with more technical knowledge and so the option of having more information for users who want it may be beneficial.

The full table of results of the SUS scores can be found in Appendix O.

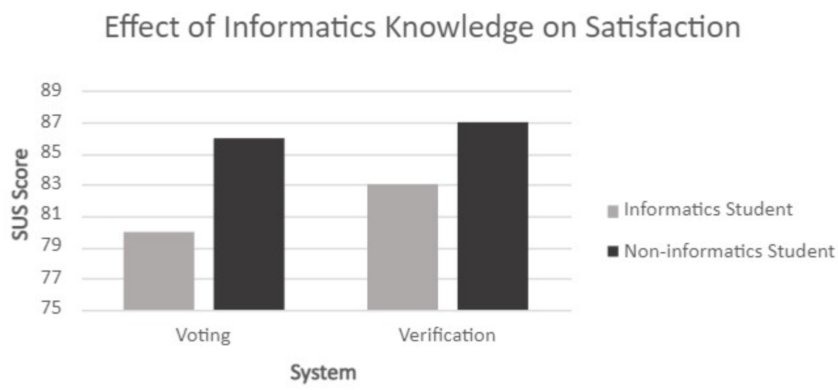

Figure 8.2: Comparison of SUS scores of the voting and verification systems for informatics and non-informatics students.

8.4.0.4 Trust

The average score for the further questions focusing on how much users trust the application during the voting stage is 4 (out of 5 or equivalent to agree), thus showing users already trust it. The average score for the further questions on if the verification step increased user's trust is also 4, thus the step did increase trust. P10 explained this was because "you can feel confident that the vote actually went somewhere and was not lost". For the participants who answered that the effect of the verification step was 3 (neutral/indifferent), they stated that this was because they already trusted the system beforehand, and so this step had no effect on their trust. This demonstrates that in the current implementation of E-cclesia, while the verification step may not increase the trust of all users, it is unlikely to negatively impact the user's trust in the system, even when a vote not recorded successfully and a user must recast.

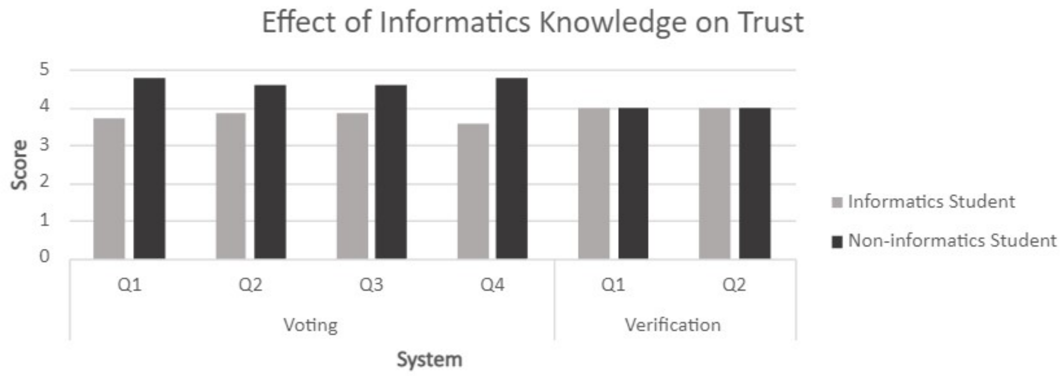

Figure 8.3: Comparison of user's trust in the system during the voting stage and after the verification stage between informatics students and non-informatics students. The list of questions can be found in Appendix M.0.1 and M.0.2.

The average scores' for the questions about trust after verification was 4 for both participants who were given extra information about verification and those who were not. Again, this suggests that the information provided on the application was sufficiently transparent. Moreover, Figure 8.3 shows that, although informatics students did not trust the application as much as non-informatics students, after the verification step, both groups equally trusted the system more because of this step. This shows that the current application can gain the trust of users who are more sceptical because of their background in technology, as well as those who do not have a technical background and are more trusting.

The full table of results of the extra questions about the system can be found in Appendix P.

8.4.1 Discussion

Generally, both the voting interface and verification interface can be considered equally usable. In terms of user satisfaction, it is certainly more usable than previously evaluated e-voting systems[20] and traditional voting systems[30]. It is also reassuring to see that the current information in the application is sufficient enough to motivate all users to perform the verification step. Notably, user's background knowledge of informatics does appear to affect their perception of the security of the application and level of trust. Though these users do still trust the application, especially with the inclusion of the verification step, optional further information would be advised.

While the unfamiliar step of verification was completed successfully by all participants, it was surprising to see that 7 participants failed to complete the setup without intervention. This low success rate is alarming and the interface should be changed to align with a users mental model and make the hierarchy of organisations and elections clear. This could be achieved by allowing users to add elections through an organisation's information page rather than a separate election home page as this is what most users who failed this step expected.

Error	Violated requirement	Suggested improvement
	Setting up	
N1: Unexpected welcome pages, N2: Too many welcome pages	P1, P2	Replace welcome page instructions with inline popup instructions the first time a user is using the application.
N3: Expected to use default camera	P1	Make the QR code scannable from default camera and the application.
N4: QR scanning may not always be possible	Accessibility issue	Include the option to join using a link (both clicking the link and typing the link into the application).
N5: Unexpected process to join an election	P1	Change format so elections are joined through an organisation and remove seprate election home page. Include an explanation of the difference between an organisation and election in the emails that have the joining QR code with screenshots demonstrating how to join.
	Vote casting	
N6: Failed to understand need for wait	P4	Include explanation on why users cannot begin voting straight away in help page and link this in the election timeline.
N7: Wanted pictures	L3	Include the option to add pictures in the admin panel and if an election includes pictures, put the pictures beside the text for each option on the voting page.
N8: Wanted selected option on dialog box	L5	Include the option the user selected in the dialog box before the final vote is cast.
	Vote verification	
N9: Unanswered questions about verification	P4, P5	During a users first time using the application, include inline popup explanations where users have the most questions.
N10: Wanted notifications	I3	Conduct a study with the affect of an application with notification and application without to see if this improves usability.
	Viewing the results	
N11: Wanted to see exact number of votes	L5	Include the number of votes for each election option in the results page.

Table 8.1: Feedback of possible errors/areas of improvement along with the design requirement (from section 4.3) which the error violates and how this error can be resolved. As per the critical incident analysis method outlined in Section 3.3,It does not include every incident observed throughout the study.

Chapter 9

Conclusions

The goal of gathering the functional requirements for the admin panel and voter application was successfully achieved. Through interviews with the E-cclesia team, detailed requirements and a user flow was gathered. Furthermore, I was able to gather sufficient design requirements through literature analysis that may have been overlooked otherwise.

The goal of designing and implementing the voter application was partially achieved. Two iterations of designs were made in Figma[6] using expert feedback. However, the implementation of the application is not currently connected to the real E-cclesia system as it is still under development and so the application had to be made using static data.

The goal of evaluating the usability of the implemented voter application was mostly achieved. A comprehensive user study was conducted using methods and metrics observed in previous user studies popular in the usable security research community. Though the usability of the interface can be considered at least ”Good” and trustworthy, too many failures occur ed in the joining stage which needs to be resolved. These results are useful, an ideal evaluation would have included the real E-cclesia system.

9.0.1 Limitations

One of the issues I faced during this study was finding participants for the fast feedback sessions. Though the HCI experts recruited for the two sessions gave comprehensive feedback, it would be more optimal to have feedback from 5 different HCI experts as this has been proven to be the standard number at which a majority of user issues will be discovered[67].

In the final evaluation, though I recruited a fairly diverse range of participants in terms of subject of study, voting experience, and ethnicity, all the participants were still students at the University of Edinburgh. Therefore, all participants would have a good level of education, English language skills and be use technology on a daily basis. This is not representative of the whole population.

In the studies carried out, the wording of the scripts could also have been clearer. Particularly in the first feedback session, the tasks were not clearly defined enough which lead to confusion for the participant. This could have affected their ability to use the interface successfully. Fortunately, with each study I conducted, the script was more clearly written however this may have affected the observations in the fast feedback sessions.

In the final evaluation, as the study took place across the whole day, it was very challenging to coordinate meeting times with all the participants. Therefore, the time between the first meeting and the second meeting with each participant was not the same. That is, participants experienced different duration's of time to verify their vote. Participants who had a shorter time between the two meetings will therefore have fewer chances to check the application and could have had higher chances of not verifying their vote (though fortunately this did not happen).

9.0.2 Future work

An important area of future work is creating the admin panel for the E-cclesia voting system. Ideally, the admin panel should be designed using a similar methodology to this project; gathering design requirements, getting feedback from experts, and then a user study with participants who represent the potential users of the system.

One of the issues discussed in the interviews with the E-cclesia development team that came up but was unresolved was how to ensure that the QR code emailed to specific users to become an eligible voter can remain secure. At the moment, these emails can be forwarded and voters who should be ineligible may be able to join the electoral roll. A solution to this problem should be explored.

In this study, I focused on the effect of how much verification information motivates users to verify their vote to indicate how much information should be presented and how. I chose this because verification is currently one of the biggest usability issues of existing voting systems. There are also more factors in addition to verification information that could affect usability that should be explored:

- As mentioned in the limitations, the participants in each study had a different period of time between their first and second meeting. This could affect the usability, for example if the election casting phase is too short perhaps users will not have time to verify their vote, or if the casting period is too long, users may forget to verify their vote. This effect should be evaluated to identify an ideal duration for the casting period to be suggested for election authorities when setting up an election.
- This study did not incorporate push notification as suggested by the E-cclesia team in Section 4.2.2. However, given that participants frequently gave feedback that this would be a useful feature, the effects of push notifications on usability should be studied.
- Although unlikely, there may be instances in which the same user's vote could not be recorded more than once in a row. The effects of this on the usability of this system and the user's motivation to continue using the system could be studied.

Bibliography

[1] Android. https://www.android.com/intl/en_uk/.

[2] Android jetpack. https://developer.android.com/jetpack.

[3] Android studio. https://developer.android.com/studio.

[4] Androidx. https://developer.android.com/jetpack/androidx.

[5] Condorcet Internet Voting Service (CIVS). https://civs1.civs.us/.

[6] Figma. https://www.figma.com/.

[7] Google. https://www.google.com/.

[8] Helios Voting. https://vote.heliosvoting.org/.

[9] Material design. https://m2.material.io/develop.

[10] Material design for android. https://developer.android.com/develop/ui/views/theming/look-and-feel.

[11] Personas. https://www.usability.gov/how-to-and-tools/methods/personas.html.

[12] Save key-value data. https://developer.android.com/training/data-storage/shared-preferences.

[13] Tulips: Trusted user interfaces for loss-prone systems. https://groups.inf.ed.ac.uk/tulips/.

[14] University of edinburgh human computer interaction course information page. http://www.drps.ed.ac.uk/22-23/dpt/cxinfr11017.htm.

[15] XML (Extensible Markup Language). https://www.techtarget.com/whatis/definition/XML-Extensible-Markup-Language.

[16] Kenya election: Turnout under 34% amid opposition boycott. *BBC*, 27th October 2017.

[17] More than 40% in us do not believe biden legitimately won election – poll. *The Guardian*, 5th January 2022.

[18] Aave Governance. Aave snapshot. https://snapshot.org/#/aave.eth.

[19] Claudia Z Acemyan, Philip Kortum, Michael D Byrne, and Dan S Wallach. From error to error: Why voters could not cast a ballot and verify their vote with helios, prêt à voter, and scantegrity {II}. {*USENIX*} *Journal of Election Technology and Systems (*{*JETS*}*)*, 3:1–25, 2015.

[20] Claudia Z Acemyan, Philip Kortum, Michael D Byrne, Dan S Wallach, Steve Schneider, and Vanessa Teague. Usability of voter verifiable, end-to-end voting systems: Baseline data for helios, prêt à voter, and scantegrity {II}. In *2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 14)*, 2014.

