Online Voting System

FINAL GROUP REPORT

Group 8 | Group Project | May 2019

PROJECT TEAM
Luke Aitkins | C1738772
Lara Ashford | C1718561
Bartosz Borne | C1723987
Lucas Bradford | C1600431
Lyubomir Kyorovski | C1737193
Connor Potter | C1635929
Caitlyn Powell | C1724819
Ondřej Romancov | C1731313
Alfie Rowett | C1732088
Harshit Verma | C1553228

Contents

Introduction

The project we have been faced with for the past year has been one tried and tested by organisations and countries all over the world. The idea of an online voting system makes a great deal of sense in terms of efficiency, ease of use and accessibility for the masses, however with such a large venture comes equally large, social, ethical and professional threats and issues. Over the past year we have assessed the necessary requirements for an online voting system, all the potential problems we could face in all areas of life, designed each aspect of the project and created a working prototype to meet our clients brief in an attempt to solve a globally recognised problem. As a group and as a project, we have come a long way since the beginning and the evidence is in the work we've produced, all of our designs, reports, test and iterations of the final piece have shown real development towards solving the problem of online voting and we have taken the project as far as we can, given our timeframe and the resources available to us.

The problem of voting electronically or online has been a subject of discussion since the widespread usage of computers and the internet (Hern, 2015). It has been thought of one of the most difficult upgrades to traditional paper-based systems and requires careful planning, deliberation and investment. Like most systems' transitions to an electronic or online equivalent, there are great benefits. However, it also introduces a wide range of new concerns and thus can cause more criticism than most system transitions (Schneier,2004).

When it comes to the main issues of an online voting system, the more obvious hurdles come in the form of security and privacy, and as a result we ensured that these aspects were our main focus when producing a working prototype of the system. The security of the system in question and the privacy of the users of such a system must be ensured to avoid a number of hugely impactful issues ranging from corruption and election tampering, to coercive voting and prejudice based on political beliefs, as a result the technical challenges faced made for an interesting problem to overcome.

From our initial understanding of the project and through following client meetings, group discussions and testing we have been successful at producing a working prototype that demonstrates how online voting can be used to replace the old out of date paper-based system. Our prototype isn't a final solution for a whole country but instead demonstrates how online voting can work for a small location first/county/what?

Finalised Product

Design Refinements

Throughout the design process, we created a list of design features we wanted implemented into the final version of our program; however, once the we reached the creation phase, we realised that it was both unrealistic and not possible to add certain features originally agreed upon. These features are as follows, the ability to store data as multiple copies in multiple locations, provide some back-up to coercive votes to allow voters to recast votes and finally the two-factor authentication for the administrator privileges.

In the case of the data storage, we had originally planned to store all data in multiple synchronised locations to improve the security of the system, by doing so we hoped to make the entire system more secure by having backups of all the data in locations that nobody could malicious access all of. This proved to be an issue in practice however, due to the cost of hosting databases on such a scale; because the project has no funding, we were not able to pay for the necessary services and therefore it would only be put in place if the project was to go on in the future with proper backing. We were however able to set-up a secure mix network which was part of the original design plan, meaning that data that would be travelling to the databases is being sent securely, other advantages of the mix network that were advantageous was the option for expansion in future and the ease of which it could be maintained and edited to suit our needs.

Another feature that we have made room for in future developments (if the project was continued), is the option to recast votes in the case of coercive voting. This was unable to be implemented due to not having the resources available to support such a feature, although it is possible to change a vote in the system and the databases, we have no physical presence in order to carry out the revote. If we were to fully implement a recasting votes system to combat coercive votes, we would provide physical locations where voters could go and correct their votes when they feel their original vote was unjustly coerced by a third party. Being students working on this project on a small scale, we obviously don't have the resources to offer a space to recast/correct votes, therefore although the fundamentals are there in the implementation the actual ability to recast votes is currently unavailable. Finally, the other feature we had planned to add to improve general system security was two factor authentication for the administrators; this would be in place to provide a secondary checker to all administrator actions. In the situation of running an online voting system the problem of having malicious users on the system is a real issue if they can tamper with elections or cause other damage, as a result two factor authentication for adding new administrators, starting, ending and editing elections would be a good solution to the problem. Unfortunately, due to a lack of time and a certain level of complexity to its implementation we have been unable to add two factor authentication to the final version of our system. In the future again though, we would like to see this added as an extra level of security when working with such delicate data.

As for all the functional requirements previously spoken about in the interim report, they are all present and working the final version of our system. Having prioritised certain features, such as logging in, registration and voting, it was inevitable that we would not be able to implement everything we had originally wished to. All the features mentioned above would be implemented fully if the project was to be continued, with the current version of the system being designed to support such features.

Development Methodology

At the beginning of this project we chose to use the agile software development method because the main concepts of self-organisation and cross-functional individuals seemed most appropriate the project we were embarking upon in addition to the size of our development team. We had created a Gantt chart that clearly details the timeframe of our project with all the tasks broken down and detailing how much time is allocated to them. This chart was stuck to quite well until the later stages of the project when it became delayed by a few weeks due to setbacks with resources available.

We did iterate our designs as planned, we produced several prototype iterations before arriving at our final iteration that we presented, however there is still many more iterations that could take place to make improvements and add additional features. With the agile development methodology, we had relatively fast turnarounds with each iteration of the system as hoped and quick and efficient resolution of bugs/issues.

The agile development methodology involves having a continuous interaction with the client in order to keep the project on track and make sure it is still what the client originally had in mind and if any changes are wanted to be made. The client was contacted multiple times when needed and a meeting was arranged to discuss the systems progress, unfortunately the client didn't attend this meeting. This along with some loss of communication in the group in the later stages of the project resulted in our methodology moving away from the traditional agile method and more towards a waterfall method. This was due to us having large blocks of code being tested with each new feature, rather than creating entirely new iterations and working on from there. With the last few major features being added through a waterfall method, the group was producing code, testing it and maintaining it for the last couple of weeks up until the deadline.