[21] Ben Adida. Helios: Web-based open-audit voting. In *USENIX security symposium*, volume 17, pages 335–348, 2008.

[22] Android Developers. Recyclerview. `https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView`.

[23] Myrto Arapinis, Nikolaos Lamprou, Lenka Mareková, Thomas Zacharias, Léo Ackermann, and Pavlos Georgiou. E-cclesia: Universally composable self-tallying elections. Cryptology ePrint Archive, Paper 2020/513, 2020. `https://eprint.iacr.org/2020/513`.

[24] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.

[25] Jonathan Bannet, David W Price, Algis Rudys, Justin Singer, and Dan S Wallach. Hack-a-vote: Security issues with electronic voting systems. *IEEE Security & Privacy*, 2(1):32–37, 2004.

[26] Benjamin B Bederson, Bongshin Lee, Robert M Sherman, Paul S Herrnson, and Richard G Niemi. Electronic voting system usability issues. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152, 2003.

[27] Belenios Project. Belenios e-voting system. `https://www.belenios.org/`.

[28] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[29] Yuriy Budiyev. Code scanner. `https://github.com/yuriy-budiyev/code-scanner`.

[30] Michael D Byrne, Kristen K Greene, and Sarah P Everett. Usability of voting systems: Baseline data for paper, punch cards, and lever machines. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 171–180, 2007.

[31] Nicholas Chang-Fong and Aleksander Essex. The cloudier side of cryptographic end-to-end verifiable voting: a security analysis of helios. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 324–335, 2016.

[32] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[33] Hwayoung Cho, Po-Yin Yen, Dawn Dowding, Jacqueline A Merrill, and Rebecca Schnall. A multi-level usability evaluation of mobile health applications: A case study. *Journal of biomedical informatics*, 86:79–89, 2018.

[34] Michael R Clarkson, Stephen Chong, and Andrew C Myers. Civitas: Toward a secure voting system. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 354–368. IEEE, 2008.

[35] Collins Dictionary. E-voting. `https://www.collinsdictionary.com/dictionary/english/e-voting`.

[36] Frederick G Conrad, Benjamin B Bederson, Brian Lewis, Emilia Peytcheva, Michael W Traugott, Michael J Hanmer, Paul S Herrnson, and Richard G Niemi. Electronic voting eliminates hanging chads but introduces new usability challenges. *International Journal of Human-Computer Studies*, 67(1):111–124, 2009.

[37] Brian J Corbitt, Theerasak Thanasankit, and Han Yi. Trust and e-commerce: a study of consumer perceptions. *Electronic commerce research and applications*, 2(3):203–215, 2003.

[38] Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-adapted interaction*, 18:455–496, 2008.

[39] Verena Distler, Marie-Laure Zollinger, Carine Lallemand, Peter B Roenne, Peter YA Ryan, and Vincent Koenig. Security-visible, yet unseen? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13, 2019.

[40] Oshima D.N. Building interfaces to vote: Development of various surfaces for e-cclesia. 2021.

[41] Sarah P Everett. *The usability of electronic voting machines and how votes can be changed without detection*. PhD thesis, Rice University, 2007.

[42] Sarah P Everett, Kristen K Greene, Michael D Byrne, Dan S Wallach, Kyle Derr, Daniel Sandler, and Ted Torous. Electronic voting machines versus traditional methods: Improved preference, similar performance. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 883–892, 2008.

[43] J Paul Gibson, Robert Krimmer, Vanessa Teague, and Julia Pomares. A review of e-voting: the past, present and future. *Annals of Telecommunications*, 71:279–286, 2016.

[44] Daniela Gotseva, Yavor Tomov, and Petko Danov. Comparative study java vs kotlin. In *2019 27th National Conference with International Participation (TELECOM)*, pages 86–89. IEEE, 2019.

[45] Kimmo Grönlund and Maija Setälä. Political trust, satisfaction and voter turnout. *Comparative European Politics*, 5:400–422, 2007.

[46] Jan Gulliksen, Bengt Göransson, Inger Boivie, Stefan Blomkvist, Jenny Persson, and Åsa Cajander. Key principles for user-centred systems design. *Behaviour and Information Technology*, 22(6):397–409, 2003.

[47] Bruce Hanington and Bella Martin. *Universal methods of design expanded and revised: 125 Ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport publishers, 2019.

[48] Paul S Herrnson, Benjamin B Bederson, Bongshin Lee, Peter L Francia, Robert M Sherman, Frederick G Conrad, Michael Traugott, and Richard G Niemi. Early appraisals of electronic voting. *Social Science Computer Review*, 23(3):274–292, 2005.

[49] Paul S Herrnson, Richard G Niemi, Michael J Hanmer, Benjamin B Bederson, Frederick G Conrad, and Michael Traugott. The importance of usability testing of voting systems. *EVT*, 6:3–3, 2006.

[50] International Institute for Democracy and Electoral Assistance (IDEA). Use of e-voting around the world. `https://www.idea.int/news-media/media/use-e-voting-around-world`.

[51] International Organization for Standardization (ISO). ISO 9241-11:2018 - Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts.

[52] Philipp Jay. Mpandroidchart. `https://github.com/PhilJay/MPAndroidChart`.

[53] JetBrains. Kotlin programming language. `https://kotlinlang.org/`.

[54] Fatih Karayumak, Michaela Kauer, M Maina Olembo, Tobias Volk, and Melanie Volkamer. User study of the improved helios voting system interfaces. In *2011 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST)*, pages 37–44. IEEE, 2011.

[55] Fatih Karayumak, Maina M Olembo, Michaela Kauer, and Melanie Volkamer. Usability analysis of helios-an open source verifiable remote electronic voting system. *EVT/WOTE*, 11(5), 2011.

[56] Dalia Khader, Ben Smyth, Peter Ryan, and Feng Hao. A fair and robust voting system by broadcast. *Lecture Notes in Informatics*, pages 285–299, 2012.

[57] Aggelos Kiayias and Moti Yung. Self-tallying elections and perfect ballot secrecy. In *International Workshop on Public Key Cryptography*, pages 141–158. Springer, 2002.

[58] René F Kizilcec. How much information? effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 2390–2395, 2016.

[59] Tadayoshi Kohno, Adam Stubblefield, Aviel D Rubin, and Dan S Wallach. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 27–40. IEEE, 2004.

[60] Frederic Lardinois. Kotlin is now google's preferred language for android app development. *Verizon Media*, 2019.

[61] Sharon J Laskowski, Marguerite Autry, John Cugini, William Killam, and James Yen. *Improving the usability and accessibility of voting systems and products*. US Department of Commerce, National Institute of Standards and Technology, 2004.

[62] Lumivero. Nvivo. `https://lumivero.com/products/nvivo/`.

[63] Epp Maaten. Towards remote e-voting: Estonian case. In Alexander Prosser and Robert Krimmer, editors, *Electronic voting in Europe - Technology, law, politics and society, workshop of the ESF TED programme together with GI and OCG*, pages 83–90, Bonn, 2004. Gesellschaft für Informatik e.V.

[64] Spenser Mestel. How bad ballot design can sway the result of an election. *The Guardian*.

[65] Stephanie A Morey, Rachel E Stuck, Amy W Chong, Laura H Barg-Walkow, Tracy L Mitzner, and Wendy A Rogers. Mobile health apps: improving usability for older adult users. *Ergonomics in Design*, 27(4):4–13, 2019.

[66] Jacob Nielson. 10 usability heuristics for user interface design.

[67] Jakob Nielson. Why you only need to test with 5 users. 18th March 2000.

[68] NIST. National Institute of Standards and Technology. `https://www.nist.gov/`.

[69] Donald Norman. User centered system design. *New perspectives on human-computer interaction*, 1986.

[70] U.S. Department of Health and Human Services. User-centered design. `https://www.usability.gov/what-and-why/user-centered-design.html`.

[71] Oracle Corporation. What is java? `https://www.java.com/en/download/help/whatis_java.html`.

[72] Mary Anne Patton and Audun Jøsang. Technologies for trust in electronic commerce. *Electronic Commerce Research*, 4:9–21, 2004.

[73] Whitney Quesenbery and Sharon J Laskowski. Handbook for vvsg 2.0 usability and accessibility test strategies. 2023.

[74] Gary H Roseman Jr and E Frank Stephenson. The effect of voting technology on voter turnout: Do computers scare the elderly? *Public Choice*, 123(1-2):39–47, 2005.

[75] Choe Sang-hun. Former south korean president park geun-hye facing possible life sentence. *The Guardian*, April 2017.

[76] Eva Schicker. What exactly are user flows in ux design? https://bootcamp.uxdesign.cc/what-exactly-are-user-flows-in-ux-design-62023c2370d6.

[77] Robert C Sinclair, Melvin M Mark, Sean E Moore, Carrie A Lavis, and Alexander S Soldat. An electoral butterfly effect. *Nature*, 408(6813):665–666, 2000.

[78] Ashok Sivaji, Alan G Downe, Muhammad Fahmi Mazlan, Shi-Tzuaan Soo, and Azween Abdullah. Importance of incorporating fundamental usability with social & trust elements for e-commerce website. In *2011 International Conference on Business, Engineering and Industrial Applications*, pages 221–226. IEEE, 2011.

[79] National Geographic Society. Democracy in ancient greece.

[80] Statista. Global market share held by mobile operating systems from 2009 to 2021, 2021.

[81] UXPortfolio. Happy path. https://uxportfolio.cc/ux-terms/happy-path/.

[82] Jonathan Watts. Operation car wash: Is this the biggest corruption scandal in history? *The Guardian*, 1st June 2017.

[83] Jonathan Watts. Bolivia in turmoil after evo morales wins fourth term as president. *The Guardian*, October 2019.

[84] J Weber and Urs Hengartner. Usability study of the open audit voting system helios. *Retrieved August*, 3:2012, 2009.

Appendix A

Participants' information sheet

Participant Information Sheet

Project title:	Mobile client for the E-cclesia electronic voting protocol
Principal investigator:	Myrto Aripinis
Researcher collecting data:	Sraddheya Gurung
Funder (if applicable):	

This study was certified according to the Informatics Research Ethics Process, RT number 617552. Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers?

The researchers of the study are Sraddheya Gurung who is an undergraduate student in the University of Edinburgh School of Informatics and Myrto Aripinis who is her supervisor.

What is the purpose of the study?

This study is part of an undergraduate project, which aims to design an app that implements a voting protocol developed by University of Edinburgh researchers called E-cclesia. The purpose of this study is to evaluate the designs of the prototypes in order to learn what the biggest faults are and how the design could be improved. This study also aims to learn about the usability of this design.

Why have I been asked to take part?

The aim of this project is to design an app to help the general population cast votes and create elections more securely. You may have a different, but still helpful suggestion or comment on how to make this app more user friendly.

Do I have to take part?

No – participation in this study is entirely up to you. You can withdraw from the study at any time without giving a reason. After this point, personal data will be deleted and anonymised data will be combined such that it is impossible to remove individual information from the analysis. Your rights will not be affected. If you wish to withdraw, contact the PI. We will keep copies of your original consent, and of your withdrawal request.

What will happen if I decide to take part?

We will let you use the app freely for a few minutes, while talking aloud about your experience with it. After you are done, we will ask you some questions about the usability and potential impact of the app and ask you to carry out some tasks on the app. This process shouldn't last longer than 30min and it will be recorded with your consent.

Are there any risks associated with taking part?

There are no significant risks associated with participation.

Are there any benefits associated with taking part?

No.

What will happen to the results of this study?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a maximum of 3 years. All potentially identifiable data will be deleted within this timeframe if it has not already been deleted as part of anonymization.

Data protection and confidentiality.

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher/research team: Sraddheya Gurung and Myrto Arapinis.