The previous report covered how the first phase of the project was carried out with the documentation and since then we have started work on the implementation of the system. We were concurrently working on the frontend, backend and the communications between the two. After this the work began on joining the two this worked quite well as hoped. The last phase of our project was carrying out all of our testing of the system.

System Architecture and Design

The initial plan for the code architecture was used, using Express.js (having ejs as the template engine) with the MVC design pattern and Objection.js as the object-relational mapping library, along with dependency injection. We took a code-first approach to designing the database, programmatically defining the schema and altering it when needed. The classes in the initial class diagram were not implemented exactly the way they were specified, the reason being that javascript is not a strictly-typed object-oriented language and therefore the use of classes is not always needed, however, conceptually, the diagram was kept the same.

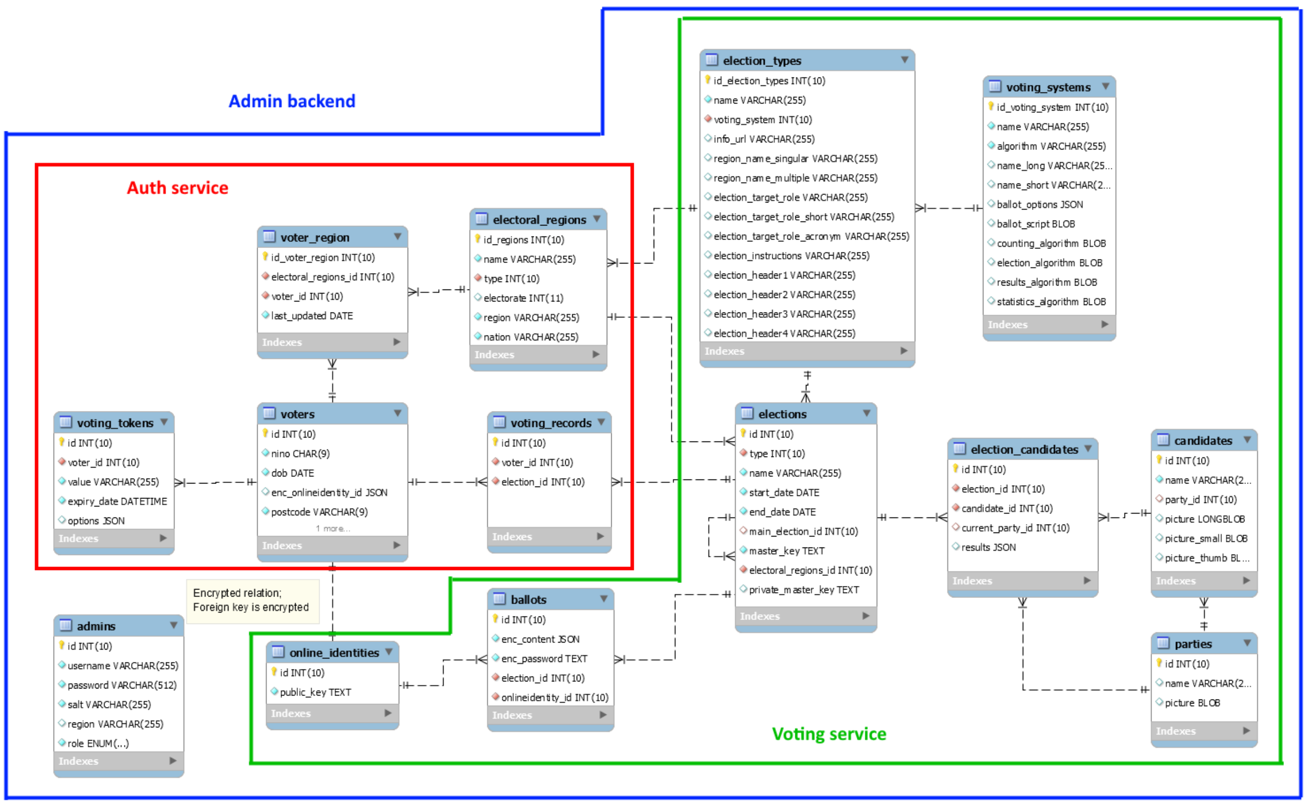
At first, we considered doing elections only by postcodes, but it turned out not to reflect the current UK voting system, where elections are done by electoral region, so the system needed to be redesigned to support electoral regions. As an addition, steps have been taken to enable the system to be extended to support different types of voting systems (like majority voting) in the future.

In our initial plan, we had only one server that handled all the functionality for voting and administration. This turned to be insufficient for voter privacy and anonymity, as they would have to send their raw credentials, such as the national insurance number, to the same place as their votes, which would undermine the procedure involving online identities that was devised previously, because there could be concerns regarding requests logging. Also, it would not be nice if anyone using the voting system had access to the admin panel from anywhere. To overcome these issues, the system was separated into three types of servers – authentication service, voting service, and admin backend. Their roles are:

- Authentication service – handles raw voter credentials, part of the authentication process
- Voting service – responsible for the voting functionality
- Admin backend – provide the functionality for managing the whole system, hidden behind a firewall and accessed only by authorised individuals

The authentication service and voting service will have their own databases, the former's containing the raw credentials for voters, and the latter's containing the data needed for the voting. The admin backend will be able to access both databases, as it needs to manage the whole system.

All these changes resulted in following the entity relationship diagram for the database:



In the current implementation, everything is stored in the same database, to ease the development process, but the different services separate the database virtually as specified above, by having them coded to access only the data they are supposed to access.

This separation of services also meant the process of voter authentication had to be redesigned, along with slight changes in the process of casting votes.

The initial system design assumed that cryptographic key pairs, like the online identities and the election master keys, will be generated on the server and sent to the client side. Unfortunately, that meant private keys were going to be sent over the network, and the new system design is avoiding that by running key generation on the client side, sending only the public keys over the network.

Core Implementation and Code

As a major part of the cryptography is happening on the front-end, it should not be taxing in terms of performance, as not all voters might have a powerful enough device. The initial plan to use RSA for the online identities needed therefore to be changed, as RSA, as all asymmetric cryptographic algorithms, is known to be slow and inefficient, and its usage for doing cryptography on the client side is becoming less popular as a result. A way to enhance asymmetric encryption performance was to switch to elliptic curve cryptography (https://www.globalsign.com/en/blog/elliptic-curve-cryptography/)so the procedures were redesigned to use it. Because RSA was needed mostly for signing, it was suitable to swap it with the elliptic curve digital signature algorithm (ECDSA). Also, because the design required performing Diffie-Hellman key exchanges, the protocol used to perform them was selected to be Elliptic-curve Diffie–Hellman (ECDH). The only instance where the use of RSA remains is for the election master keys, so they can be used in the process of encrypting/decrypting ballots.

For storing of passwords, in the case of admins, and obtaining AES keys with sufficient length from shorter strings, in the case of the personal access token (PAT), password-based key derivation Function 2 (PBKDF2) was used, aimed to minimise the risk of someone brute-forcing an admin's password, or a lost flash drive containing an online identity respectively.

For doing cryptography on the front-end, the Web Crypto API (https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API) is used, because it is available in all browsers, and for the backend, the built-in node.js crypto library (**https://nodejs.org/docs/**latest-**v8.x/api/crypto.**html). There are compatibility issues between the two, as they are not implemented in the same way, but wrapping functions, adapting one's implementation to the other's, were devised to address them.

As the result of switching to elliptic-curve cryptography, and the separation into services, the following base algorithm descriptions were produced:

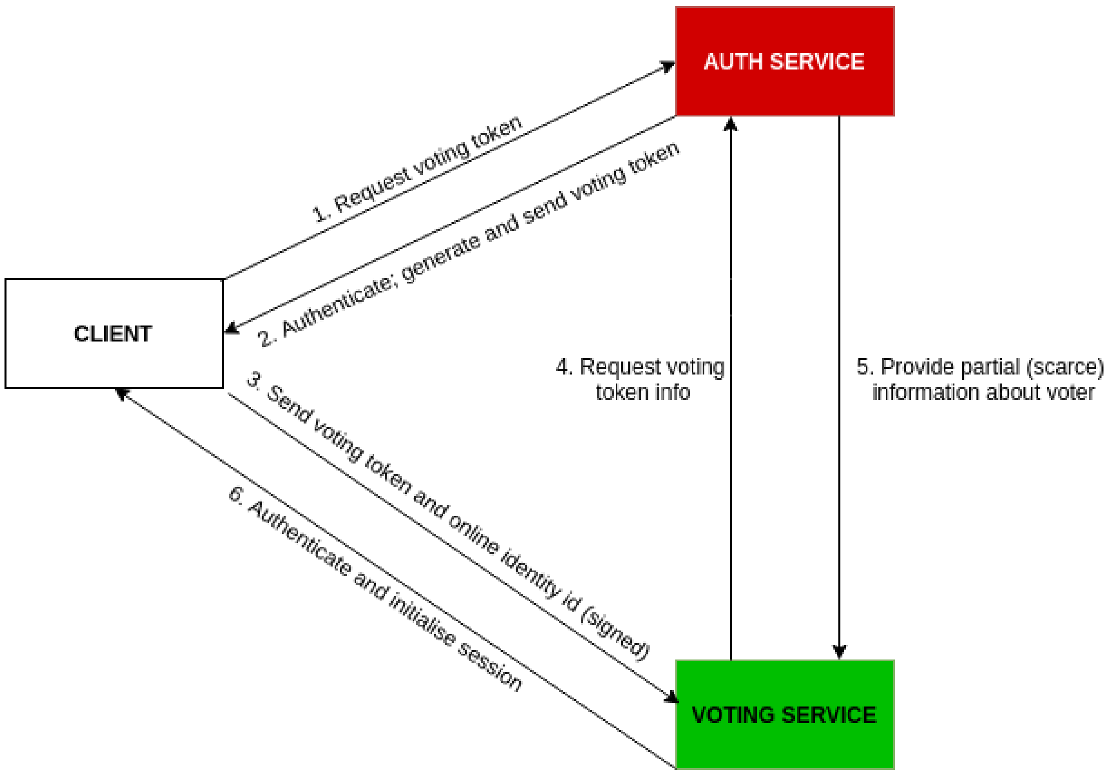
Setting up elections:

1. Admin populates elections details
2. A 4096-bit RSA Keypair, providing OAEP padding with SHA-512 verification signature, is generated on the admin's device
3. The public key from the pair, gets sent along with the election details, and stored as the elections master key
4. The private key from the pair is stored to the admin's device (later put on a flash drive and in the secure briefcase as outlined in the original project plan)

Registering voters:

1. Voter register their details beforehand, which get sent to the auth service to be stored
2. *(Voter goes to the local registrar to receive the letter containing the PAT, and the flash drive containing their online identity file, providing proof of their identity to the registrar)*
3. Admin confirms the data in-person and if valid, selects to confirm the voter registration
4. An ECDSA keypair over the prime-256 curve with SHA-256 as the verification signature is generated on the admin's device
5. The public key from the pair is sent to the admin backend, instructing it to create a new online identity entry and retrieve its id in the database
6. A PAT is generated, using a cryptographically strong random function, on the admin's device
7. An AES-256 key is derived from the PAT using PBKDF2 on the admin's device
8. The private key from the pair and the database id of the online identity entry are encrypted with AES-256(GCM block cipher) on the admin's device
9. The encrypted online identity database id is sent to the admin backend, instructing it to store in this voter's entry
10. The encrypted private key and encrypted online identity database id are stored in an ".oid" file that is written to the flash drive
11. The PAT letter is generated and printed, and handed to the voter

Voters authentication procedure:


AUTH SERVICE
CLIENT
VOTING SERVICE
1. Request voting token
2. Authenticate; generate and send voting token
3. Send voting token and online identity id (signed)
4. Request voting token info
5. Provide partial (scarce) information about voter
6. Authenticate and initialise session