All electronic data will be stored on a password-protected encrypted computer, on the School of Informatics' secure file servers, or on the University's secure encrypted cloud storage services (DataShare, ownCloud, or Sharepoint) and all paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses in order to minimise risk.

What are my data protection rights?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.

Who can I contact?

If you have any further questions about the study, please contact the lead researcher, Sraddheya Gurung (s1908227@ed.ac.uk). If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact us, please provide the study title and detail the nature of your complaint.

Updated information.

If the research project changes in any way, an updated Participant Information Sheet will be sent by email to you by Sraddheya Gurung.

Alternative formats.

To request this document in an alternative format, such as large print or on coloured paper, please contact Sraddheya Gurung (s1908227@ed.ac.uk).

General information.

For general information about how we use your data, go to: edin.ac/privacy-research

Appendix B

Participants' consent form

Participant number:______________________

Participant Consent Form

Project title:	Mobile client for the E-cclesia electronic voting protocol
Principal investigator (PI):	Myrto Aripinis (marapini@exseed.ed.ac.uk)
Researcher:	Sraddheya Gurung (s1908227@ed.ac.uk)

By participating in the study you agree that:

- I have read and understood the Participant Information Sheet for the above study, that I have had the opportunity to ask questions, and that any questions I had were answered to my satisfaction.
- My participation is voluntary, and that I can withdraw at any time without giving a reason. Withdrawing will not affect any of my rights.
- I consent to my anonymised data being used in academic publications and presentations.
- I understand that my anonymised data will be stored for the duration outlined in the Participant Information Sheet.

Please tick yes or no for each of these statements.

		Yes	No
1.	I agree to being audio recorded.		
2.	I agree to being video recorded.		
3.	I allow my data to be used in future ethically approved research.		
4.	I agree to take part in this study.		

Name of person giving consent	Date	Signature
	dd/mm/yy	

Name of person taking consent	Date	Signature
	dd/mm/yy	

THE UNIVERSITY *of* EDINBURGH
informatics

Appendix C

SUS rating system

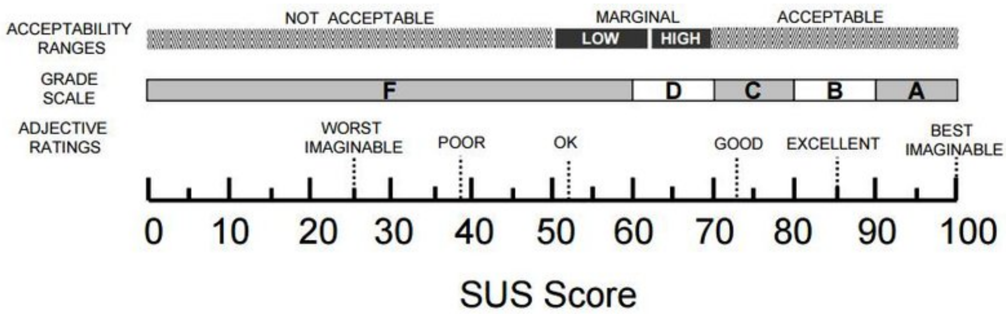

Figure C.1: A comparison of the adjective ratings, acceptability scores, and school grading scales, in relation to the average SUS score created by Bangor et al.[24].

Appendix D

Low fidelity wireframe

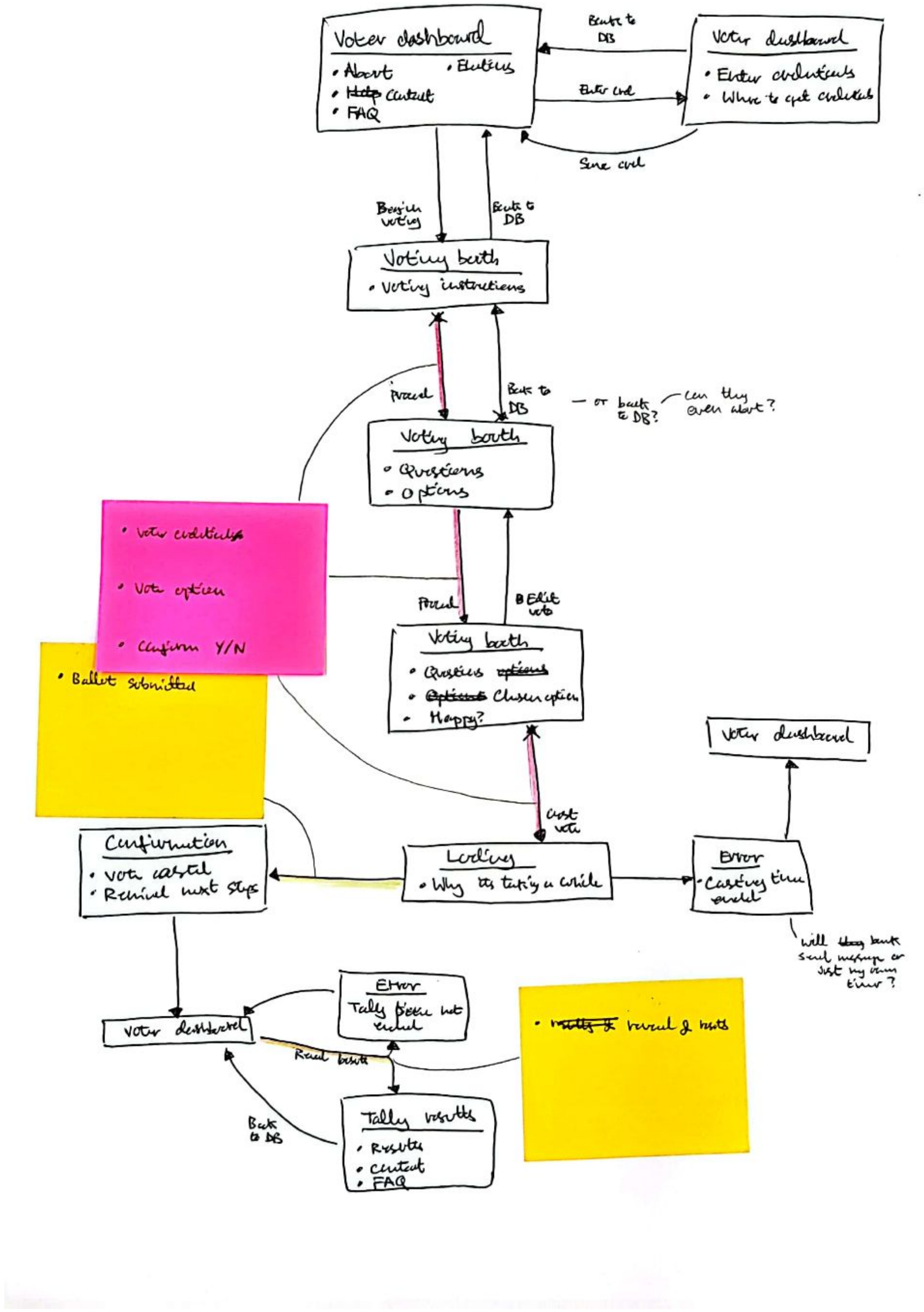

Figure D.1: First iteration of the voter application user flow.

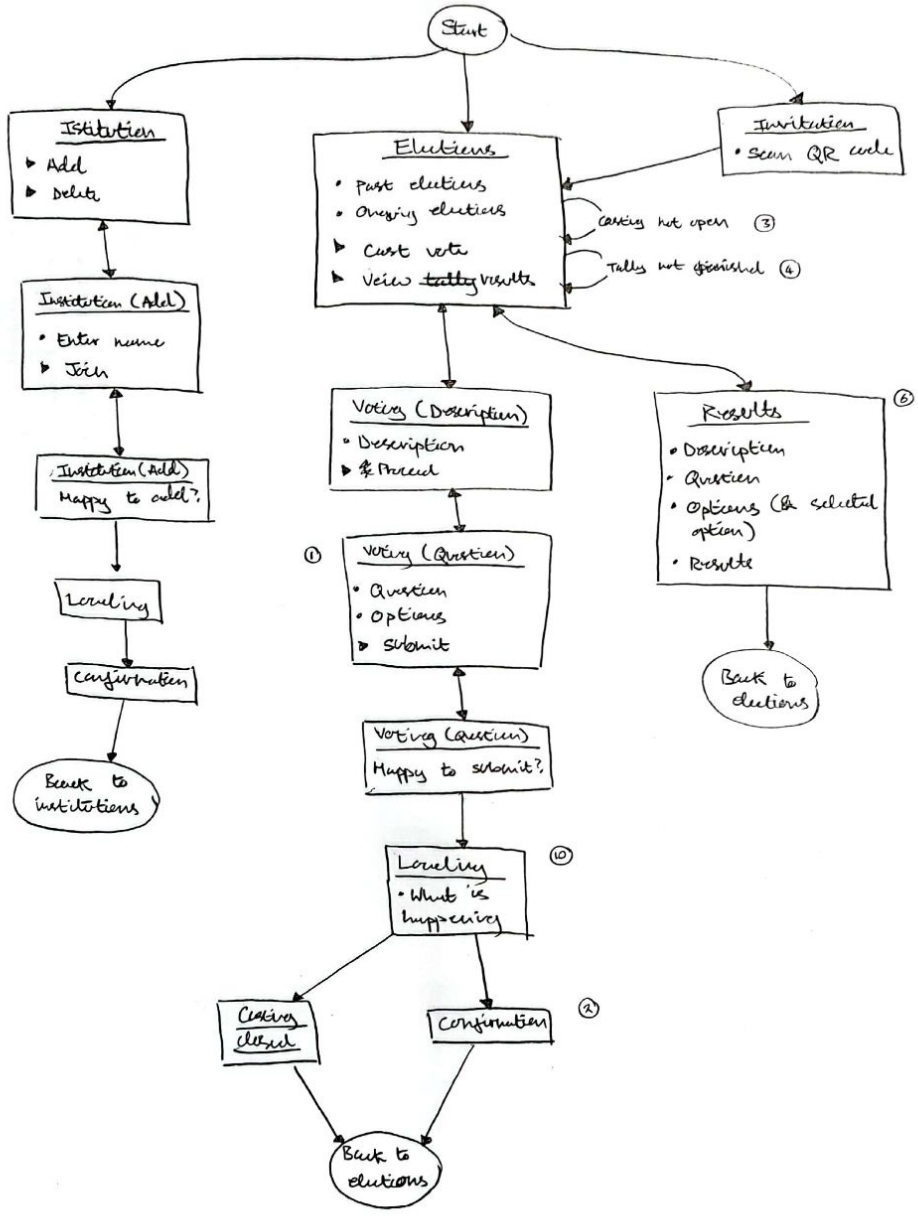

Figure D.2: Second iteration of the voter application user flow.