1. Voter types in their National insurance number and PAT, and chose their online identity from the file picker and click login
2. The encrypted private key and encrypted online identity database id are loaded from the online identity file on the voter's device
3. The AES-256 keys needed for decrypting both are generated using PBKDF2 from the PAT on the voter's device
4. Decrypt both using AES-256(GCM) block cipher on the voter's device and check if decryption is successful, otherwise the voter has provided wrong PAT, if successful, store the encrypted private key to browser's localStorage, to be used later during voting
5. Produce an ECDH keypair over the prime-256 curve
6. Produce a SHA-384 hash of the encrypted online identity database id and send it along with the National insurance number and public ECDH key to the auth service, requesting a voting token
7. Voting system retrieves data about the voter from the database and computes a SHA-384 hash of the voter entry's enc_onlineidentity_id, and if the hash sent by the voter matches it, generate an ECDH keypair, compute the shared secret from the private key and the voter's ECDH public key and issue a voting token(a uuidv4 string) to the voter, storing as details the computed shared secret, expiry date (40 minutes after issuing), the electoral regions the voter can vote in (obtained by the voter's postcode through from an api) and computing voter's age group from their date of birth. Only one voting token may be active per user. The auth service sends back the voting token id and the ECDH public key that it generated.
8. Voter receives voting token, computes the shared secret from the received ECDH public key and the voting (on the voter's device)
9. Voter's online identity database id is encrypted with AES using the shared secret, a signature is produced from it with ECDSA using voter's online identity private key

    (On the voter’s device) and the two sent to the voting service alongside the voting token id

10. Voting system retrieves voting token details from auth service, from those details, it gets the shared secret and decrypts the received encrypted online identity database id with AES. Then it retrieves the public key for that online identity database id and using ECDSA with the public key on online identity database id, verifies that the signature send by the voter is valid. If it is, the voting service creates a session for the voter, the online identity database id and other voting token details are stored as session variables

- NOTE: The communication between the voting service and the auth service is secured with ECDSA signatures and ECDH keys, so that no spoof requests are sent to the auth service regarding token info retrieval;

The logic behind counting votes, voting and dealing with coerced voting remains the same as in the original specification, only the algorithms being replaced by elliptic curve ones.

This separation of services has unfortunately placed the voter (client on the diagram above) in the position of a man in the middle. Which could mean that another rogue voter could be spoofing other’s data. Ideally, we want to run extensive penetration testing and security analysis to develop the authentication protocol further. As a whole, we would want to create a secure authentication procedure that can be used even over an insecure network.

Architecture for the MIX networks has remained largely unchanged from the specification in the interim report. The only notable difference is that the requests are now randomly sent through multiple nodes, meaning that a man in the middle anywhere in the network would never be able to tell which request came from which client nor the server.

Team work

Initially when the group first met up, we began organising effective means of communication. The consensus among the group was that Facebook Messenger was the preferred medium and thus this was the medium the team all used. Using just one platform to communicate general queries and organise meetings meant every team member was on the same page and aware of everything that was happening at any given time. The next step was to have a means of file sharing. For this, the team used two different platforms; Microsoft OneDrive for code, and GitLab for code. OneDrive was used as it allowed us all to view, upload, and edit documents to ensure they were at an excellent standard before submitting. It was decided that we would use another platform, GitLab, for any code written. GitLab is well known to be an excellent means of sharing code, allowing users to upload and edit code in full view of all the other users who also have access to it. This means that all users can see when any change has been made, allowing everyone in the team to view these changes and stay up to date with the most current build of the system.

Now we have a means of communication and a space to upload and share our work, the next step was to have a means of dividing up tasks. For this, Trello was the software of choice. Trello is a task management application that provides users with a visual overview of what is being worked on and by who. Trello provided features such that if someone in the team had nothing to do, we would be able to see this and assign them a task, or if a team member was waiting for some task to be completed, they could check the teams Trello to see how it is progressing. It also prevented multiple people doing the same task, for everyone would check the Trello before starting to ensure it is not already in progress. The use of this application proved invaluable and resulted in tasks being completed much more fluidly than what would be if we did not have the aid of Trello for organisation.

On top of using the appropriate software to keep the team organised, then came decisions on how the group itself will function. One problem we faced was deciding on how we would organise such a large group. With ten of us, it could have been problematic ensuring every member had a purpose. To overcome this, we made the decision to split the team up into three sub-teams, two of which had four members and the third had two. Each with a specific purpose and a clear goal to work towards: One sub-team's purpose was to work on the back-end of the system, another was to work on the front-end, and the third team, the smallest of the three, was dedicated to managing the mix network. Each one of these teams had their own designated 'manager', a team leader which would dictate the vision of the project. All three teams met up for an hour meeting every Thursday to discuss progress made that week, the current status of the system, and what the goals are for the coming week. These meetings proved unquestionably valuable as many of the important decisions made for the project were during this one hour period. For example, initially only one group member was working on the MIX network and there was an additional team member working on front-end and documentation. However, three weeks into the project it was decided that more help would be needed on the MIX network for it to be completed within the desired time frame. Decisions such as this one displays the flexibility we were able to implement which was an important factor that lead to our success.

Throughout this group project there were some minor disagreements among some team members. These were due to some miscommunication, where one team member may have misinterpreted what was expected of them resulting in progress being temporarily delayed. This dispute, much like any other that was encountered, was swiftly remedied by discussing and clarifying the issues raised in the next weekly group meeting.

Testing

In order to fully analyse and test the functionality of our online voting system, we carried out our test plan that we developed prior to developing the solution, on top of this we also carried out a user survey and asked individuals outside of our development group to use the system and mark the system from a user perspective.

Regarding the scope of the testing, there are several boundaries we have which mainly come from the scale of the project, short timescale it needs to be done in and resources available to us. The system we have produced is a When producing a working and testable prototype voting system, tests that require larger scale environments are not possible without full implementation into either the real world or a much larger testing pool. The most obvious issue we face here is the testing of system stability, as without releasing the system to the entire population, it is difficult to fully test the robustness of the system and how it can handle large scale network traffic of an entire population of users. Another function we are unable to test is the ability to stop coercive voting, as although we have a system in place where voters can change their vote, the need for a separate database which stores the traditional votes from toll booths as well as the changed coercive votes is too much work and time for us to implement in a prototype system. Therefore, until full application of our system is in place we are unable to test these functions.

We tested individual functions of the system by carrying out black box tests in which we recorded the expected result, the actual result generated and any other noteworthy information we need to refer to and look into at a later date. During this process it helped us to locate errors in the system and eliminate any broken features that we can fix. The testing took place once each function was complete (for example once voting was available, voting was tested), the testing was carried out by both our development team and an outside user. This allowed us to make sure there that usability without any prior knowledge is possible. Below is the complete test tables from the developer perspective which tested the functionality of all aspects of the system required from our original functional requirements.

Test Case Id: 1	**Test Purpose: To test that registrars can register voters**			
Environment: Running in Google Chrome on Windows 10				
Preconditions: The voters are not currently registered.				
Test Case Steps:				
Step No	Procedure	Expected result	Actual Result	Pass/Fail
1	Registrar selects “Register” button	Interface shows registration screen with input boxes	Registration screen shows.	Pass
2	Registrar selects “Date of Birth” box and types “01/01/1990”	"Date of Birth “ box displays registrar’s input	"Date of Birth “ box displays registrar’s input.	Pass
3	Registrar selects “Postcode” box and types “CF24 3AA”	“Postcode” box displays registrars’ input	Postcode is displayed in input box	Pass
4	Registrar selects “National Insurance Number” box and types “12345”	Dialogue box pops up informing registrar that the NI needs to be in the format “AA 00 00 00 A”	Dialogue box pops up if incorrect, or works otherwise	Pass
5	Registrar selects “National Insurance Number” box and types “AA 11 11 11 A”	“National Insurance Number” box displays registrar’s input	Input displayed in field correctly	Pass
6	Registrar clicks “Register” button	Interface shows registration confirmed screen	Registration is successful	Pass
Comments: Registrar enters voter’s information, not their own				

Author: Caitlyn Powell	**Checker: Alfred Rowett**

Test Case Id: 2	**Test Purpose: To test the Login requirement**			
Environment: Run in Google Chrome on Windows 10				
Preconditions: Users know their log in details				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/Fail
1.	Voter selects the “Log in” button	Voter is redirected from the home page to the Log in Page	Login screen displayed	Pass
2.	Voter enters “HJ 68 30 23 K” into the National Insurance dialogue box	The text entered appears in the dialogue box	User can enter national insurance number	Pass
3.	Voter enters “testpassword” into the password dialogue box	The text appears in the password box in the form of “*”	Password (PAT) entered	Pass
4.	Voter selects the “Login” button displayed at the bottom of the screen	Voter is redirected to the current election page	Login works correctly	Pass
Comments: N/A				
Author: Lara Ashford			**Checker: Alfred Rowett**	

Test Case Id: 3	**Test Purpose: To test the Voting, pre-defined voting selection and disallowing of duplicate votes requirement**			
Environment: Run in Google Chrome on Windows 10				
Preconditions: User is Logged in and is on the elections page				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/ Fail
1.	Voter selects the “Vote” button	Interface displays drop down list	Drop down list appears	Pass

2.	Voter selects “2018 General Election” from drop down list	The dialogue box displays the chosen election “2018 General Election”	Chosen election is displayed	Pass
3.	Voter selects the “Submit button”	Interface redirects to page showing the pre-set candidate details for that election	Voter is redirected to the correct page	Pass
4.	Voter selects the “Vote” button next to “Jeremy Corbyn”	The vote button for “Jeremy Corbyn“ turns Green	Voter can select candidate	Pass
5.	Voter selects the “Vote” button next to “Boris Johnson”	The Vote button for “Jeremy Corbyn” reverts back to grey, the vote button for “Boris Johnson” turns green	Voter can change their vote	Pass
6.	Voter selects the “Submit” button	Interface redirects to page showing the voter’s selection of “Boris Johnson” and asks the voter to confirm their choice	Voters selection is submitted	Pass
7.	Voter selects the “Cancel” button	Interface redirects back to the page showing the list of candidates for the “2018 General Election” with all the vote buttons grey.	Voter cancels their choice	Pass
8.	Voter selects the “Vote” button next to “David Cameron”	The vote button for “David Cameron” turns green	Voter can change choose another candidate	Pass
9.	Voter selects the “Submit” button	Interface redirects to page showing the voter’s selection of “David Cameron” and	Voter is redirected to another page with their selection	Pass

		asks the voter to confirm their choice		
10.	Voter selects the “Confirm” button	Interface redirects to page confirming voter has voted for their chosen candidate “David Cameron”	Voter can confirm their choice	Pass
11.	Voter selects the “Elections” button followed by the “Vote” button on the elections page	Interface displays drop down list	Interface displays drop down list	Pass
12.	Voter selects the drop-down list and views the elections displayed	The drop-down list does not display “2018 General Election”	No General Election displayed	Pass
13.	Voter logs out and logs back into the system with the same credentials and carry out step 11 and 12 again.	After logging out and back in the Interface displays drop down list on elections voting page. The drop-down list does not display “2018 General Election”	Still no General Election displayed on the drop-down list	Pass
Comments: N/A				
Related Tests:				
Author: Caitlyn Powell & Lara Ashford			**Checker:**	

Test Case Id: 4	**Test Purpose: To test the Admin Login requirement**			
Environment: Run in Google Chrome on Windows 10				
Preconditions: Admin knows their log in details				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/Fail
1.	Admin selects the “Admin Log in” button	Admin is redirected from the home page to the admin log in page	Login screen displayed	Pass

2.	Admin enters “K123” into the username dialogue box	The text entered appears in the dialogue box	Dialogue box holds text	Pass
3.	Admin enters “adminpassword” into the password dialogue box	The text appears in the password box in the form of “*”	Admin password is entered and asterisked	Pass
4.	Admin selects the “Login” button displayed at the bottom of the screen	Admin is redirected to the administrator control panel page	Admin is redirected to admin page	Pass
Comments: N/A				
Related Tests:				
Author: Lara Ashford			**Checker: Alfred Rowett**	