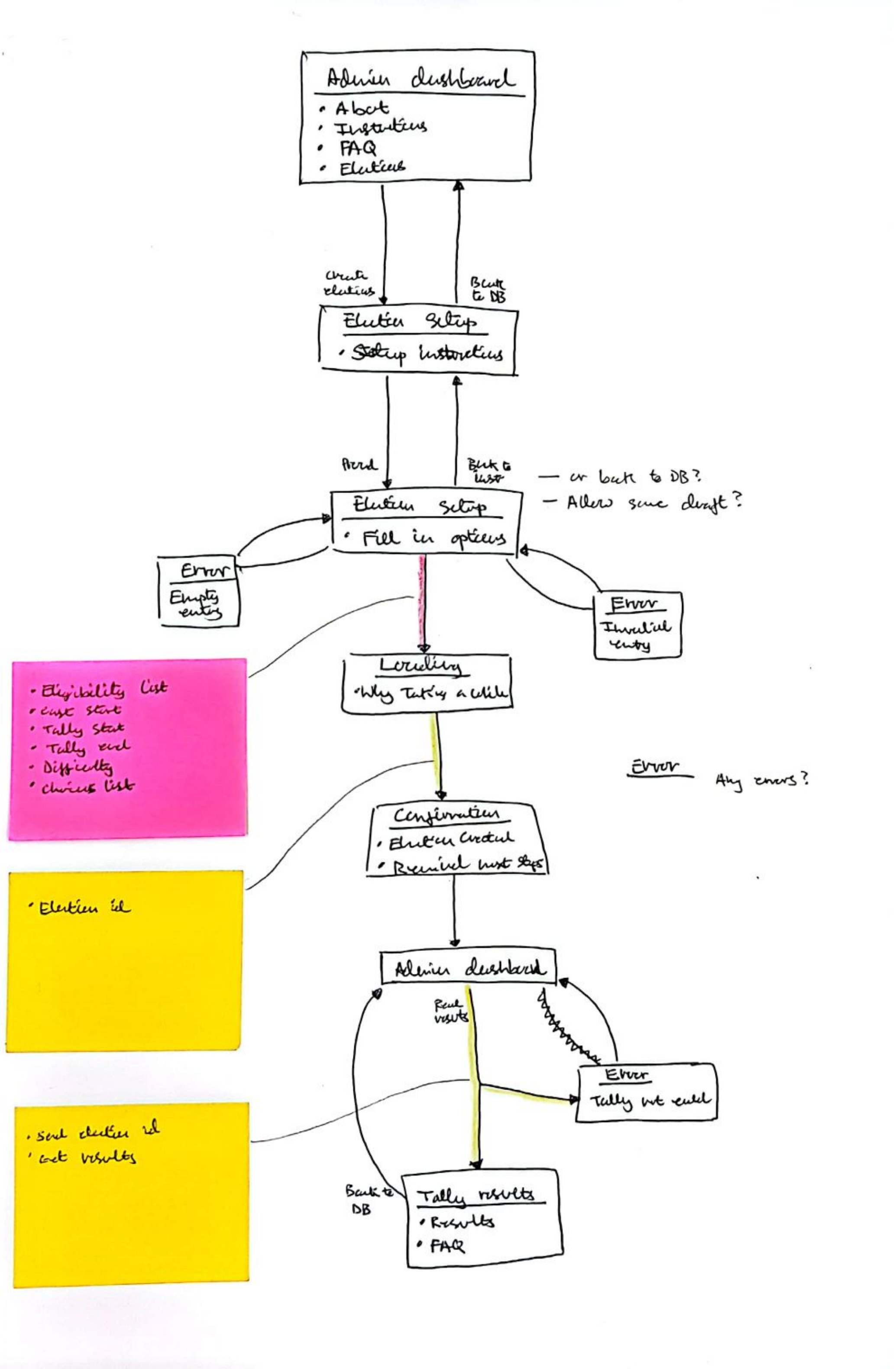

Figure D.3: First iteration of the admin panel user flow.

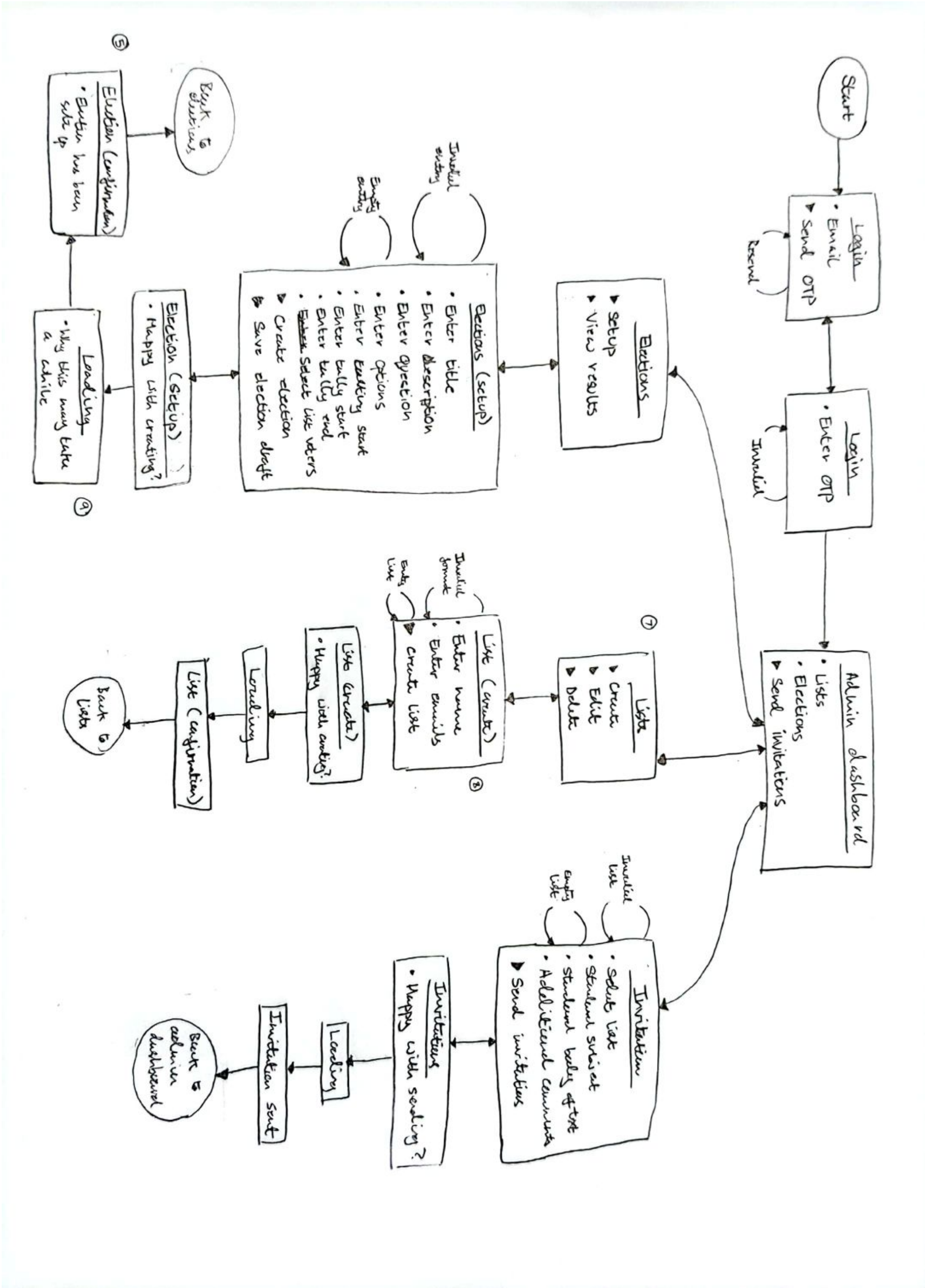

Figure D.4: Second iteration of the admin panel user flow.

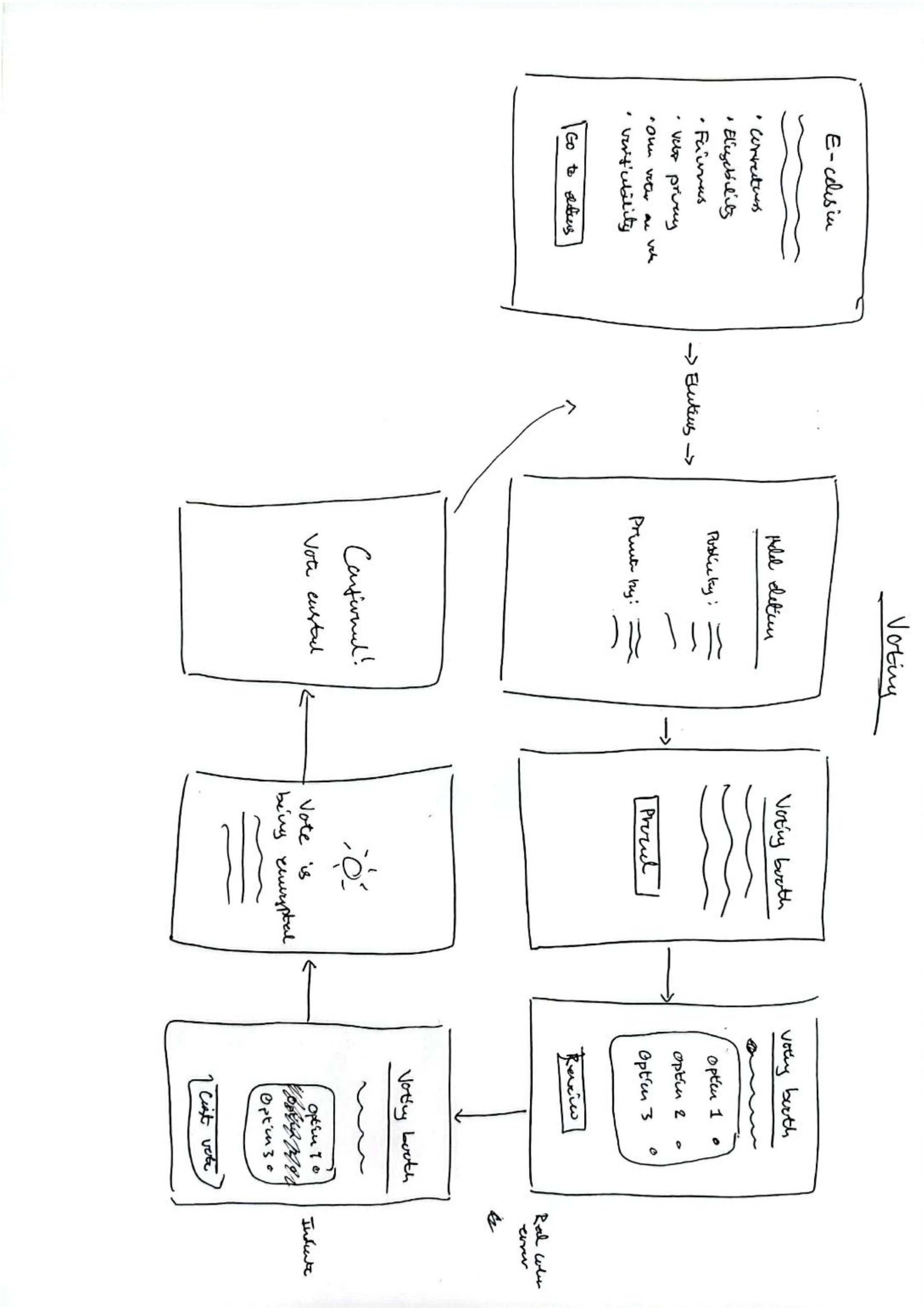

Figure D.5: First iteration of the voter application interface design.