Test Case Id: 5		**Test Purpose: To test the Statistical summaries requirement**		
Environment: Run in Google Chrome on Windows 10				
Preconditions: The “2018 General Election” has finished and votes have been cast and the Admin is Logged in				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/Fail
1.	Admin selects the “Login” button displayed at the bottom of the screen	Admin is redirected to the administrator control panel page	Redirected to login page	Pass
2.	Admin selects the “View Results” button	Admin is redirected from the home page to the election result page	Admin is redirected to results page	Pass
3.	Admin selects the “2018 General Election” button	Admin is redirected to a page showing the results of the election, a list is displayed showing all the candidates IDs, Name, Image, Info, End date Vote Count.	Admin can select the election they wish to view results for	Pass

4.	Admin selects “View results by age demographic”	Admin is redirected to a page showing a list of all the candidates. There is a percentage of how many people voted within each age bracket displayed next to each candidate.	Admin redirected to age demographics page shown by candidate	Pass
Comments: N/A				
Related Tests:				
Author: Lara Ashford			**Checker: Alfred Rowett**	

Test Case Id: 6	**Test Purpose: To test the administrator privileges requirement**			
Environment: Run in Google Chrome on Windows 10				
Preconditions: Administrator is already registered				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/ Fail
1.	Admin 1 selects “Login” button on interface’s navigation bar	Interface shows login screen	Login screen is displayed	Pass
2.	Admin 1 selects “Username” box and types “K123”	“Username” box displays admin’s input	Username is taken as input	Pass
3.	Admin 1 selects “Password” box and types “adminpassword”	“Password” box displays asterisks from admin’s input	Password is taken as input	Pass
4.	Admin selects “New Election” button	Interface shows election detail interface with drop down boxes	Election detail is displayed	Pass
5.	Admin selects “Wales” from “Location” drop down list	Dialogue box pops up informing admin that this location is not suitable for the election type chosen	Dialogue displays the location if its suitable	Pass

6.	Admin selects “United Kingdom” from “Location drop down list	“Location” box displays admin’s selection	Location displays admins selection	Pass
7.	Admin selects “3” from “No. of Candidates” drop down list	“No. of Candidates” box displays admin’s selection	No of candidates displayed from admins selection	Pass
8.	Admin selects “01/01/2020” from “Date” pop up box	“Date” box displays admin’s selection	Date displays admin selection	Pass
9.	Admin selects “Next” button	Interface shows screen with candidate placeholders	Candidate placeholder shown	Pass
10.	Admin selects “Details” button next to Candidate 1	Interface shows screen with input boxes for candidate details	Input screen appears	Pass
11.	Admin selects “Candidate’s Name” box and types “123Bob”	Dialogue box pops up informing admin that candidate name must be made up of letters	Pop-up appears	Pass
12.	Admin selects “Candidate’s Name” box and types “Jeremy Corbyn”	“Candidates Name” box displays admin’s input	Box displays the admins input for name	Pass
13.	Admin selects “Party” drop down list and selects “Labour”	“Party” box displays admin’s selection	Admins candidate party selection is displayed.	Pass
14.	Admin selects “Information” box and types “CF24 3AA”	Drop down list suggests house and street names for admin’s inputted postcode	Houses are suggested to user	Pass
15.	Admin selects “5 The Parade” from drop down list	“Information” box displays admin’s selection	Box displays correct information	Pass
16.	Admin selects “Image” button and uploads image file	Image box displays name of file chosen	Box displays file name	Pass
17.	Admin selects “Submit” button	Interface shows screen with candidate placeholders	Placeholder screen is displayed	Pass
18.	Admin selects “Details” button next to Candidate 2	Interface shows screen with input boxes for candidate details	Input screen for candidate 2 appears (repeat steps from previous candidate entry)	Pass

19.	Admin selects “Next” button	Dialogue box pops up informing admin that not all candidates have been assigned	Pop-up appears correctly	Pass
20.	Admin selects “Remove” button next to Candidate 3	Interface shows screen asking to confirm removal of candidate	Screen asks for removal confirmation	Pass
21.	Admin selects “Yes”	Interface shows screen with candidate placeholders	Admin is redirected back to candidate page	Pass
22.	Admin selects “Next” button	Interface shows screen showing all election and candidate details asking admin to confirm election set up	Screen asks for admin confirmation of selection details	Pass
23.	Admin selects “Confirm” button	Interface shows list of elections screen with options for the set-up election	Election set-up screen is shown	Pass
24.	Admin selects “Edit” button next to “01/01/2020 General Election”	Interface shows election detail interface with drop down boxes showing previously selected information	Edit screen for general election on drop-down menu shown	Pass
25.	Admin selects “05/05/2020” from “Date” pop up box	“Date” box displays admin’s selection	Date box displays admins choice	Pass
26.	Admin selects “Local Election” from “Type of Election” drop down list	“Type of Election” box displays admin’s selection	Election type dialogue box displays admins choice	Pass
27.	Admin selects “Next” button	Dialogue box pops up informing admin that type of election and location conflict	Pop-up displays correctly	Pass
28.	Admin selects “General Election” from “Type of Election” drop down list	“Type of Election” box displays admin’s selection	Type of election displays admins choice	Pass
29.	Admin selects “Next” button	Interface shows screen with candidate placeholders	Candidate placeholders displayed	Pass

30.	Admin selects “Edit” button next to Jeremy Corbyn	Interface shows screen with input boxes for candidate details showing previous input	Previous inputs displayed for candidate details	Pass
31.	Admin selects “Next” button	Interface shows screen showing all election and candidate details asking admin to confirm election amendments	Screen appears asking for admin confirmation of election amendments.	Pass
32.	Admin selects “Confirm” button	Interface shows list of elections screen with options for the set up election, election has been added to database with all selected data correctly	Elections is added to database with all correct data	Pass
33.	Database is checked to see changes	Election has been added to database with correct information	Election appears in the database	Pass
34.	Admin selects “Delete” button next to “05/05/2020 General Election”	Interface shows screen asking admin to confirm deletion of election	Interface asks for admin confirmation of deletion	Pass
35.	Admin selects “Confirm” button	Interface shows list of elections screen with no elections	Screen confirms deletion and shows no elections remaining	Pass
36.	Database is checked to see changes	Election has been removed from database	Database no longer contains the elections deleted	Pass

Comments: N/A	
Related Tests:	
Author: Caitlyn Powell	**Checker: Alfred Rowett**

Test Case Id: 7	**Test Purpose: To test the data result/storage requirement**

Environment: Run in Google Chrome on Windows 10				
Preconditions:				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/Fail
1	Voter attempts to access voting page after election is finished	The page will not display, and error message is given	Page doesn't display	Pass
2	Spoof AJAX call is sent to the verification server requesting a signature	Signature will not be sent and an error response will be returned	No signature sent	Pass
3	Spoof AJAX call is sent to server attempting to vote without a signature	Vote will not be added to database and error response will be given	Vote isn't counted	Pass
Comments: N/A				
Author: Luke Aitkins			**Checker: Alfred Rowett**	

Test Case Id: 8	**Test Purpose: To test the accessibility requirement**			
Environment: Run in Google Chrome on Windows 10				
Preconditions: A voter has the log in page open on a computer, laptop and smartphone				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/Fail
1	Voter Enters "HJ 68 30 23 K" into the National Insurance dialogue box on the computer, laptop and smartphone	The text entered appears in the dialogue box on the computer, laptop and smartphone	National insurance number is held in dialogue box	Pass
2	Voter Enters PAT into the PAT dialogue box on the computer, laptop and smartphone	The text appears in the password box in the form of "*" on the computer, laptop and smartphone	PAT is displayed in dialogue box	Pass
3	Voter selects the "Login" button displayed at the bottom of the screen on the computer, laptop and smartphone	Voter is redirected to the current election page on the computer, laptop and smartphone	Voter is redirected to voting side of program	Pass

Comments: N/A				
Related Tests:				
Author: Lara Ashford			**Checker: Alfred Rowett**	

Test Case Id: 9	**Test Purpose: To test the user privacy requirement**			
Environment: Run in Google Chrome on Windows 10				
Preconditions:				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/Fail
1	Add new user to the system and look at the packets sent to the MIX layer	No user information is identifiable or in plaintext (body of request is encrypted)	All information is encrypted	Pass
2	Add new user to the system and look at the packets sent to the Verification Server layer	No user information is identifiable or in plaintext (body of request is encrypted)	All information is encrypted	Pass
3	Add new user to the system and look at the packets sent to the Main Server layer	No user information is identifiable or in plaintext (body of request is encrypted)	All information is encrypted	Pass
4	Send vote and check what changes have been made to all database tables	Users votes are not linked to their personal data	No user linked to their votes	Pass
Comments: N/A				
Related Tests:				
Author: Luke Aitkins			**Checker: Alfred Rowett**	

Test Case Id: 11	**Test Purpose: To test the performance requirement**			
Environment: Run in Google Chrome on Windows 10				
Preconditions: Test Case 1-4 are performed				
Test Case Steps:				
Step No	Procedure	Expected Result	Actual Result	Pass/Fail
1	Whilst carrying out Test Case ID 1-4 each action on the webpage is timed from the point a button is clicked to the time it takes for the corresponding action to occur	The system does not lag during any system process and the system is responsive to all user inputs. The system should not take longer than 3 seconds to complete any action.	All processes run in an acceptable time.	Pass
Comments: N/A				
Related Tests: Test Case 1-4				
Author: Lara Ashford		**Checker: Alfred Rowett**		

Upon completion of testing we can confirm that we have met all the targets/requirements we set out to achieve during the process of this project. With all tests being passed in a very satisfactory way, we are happy with the work we've done and the functionality of the project. In a couple of cases, the tests did initially show areas in which we needed to make improvements; one area in particular was the absence of an election results statistics page which would be used to display results by demographic such as age group. Looking at this test result early on showed us that although we had the results, the demographic charts weren't being displayed and required work before final implementation and retesting. Overall, the use of developer side testing was very useful and allowed us to tick off requirements we wanted to meet during the project, with only features that were deliberately missed not being tested in the final version.

The second phase of testing was to provide several outside individuals access to our system in order to test the functionality and usability from a user perspective. Where in the real world we would require the online voting system to be used by millions of individuals with varying levels of technical understanding, we hoped that by testing our near final iterations of the system on outside people we could gain a better understanding of where the system needed improvement from a user point of view. Luckily, we had considered the importance of the usability of such a system in the earlier stages of the project, and as a result strong wireframe models and human computer interaction understanding ensured that our system was largely well received by those who tested the system. With all members of our test group saying that the systems navigation and user feedback was very satisfactory, and the general aesthetic appeal of the platform was a strong aspect. As you can see from the survey forms below, our test users were more than happy with all areas of the online voting system, with only one individual expressing some concern with the functionality of the platform (by rating it only just above average on the scale). Going forward into the future with the project however, we would make sure to test our platform on users with specialist conditions (such as varying levels of visual impairment) to check the accessibility of the platform for everyone who might use it. Unfortunately, we couldn't test our system in such a way at this time but would hope to do so in the future to improve overall usability.

Group 8 – Online Voting System Feedback form:	
Criterion:	**Rating from 1-10 (1 being 'Definitely not' to 10 being 'Definitely yes')**
Is the language used clear and understandable?	*10*
Is the system easy to access?	*10*
Is the user interface easy to understand?	8
Strong use of layout and easy to interpret?	*8*
Is the system easy to navigate through all screens?	*10*
Is it always clear at which point the user is at?	*8*
Are icons being used helpful when navigating the system?	10
Does the programs visual appearance appeal to you?	*10, especially the Xes*
Does the system offer all the services you wish to see from an online voting system?	10
Any further comments?	The visual design is very good. It is easy to use and well laid out.

These are two of the user feedback survey results that we received, showing high scores across the board for all areas of usability and user functionality.

Group 8 – Online Voting System Feedback form:	
Criterion:	**Rating from 1-10 (1 being 'Definitely not' to 10 being 'Definitely yes')**
Is the language used clear and understandable?	*9*
Is the system easy to access?	*8*
Is the user interface easy to understand?	*10*
Strong use of layout and easy to interpret?	*8*
Is the system easy to navigate through all screens?	*8*
Is it always clear at which point the user is at?	*9*
Are icons being used helpful when navigating the system?	*9*
Does the programs visual appearance appeal to you?	*8*
Does the system offer all the services you wish to see from an online voting system?	*6*

Any further comments?	Insert comments here….