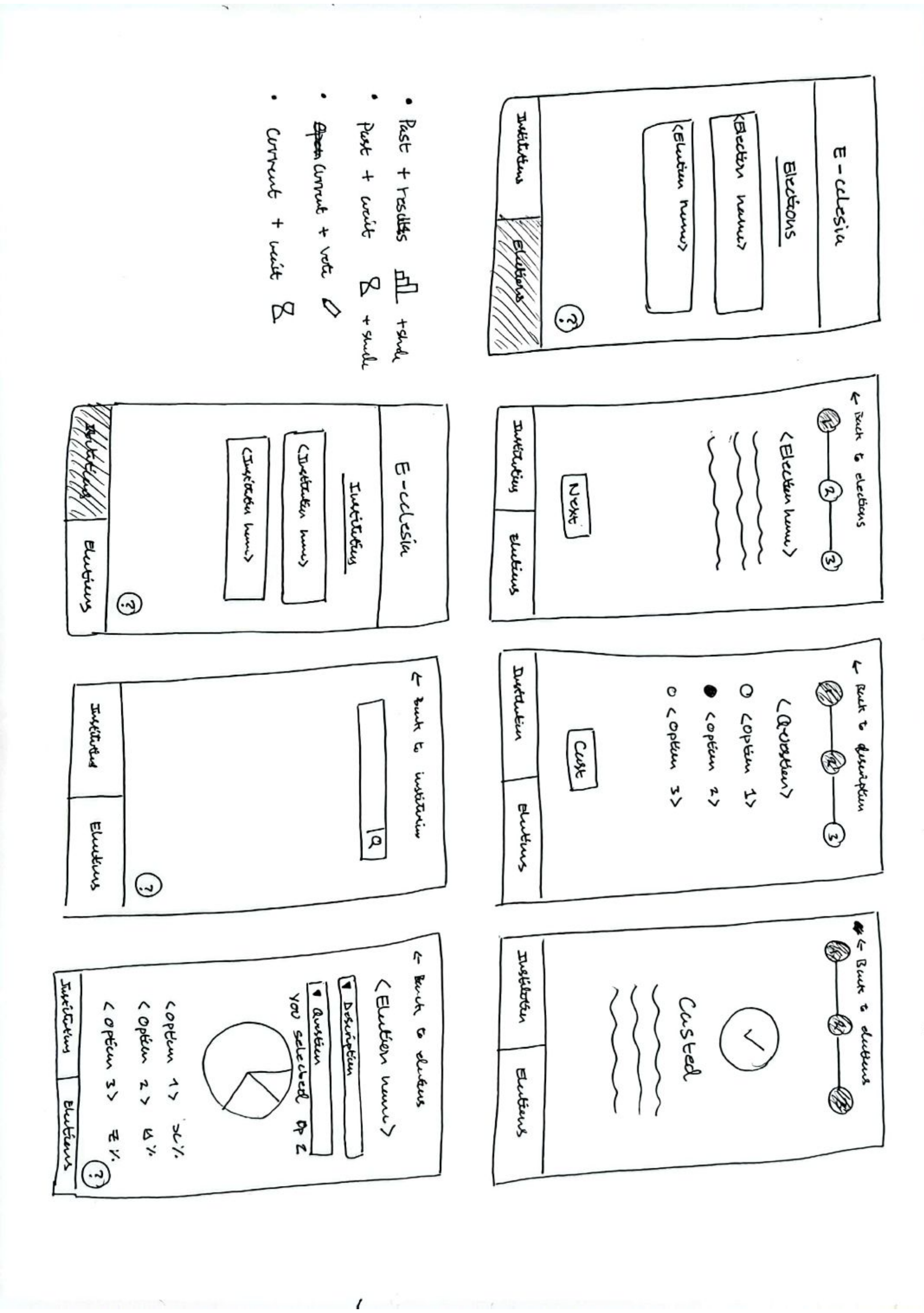

Figure D.6: Second iteration of the voter application interface design.

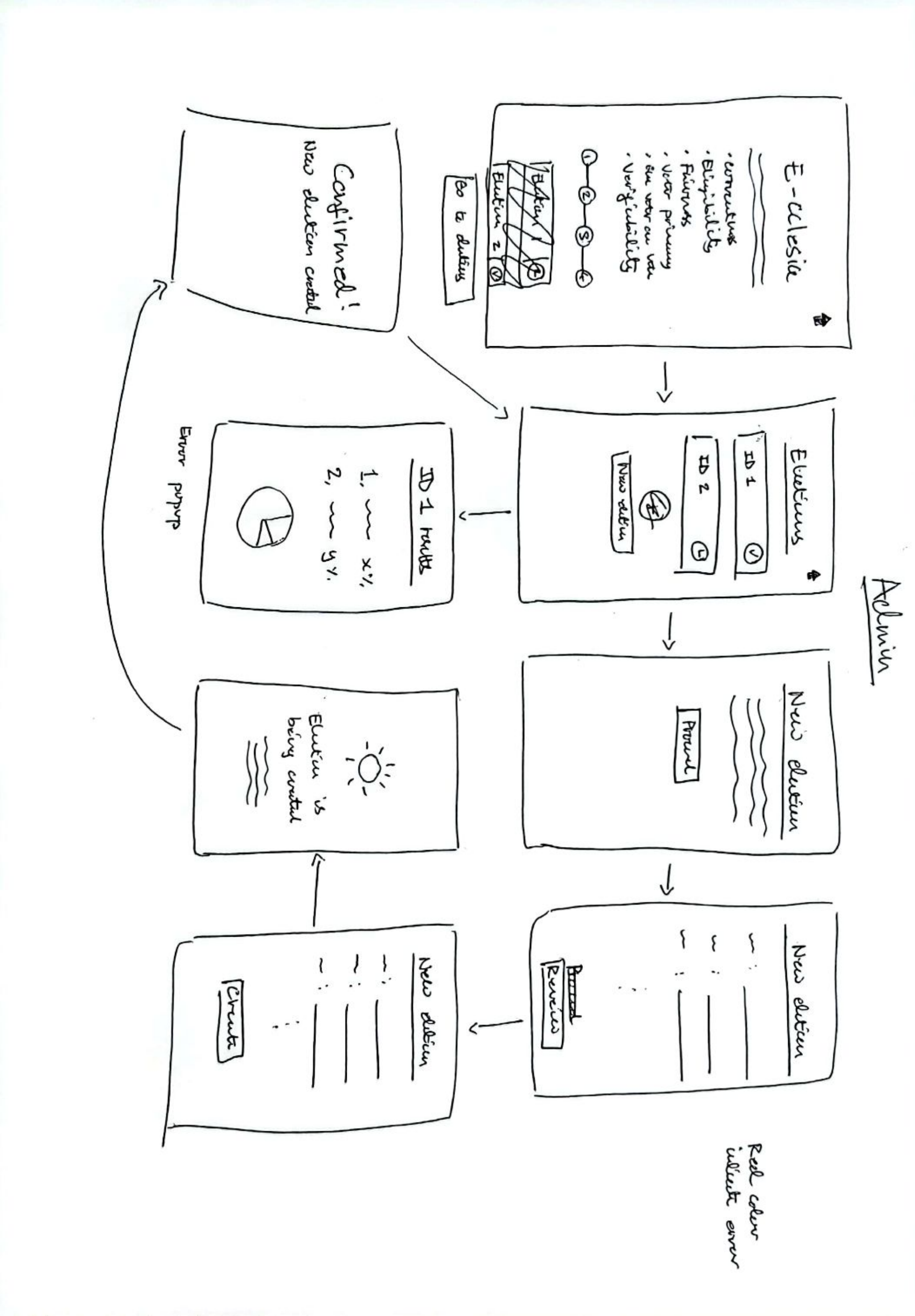

Figure D.7: First iteration of the admin panel interface design.

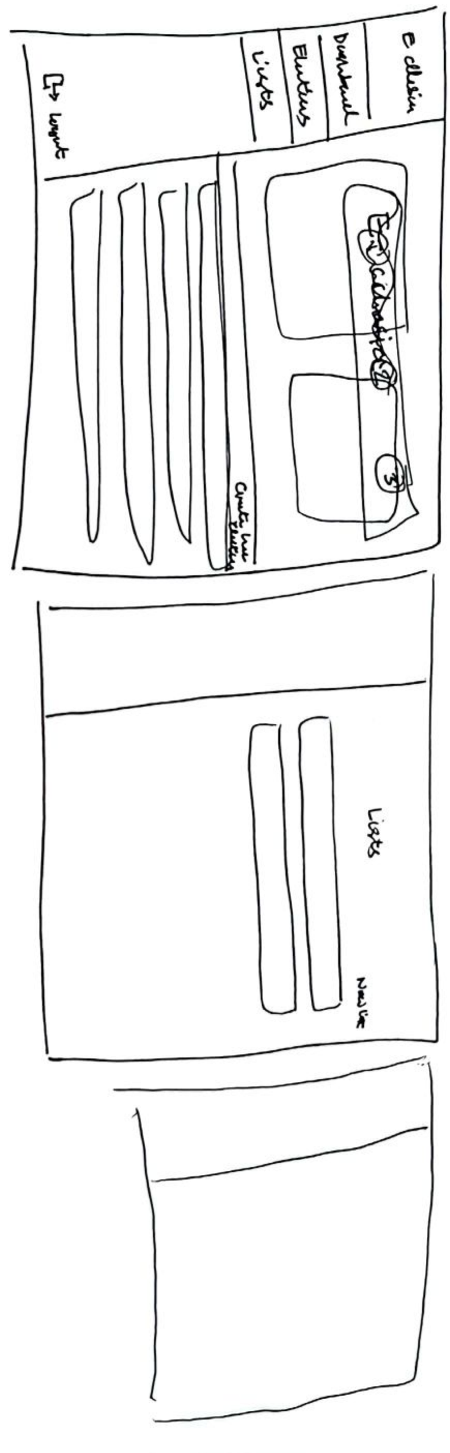

Figure D.8: Second iteration of the admin panel interface design.

Appendix E

User flows

Figure E.1: Key of each icon used in the flow diagrams.

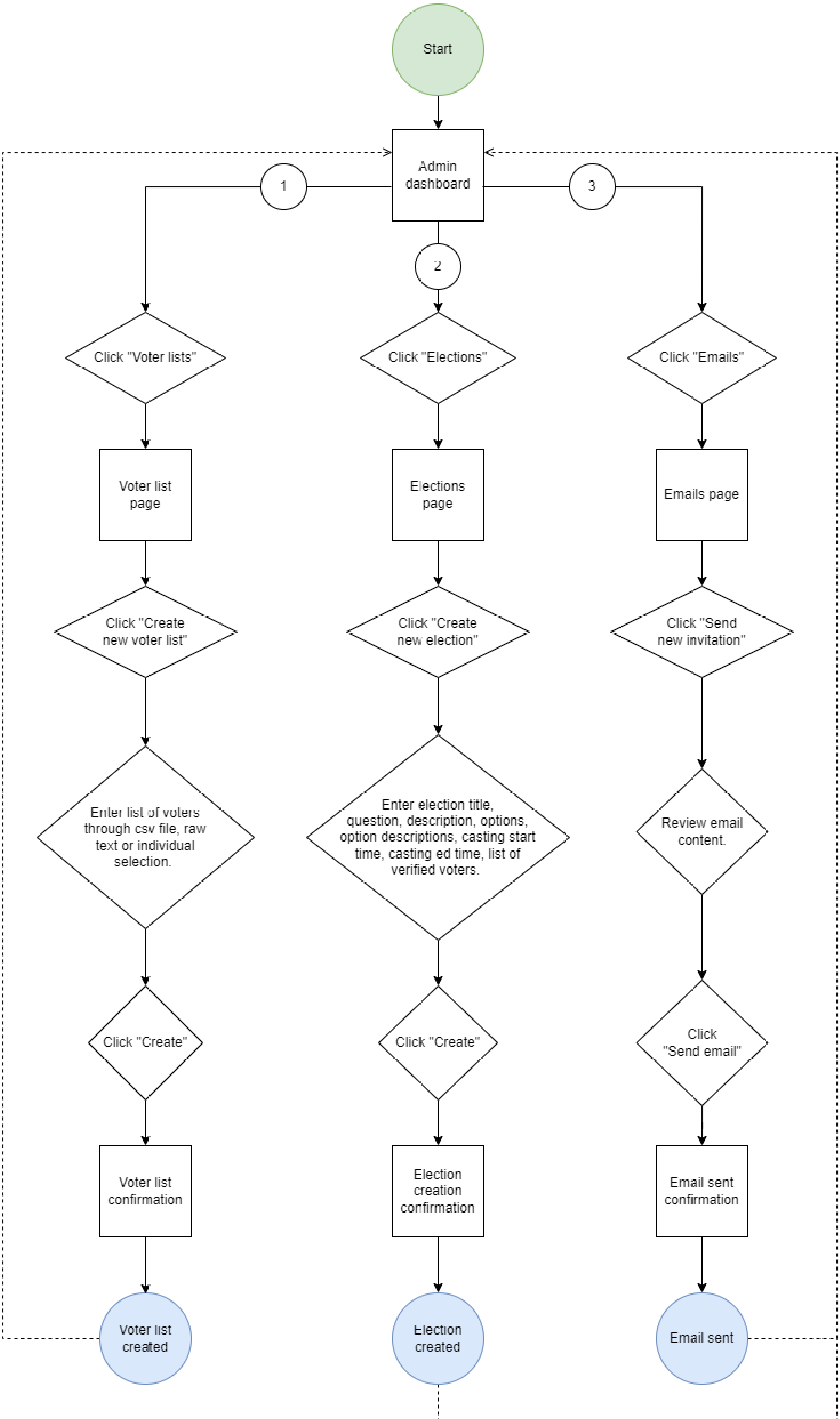

Figure E.2: Final user flow of the admin system.