Legal, Social, Ethical and Professional Issues

In regard to the legal, professional, ethical and social issues we face when producing such an impactful system, there were a number of problems we had taken into account whilst designing the program at the start of the project. Some of the issues we believed we would face included the problem of third-party advertising on an online platform such as our own, coercive voting, data protection and the risk of malicious attack at any point of our system. Having now created our final prototype there are a number of these issues we have had to ignore for the time being, simply due to their complexity and the lack of resources required in order to resolve them. For example, in our current version of the system, it isn't possible to recast votes in the case of coercive votes (a very possible social threat to many users), we haven't been able to deal with this problem however due to a lack of time and team size. In another case, we aren't able to test and develop protection for our database storage, with a plan to create a synchronised database system spread over multiple databases; this is due to a complete lack of funding given the size of the project at this stage and we can't afford to host databases for the purpose of our testing; however in the case of gaining funding in the future this would definitely be implemented to improve security. All the legal, professional, ethical and social issues that were mentioned in the interim report are still relevant at the end of the project and as a result have affected our design methods and features. With key areas in security and website neutrality being of the highest importance, with the system being an online voting system we must provide a secure, neutral, inclusive and accessible platform and as a result this is how we've both designed and created our final working iteration of the online voting system.

Evaluation

The solution meets the basic functional requirements expected by the client. Voters and admins can log in on – users are authorised and so no unregistered users may access the system, protecting from fraudulent use. Voters can access the system from any network-enabled device and participate in voting for their region. Admins have specific privileges (for their regions) such as adding, editing and removing elections, and admins can further add additional administrators. The results of the elections can be viewed, and some statistical summaries are available. Some of these requirements however are not fully implemented – adding new administrators does not use the two factor authentication we first proposed using. The statistics available are also rather limited.

Regarding the non-functional requirements, our solution is not fully complete; while the system is intuitive, has consistent system aesthetics in line with official UK government websites, is accessible from any internet enabled device, prevents voters casting duplicate votes, and stores votes securely and privately, it does not have any preventative measures against coercive voting (although work has been done towards this end) and data is not stored as multiple copies to ensure the system is robust. There is also an issue regarding performance – we originally proposed that no task will take any longer than 3 seconds to load, however some tasks such as viewing election results as an admin do not meet this criterion.

Overall the solution meets the vast majority of the client's requirements, functional and non-functional, with criteria such as coercive voting being close to being implemented, if we had more time. It fulfils all the functional requirements given to us, and the key points originally given to us in the project brief – to support secure voting over the internet from any network-enabled device; for it to be usable in local and national and elections; for the vote information to be stored in a secure database; and for authorised election officials to be able to extract final voter tallies as well as statistical summaries.

Future Development

Due to time and resources limitations, our team had to prioritise work in order to deliver a functional prototype in time. In the case of further development, our team has outlined the additional features that would have been added to the system, improvements to the currently realised solution and also revisions on the implementation process itself in the case of eventual system development in a more realistic scenario with a larger budget as well as without other structural and time limitations.

The current prototype implementation is only supporting the model of general elections but in the final state as a nationwide system posing as an optional way to vote alongside of traditional voting, the system is to support other type of elections. In the case of statistics, the current system is only able to provide general statistics about the age and registered address of the users across the nation but is not able to generate more sophisticated demographic models based on for example gender, ethnicity or social status or even local statistics in different regions. These additional statistics would be added in the future.

In terms of security, our team did not implement all of the originally planned features and would therefore plan to include them in the future. One of the most critical security threats is the absence of coercive voting prevention feature. This feature would allow the users to recast their vote in the case of them being coerced by a third party. The prototype is also missing additional security features such as two factor authentication of the users which would further enhance the security of the system.

Our team has been developing the system with these features in mind and the potential future addition of these features should require minimal changes to the software and maintenance of the code, but was still not capable of implementing these fully, mainly from financial limitations of the project. Further data validation in the database and other backend services are also amongst the future to-dos as they would also enhance the reliability and security of the system as the prototype only has frontend data validation.

It would also be wise to include more accessibility features such as high contrast mode and making sure that voter interface performs well with screen readers for users that are visually impaired. This would be quite straight forward to ensure given more development time.

The implementation of the backend systems including the user and ballot database and mixed network proxy servers was intensely influenced by the significantly low amount of financial resources. As of that the prototype is hosted on the university servers with obvious complications such as limited availability due to the university's VPN access requirements or limited bandwidth. This could be solved in the future by hosting the system on governmental servers or at partner firms with scalable and available servers which would additionally solve the current problem of the solution’s slow performance at times.

Due to the financial restrictions of the project, the prototype implements a single MySQL database. In the case of further development, we would focus more on reliability and availability of our database – data needs to be stored in multiple places for robustness of the system. The originally proposed MySQL cluster requires a relatively cheap license for a governmental institution and would be more suitable for the final version of the system.
Additionally, in order to fully execute the project in the business case we would need to support the huge numbers of people using the system to vote nationally; potentially hundreds of thousands of people on the system at once at peak voting times. While a third-party cloud service solution could solve this in future, the nature of the cloud means that we would not fully be in control of the security of our project, and national voter data would be passed through these third parties.

Conclusion

To summarise, we have made significant progress towards a complete solution. The fundamental components of the project are operational and most of the requirements for the project have been met. Although our plan, design and methodology has changed during the timeframe of this project, we have worked in a very flexible manner that has allowed us to work efficiently and fluidly as a team; breaking up into smaller teams or even pairs to work on individual aspects to ensure we meet all our milestones throughout production. As a team we were able to look to each other for help in both the practical production side of the project as well as the assessment, testing and report side with each individual showing off their strengths at each stage. With more time and financial support and better resources available to us, it would be feasible for us to meet all the requirements and hopes for the project as set out by our client during the initial brief and design talks; and not only this, but the system could be improved to be even more secure and robust, capable of meeting the demands of a large number of users in the real world, whilst maintaining a high level of usability.