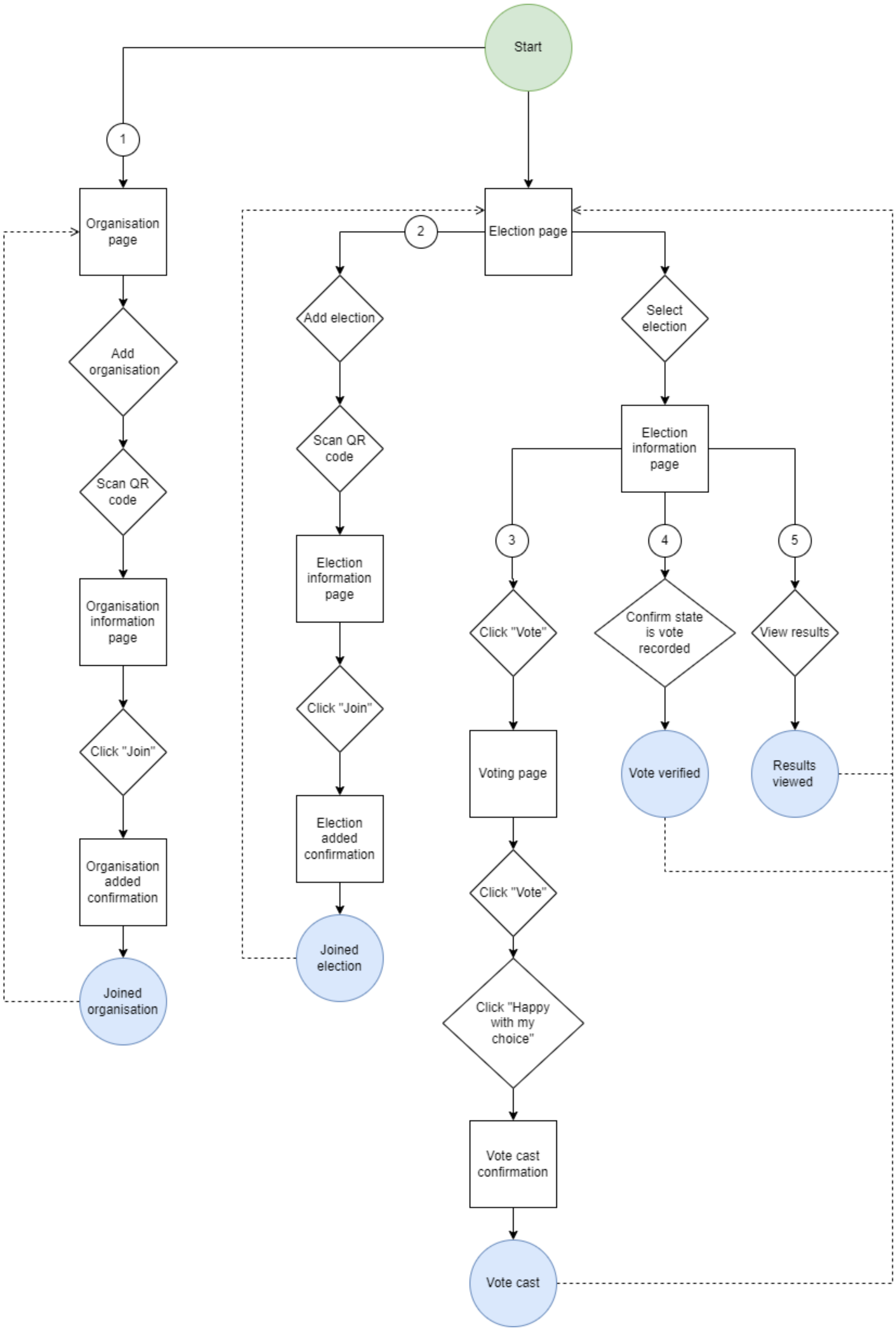

Figure E.3: Final user flow of the voter system.

Appendix F

Figma mock-up 1

Can be accessed online here: `https://www.figma.com/file/20T7h3a5cXyj4iArMY3htW/Voter1?t=goM37SWpjBquVjGN-1`

Appendix G

Fast feedback 1 script

Hello, my name is Sraddheya Gurung and, as part of my Honours project, I am creating a voting app and have designed some mock-ups on Figma of the voter mobile application. E-cclesia is a voting protocol created by researchers at the University of Edinburgh that allows for more secure voting. It does this mainly through the fact that it is decentralised. This means that all parties involved in the election can verify the tally of the votes, not one central party.

Today I will be asking you to perform a series of tasks using Figma to gather feedback on how usable the application is. This session will be run in the style of a think aloud, so as you go through each task, please articulate your thoughts about the interface, the navigation, the language, etc.

Scenario

- You are a student at the University of Edinburgh who wants to vote in the Computing society's elections to make sure you get the food and drinks you like the most at the next event.
- You want to make sure you vote for your favourites correctly, but you also have a lot of deadlines coming up and so you want to complete the tasks fairly quickly.

Tasks

1. Cast a vote for the election "Favorite Pizza Toppings". You will first vote for pepperoni but change your mind to Mushroom. Mushroom will be the vote option that you cast.
2. View the results for the election "Favorite Pizza toppings" and try to find out what the options were and if what the option you voted for was (to see if it matches what you think was cast).
3. Try to cast a vote for the election "Favorite colour".
4. Try to see the results of election "Election 2".

Appendix H

Figma mock-up 2

Can be accessed online here: `https://www.figma.com/file/dhhf6wpQa3hFXq7cqSeBUy/Voter2?t=goM37SWpjBquVjGN-1`

Appendix I

Fast feedback 2 script

Hello, my name is Sraddheya Gurung and, as part of my Honours project, I am creating a voting app and conducting usability studies. Before we begin, we can fill in the participant consent form.

Today, I will be asking you to perform certain tasks using a prototype of a voting app to assess how easy it is to navigate and to get feedback on other usability aspects of the design. This session will be in a think aloud style, where I will ask you to perform a series of small tasks and you can say your thoughts aloud as you complete the tasks. In particular I am very interested in finding out what you would have expected to happen when you pressed on a particular button or what you expected to see in the next page of the app.

Most of the tasks will be very short so don't be surprised if it is over in a few clicks. I will tell you the next task to carry out as the session progresses. I may also ask you a few questions along the way. If you have any questions along the way, please say them aloud as well so I can get more ideas of the thought process a user might have, although I may not answer all of them. Please remember I am testing the prototype, not you. Also, as this is a prototype, some of the buttons may not be functional but these will be indicated by the software or I can let you know. For now I will let you know that the help button is not functional but if you have any questions that would typically be on the help button, I might answer them.

Scenario

- You are a student at the University of Edinburgh who wants to vote in the Computing society's elections to make sure you get the food and drinks you like the most at the next event.
- You want to make sure you vote for your favourites correctly, but you also have a lot of deadlines coming up and so you want to complete the tasks fairly quickly.

Tasks

1. You have received an email from the Computing society with the QR code. Add the Computing society to the app as an organisation.
2. You have received an email from the Computing society with a QR code for their next upcoming election. Add the election to the app.
3. Vote for your favourite pizza topping in this election. For the prototype purpose, please vote for pepperoni as the other options are nonfunctional. *Casting period has not started yet).*
4. Now imagine some time has passed and the voting period has started. Please vote.
5. Verify that your vote for the pizza has been recorded. *(Vote has not been recorded yet)*
6. You have received another email from the Computing society with a QR code for their next upcoming election. Add the next election to the app.
7. Imagine that some time has passed, and the voting period of this election has started. Vote for your favourite drink in this election. For the prototype purpose, please vote for sprite as the other options are nonfunctional.
8. Check the pizza election to see if your vote has been recorded again.
9. Imagine some time has passed, please check the results for the pizza election.
10. Now check if your vote for the drink election has been recorded. *(Vote was not recorded)*
    - What page would you expect to be taken to?
    - What would you like to be able to do about this?
    - Has this affected your opinion of the security of this app?

Further questions

1. Which two aspects did you like most about the app? Why?
2. Which two aspects did you like least about the app? Why? How can this be improved?
3. Any other aspect that you would like to mention about the interface?
4. Would you recommend this voting app to others who need to vote electronically? Why or why not?
5. How does this compare to your experiences of voting apps?
6. How do you feel the security of this compared to other voting apps?
7. Have you voted before? Either electronically or using traditional methods like paper? If yes, was the flow in this app similar to what you experienced? If no, was the flow in this app similar to what you expected?
8. How clear are the steps, did you understand why you did each step?

Appendix J

Application screenshots

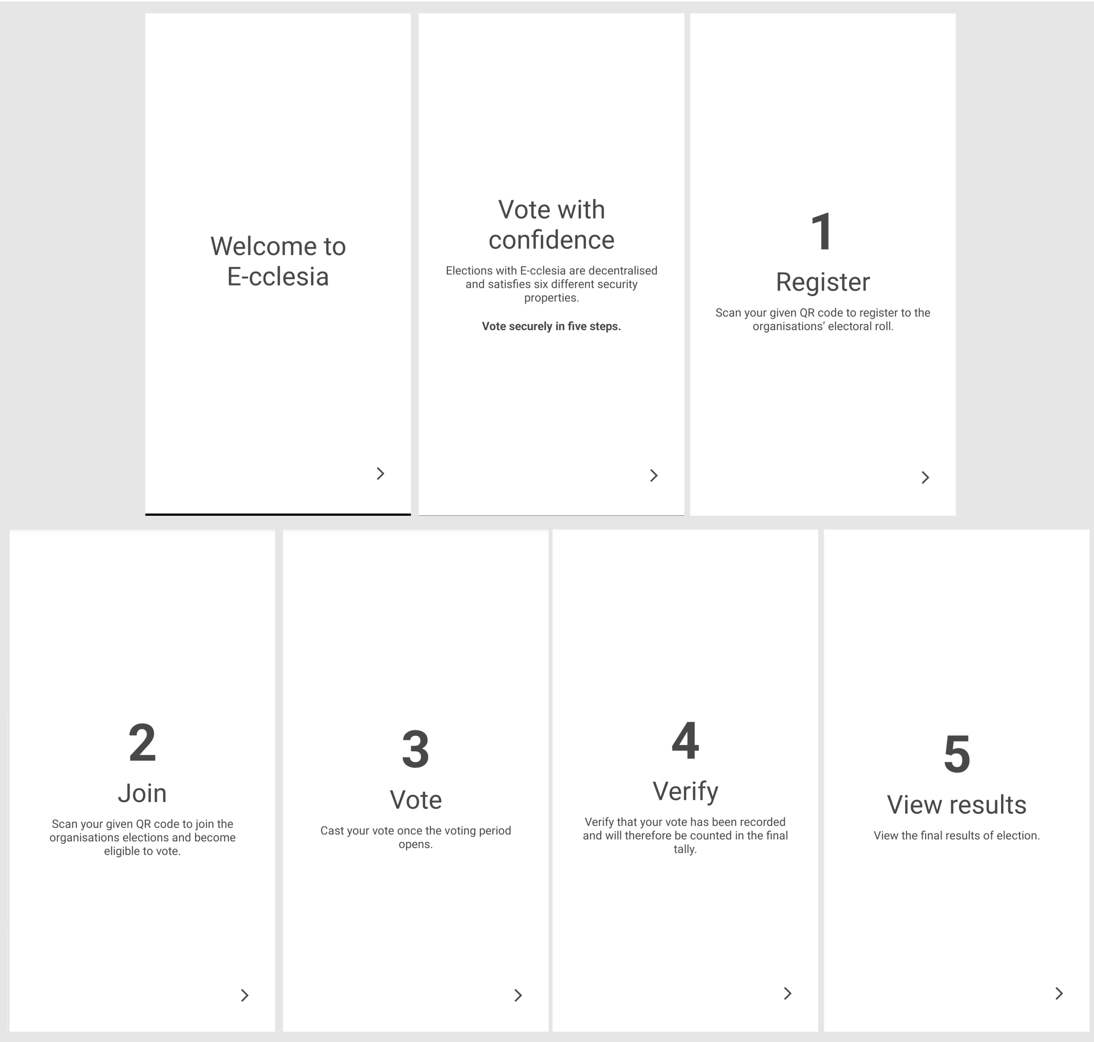

Figure J.1: Screenshots of the welcomes pages of the application used to give users some information about the process of the system before they begin.

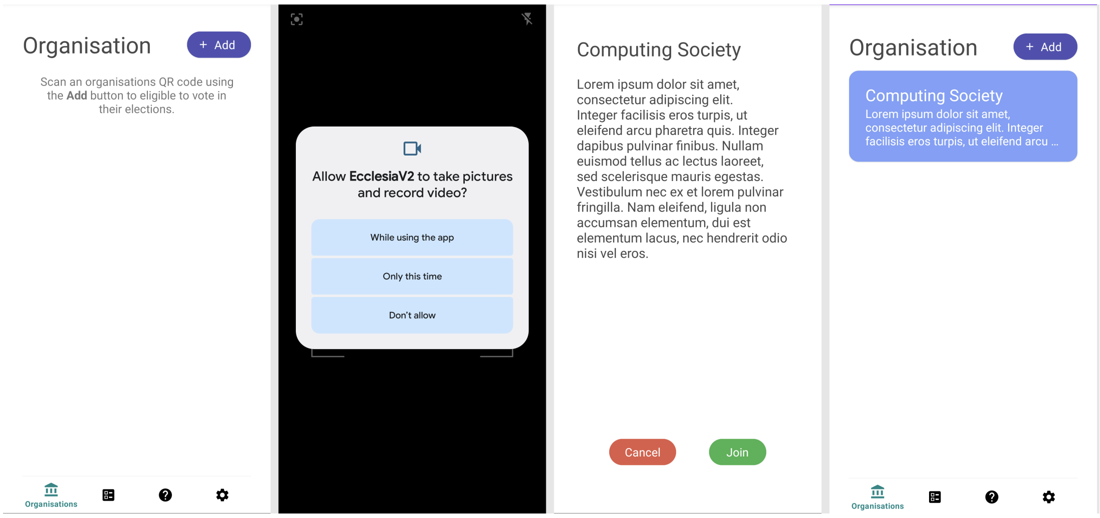

Figure J.2: Screenshots of the pages related to the organisation. From left to right: empty organisation home page, qr code scanner consent page, organisation joining page, and the organisation home page with one organisation item listed.

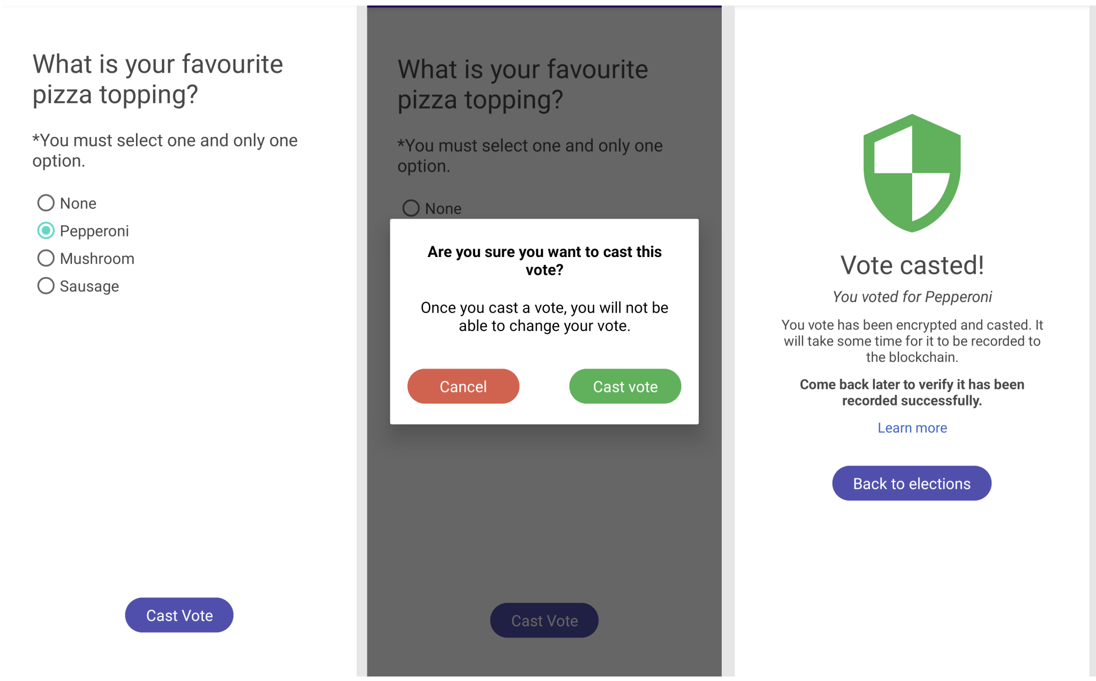

Figure J.3: Screenshots of the pages to cast a vote. From left to right: vote selection page, dialog confirmation box, vote cast confirmation page.

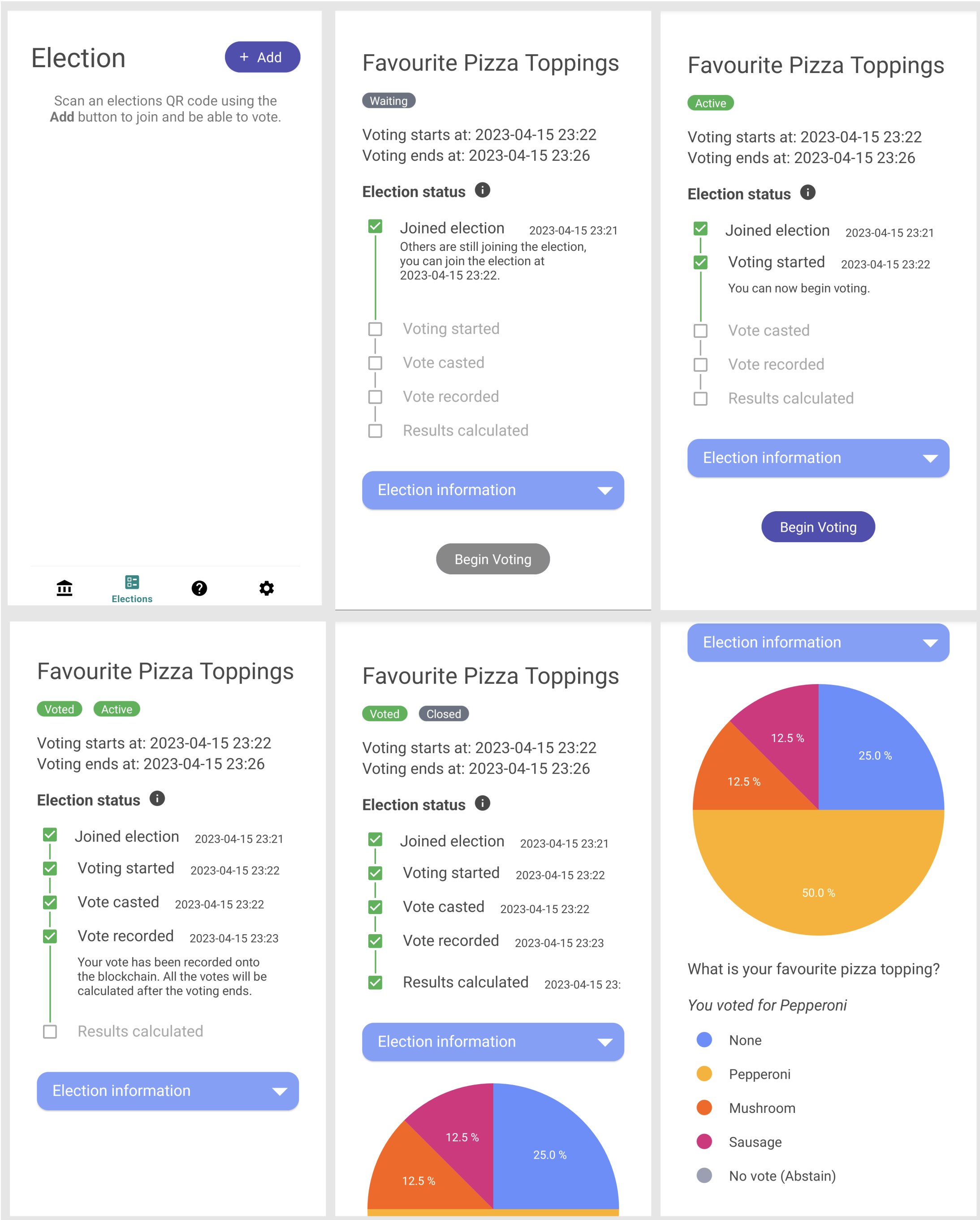

Figure J.4: Screenshots of the pages of a successful vote casting. Top row, from left to right: empty election home page, election information page before casting begins, and the election information page once casting begins. Bottom row, from left to right: election information page once vote has been recorded, election information page once results have been calculated (just the timeline) and election information page once results have been calculated (results information).

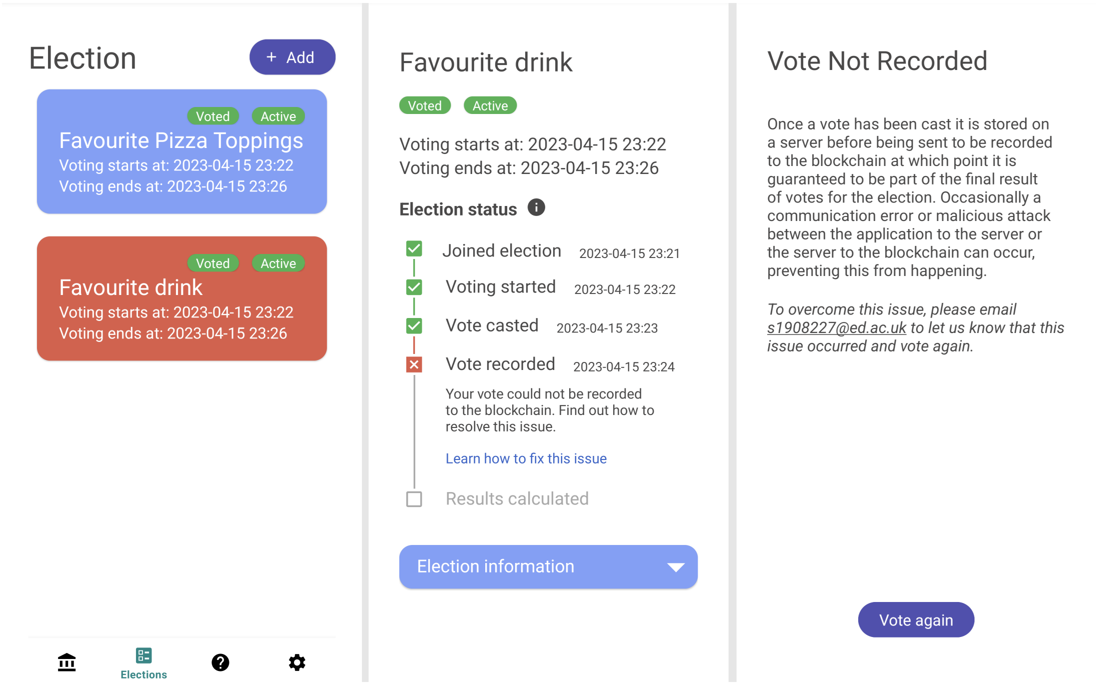

Figure J.5: Screenshots of the election pages when a vote has not been recorded. From left to right: election home page, election information page, and the vote not recorded information page that displays just before a user begins recasting their vote.

Appendix K

Evaluation script

K.0.1 Voting

Hello, my name is Sraddheya Gurung and, as part of my Honours project, I am creating a voting app and conducting usability studies which you have volunteered to be a part of. Before we begin, let's fill in the participant consent form.

Fill in the participant consent form.

Do you mind if I start recording?

Let me tell you a little bit more about my project. This app is an implementation of a voting system called E-cclesia, a voting system created by researchers at the University of Edinburgh. It is different from traditional voting because it is electronic. This means the votes can be encrypted and added to a blockchain, allowing for more security. By posting votes to the blockchain, E-cclesia is also decentralized which means that everyone who participates in an election can verify their vote and can take part in counting the votes at the end of the election while maintaining anonymity. I won't go into any more detail because it is quite complex and is not needed for this usability test.

Let me recap (from the email) how this usability test will be conducted. This usability test will be split into three sections and take place across the whole day.

1. The first section, which will start after this introduction, will be a think aloud. This means I will ask you to perform a series of tasks and I would like you to speak out your thoughts as you are performing each task. I will give an example just before we bgein. In particular I am very interested in finding out what you would have expected to happen when performing an action. At the end I will ask you to fill in a questionnaire about your opinion of the usability of the app and may ask some follow up questions.

2. After the think aloud has finished, you can go about your day as usual but will have to check the app from time to time for the status of the election. I will not be with you at this time.

3. At the end of the day, we will meet again (remind them of meeting time and online/in person). I will ask you to fill in a questionnaire about your opinion of

the usability of the app again and may ask some follow up questions.

Ask if they have any questions.

Okay, let's begin the think aloud. Remember to try to speak out all your thoughts as you are completing the tasks, whether it is something you think could be improved, something you think was good or just an observation.

Here is an example of a think aloud. Lets say the product is this window and my task is to open the window (in Figure K.1. My thoughts may be:

- I can see a big window and a small window. I like this because it gives me the option to allow different levels of ventilation.
- I open the big one first. I was not expecting to have to press a button as I turn to open the window but this does not bother me. It has opened as expected.
- I will now open the small window. This open using the same button mechanism as expected. I don't mind this however it is a bit high for me to reach so it is a bit annoying having to reach up so high to open it.

Figure K.1: Image of window to open for example in think aloud.

I may also ask you a few questions along the way. If you have any questions, please say them aloud as well, I might not answer them because it might interfere with how you would use the app, but it will still be helpful for me to know what you are confused about or want to know. Please remember I am testing the prototype, not you. Also, as this is a prototype, please note that the settings page is not functional and you would usually be logging into your account for the app first.

Scenario

- You are a student at the University of Edinburgh who wants to vote in the Computing Society's elections to make sure you get the food and drinks you like the most at the next event.
- You want to make sure you vote for your favorites correctly, but you also have a lot of deadlines coming up and so you want to complete the tasks quickly.

Ask if they have any questions.

Task

1. (Hand sheet of paper with email and QR code) You receive this email from the Computing Society about the elections they are going to host about their next event. What will you do next?
2. (Hand sheet of paper with new email and QR code) Now you have registered and are eligible to vote in the Computing Society elections and receive another email from the Computing society. Let's imagine for this scenario that your favorite pizza topping is Pepperoni. What will you do next?
3. Could you check the status of your vote?
4. (Hand sheet of paper with new email and QR code) You receive this new email from the Computing society. Let's imagine for this scenario that your favorite drink is Sprite. What will you do next?
5. Could you check the status of your vote?

Great, you have finished all the tasks. Now I will give you a few minutes to fill in this questionnaire. Please let me know when you are done.

Questionnaire done. Ask follow-up questions on why they chose some of the specific scores they chose.

For the group with no information about verification This is the end of the first section of this usability test. Now you can go about your day and check the app every so often to make sure that the vote you cast has been recorded and that the results are calculated correctly.

For the group with information about verification This is the end of the first section of this usability test. Now you can go about your day and check the app every so often to make sure that the vote you cast has been recorded and that the results are calculated correctly. It is important to verify that your vote has been recorded correctly because sometimes there may be an issue with it being sent from the server and to the blockchain. If this issue is not spotted and resolved before the final results are calculated, then your vote will not be counted.

Please email me every time you check the app throughout the day with the word "checked" so I can keep track of the times that you checked the app. The email can be in the same thread as the one in which I contacted you - so the email with the subject: x.

At the end of the day I will meet you again at (remind them of the meeting time and online/in person).

Ask if they have any questions.

End recording.

K.0.2 Verification

Do you mind if I start recording?

Now that you have been able to use the app all day, I will give you a few minutes to fill in this questionnaire. Note that this questionnaire is specifically about what you thought about the verification system, so that is the part of the application where you had to check if your vote was recorded and recast it. Please let me know when you are done.

Questionnaire done. Ask follow up questions on why they chose some of the specific scores they chose.

Great, thank you so much for taking part in this study. Please let me know your bank details so I can send you the £10 for being a participant.

End recording.

Appendix L

Evaluation emails

L.0.1 Email to join an organisation

Dear student,

For our activities this year, we will be using a new voting app called E-cclesia to let our members vote on different topics securely.

E-cclesia is the only currently available fully decentralised voting system. For a voting system to be decentralised, this means that everyone involved in the election can take part in counting the votes. This is more secure than centralised voting systems that depend on one party to count all the votes because it is secure against corruption and counting errors.

To register onto our electoral roll so that we can send you information about our upcoming elections, please join add us as an 'organisation' on the E-cclesia app via the following QR code.

From the Computing Society

L.0.2 Email to join election 1

Dear student,

As you have registered onto our electoral roll, we would like to invite you to take part in our upcoming election and vote using the E-cclesia app.

Election name: Favourite pizza topping

Voting start date: (Custom to participant)

Voting end date: (Custom to participant)

To vote in this election, please add this election to on the E-cclesia app via the following QR code. To take part in this election, **you must join the election before the voting start date**.

From the Computing Society

L.0.3 Email to join election 2

Dear student,

As you have registered onto our electoral roll, we would like to invite you to take part in our upcoming election and vote using the E-cclesia app.

Election name: Favourite drink

Voting start date:(Custom to participant)

Voting end date:(Custom to participant)

To vote in this election, please add this election to on the E-cclesia app via the following QR code. To take part in this election, **you must join the election before the voting start date.**

From the Computing Society

Appendix M

Evaluation questionnaires

M.0.1 Voting questionnaire

The layout of this form is slightly different to the real form to fit it into this page, but the content is all the same.

Please rate the following statements according to your experience.

(1 = Strongly disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly agree)

SUS questions:

1. I think that I would like to use this voting system frequently.
2. I found the voting system unnecessarily complex.
3. I thought the voting system was easy to use.
4. I think that I would need the support of a technical person to be able to use this voting system.
5. I found the various functions in this voting system were well integrated.
6. I thought there was too much inconsistency in this voting system.
7. I imagine that most people would learn to use this voting system very quickly.
8. I found the voting system very awkward to use.
9. I felt very confident using the voting system.
10. I needed to learn a lot of things before I could get along with this voting system.

Further questions

1. I am satisfied with the support information (messages, titles) when completing the tasks.
2. I trust the system that my vote will be correctly cast.
3. I trust the system that my vote will be correctly tallied.

4. I trust that the secrecy of my vote will be protected.

Do you have any other comments/suggestions to improve the system?

M.0.2 Verification questionnaire

The layout of this form is slightly different to the real form to fit it into this page, but the content is all the same.

Please rate the following statements according to your experience.

(1 = Strongly disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly agree)

SUS questions:

1. I think that I would like to use this voting system frequently.
2. I found the verification system unnecessarily complex.
3. I thought the verification system was easy to use.
4. I think that I would need the support of a technical person to be able to use this verification system.
5. I found the various functions in this verification system were well integrated.
6. I thought there was too much inconsistency in this verification system.
7. I imagine that most people would learn to use this verification system very quickly.
8. I found the verification system very awkward to use.
9. I felt very confident using the verification system.
10. I needed to learn a lot of things before I could get along with this verification system.

Further questions

1. I understood why I needed to verify that my vote had been recorded.
2. I trusted the system more because of the verification step.

Do you have any other comments/suggestions to improve the system?

Appendix N

Evaluation participants

Participant ID	Gender	Degree	Fluent English speaker	Voted in national election before	Voted in other elections before	Was given more information about verification
P1	Female	Computer Science and Maths	Y	Y	Y	N
P2	Female	Computer Science and Maths	Y	Y	N	Y
P3	Female	Computer Science	Y	Y	Y	N
P4	Male	Software Engineering	Y	Y	Y	Y
P5	Male	Computer Science	Y	Y	Y	N
P6	Male	Informatics	Y	N	N	Y
P7	Female	Politics	Y	N	N	Y
P8	Female	Social work	Y	N	Y	N
P9	Male	Computer Science	Y	Y	Y	Y
P10	Female	Biological Sciences with Management	Y	N	Y	N
P11	Male	Electronics and electrical enineering	Y	Y	Y	Y
P12	Female	Chemistry	Y	Y	Y	N

Table N.1: Information about the participants in the final evaluation. Where ”Y” corresponds to ”Yes” and ”N” corresponds to ”No”.

Appendix O

Evaluation SUS results

	SUS score	
Participant	Voting system	Verification system
P1	55	68
P2	78	75
P3	93	88
P4	88	95
P5	75	85
P6	85	88
P8	60	75
P9	88	85
P10	95	95
P11	78	68
P12	98	100
Average	83	85

Table O.1: SUS scores of each participant (out of 100), calculated using the formula in Section 2.3.3

Appendix P

Evaluation further questions results

Participant	Q1	Q2	Q3	Q4
P1	4	4	4	3
P2	4	3	3	3
P3	4	4	3	3
P4	4	4	4	4
P5	3	3	3	4
P6	3	5	5	4
P7	5	5	5	5
P8	5	5	5	5
P9	4	4	5	4
P10	4	4	4	4
p11	5	4	4	5
P12	5	5	5	5
Average	4	4	4	4

Table P.1: Results of participants answers to the extra questions on the voting systems on a scale of 1 (Strongly disagree) to 5 (Strongly agree). The question Q1-Q4 corresponds to the further questions in Appendix M.0.1

Participant	Q1	Q2
P1	2	3
P2	4	5
P3	4	4
P4	5	4
P5	4	4
P6	5	5
P7	3	5
P8	1	4
P9	4	4
P10	5	5
p11	5	3
P12	5	3
Average	4	4

Table P.2: Results of participants answers to the extra questions on the verification systems on a scale of 1 (Strongly disagree) to 5 (Strongly agree). The questions Q1 and Q2 correspond to the further questions in Appendix M.0.2.