Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

At the Heart of South Leinster

Online Voting System

Design Document

Name: Mark Kelly

Student Number: C00198041

Course: Bachelor of Science (Honours) Software Development

Supervisor: Dr. Lei Shi

Date: 18th April 2018

Table of Contents

1. Introduction

This document outlines the architecture required to deliver the requirements documented in the functional specifications. Included in this document is, the database design, use case diagram, brief and detailed use cases, system sequence diagrams and a functional overview model. The purpose of this document is to familiarise the reader with the structure and components of the web application.

2. System Architecture

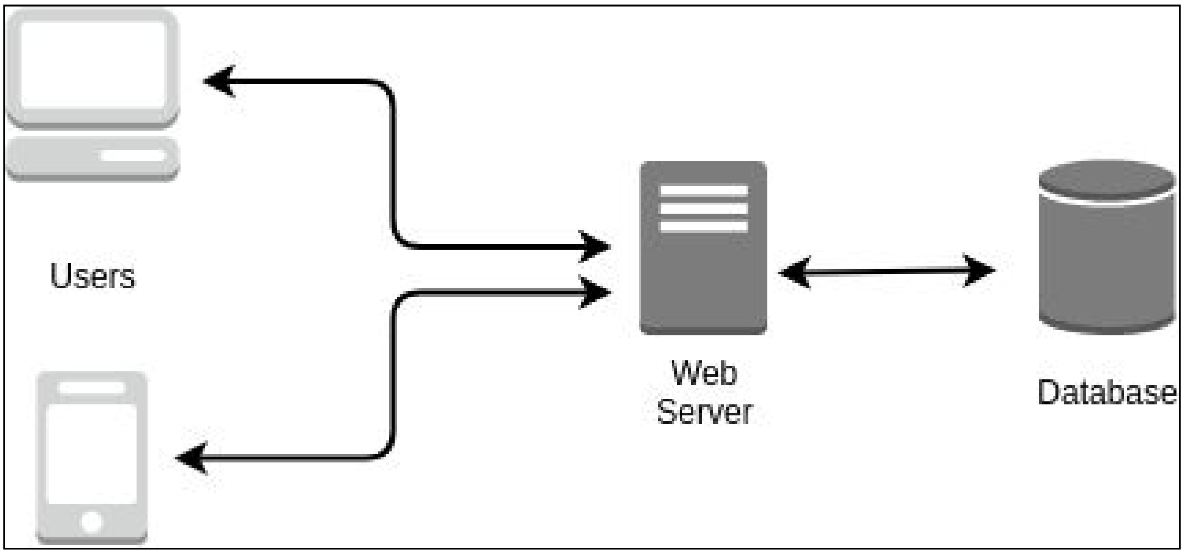


Fig. 2.1 System Architecture

Users
There are two types of user for this web app: the first is the election administrator, who will be primarily using the system on a desktop. The second user is the voter, who will most likely be using their smartphone or tablet to access the web app. I'm a firm believer in designing for mobile first, and then scaling out to desktop design. A responsive design is a must here to accommodate for both types of users.

Web Server
The web app will be hosted on pythonanywhere as I have plenty of experience with it and pythonanywhere have a neat feature that allows you to set up daily scheduled tasks. This is very handy for checking the database for inactive accounts, and sending reminder email when an election is active.

Database
Using a database cloud provider based outside the EU is a problem because right now the EU provides strong protection for personal data. If data belonging to EU businesses or

citizens is stored outside the EU, the transfer of that data needs to be secure with data protection requirements at the other end at least as strong as those in the EU.

The initial system architecture design presents a problem when using the database on pythonanywhere. This is because the pythonanywhere servers are outside the EU. For this reason We shall use Amazon Web Services (AWS) to host the database. The reason for this is because AWS allows you to pick the region for you database. Having the database in the EU will also speed up response times when querying the data.

3. Database Design

Database name: evote.

3.1 EER Diagram

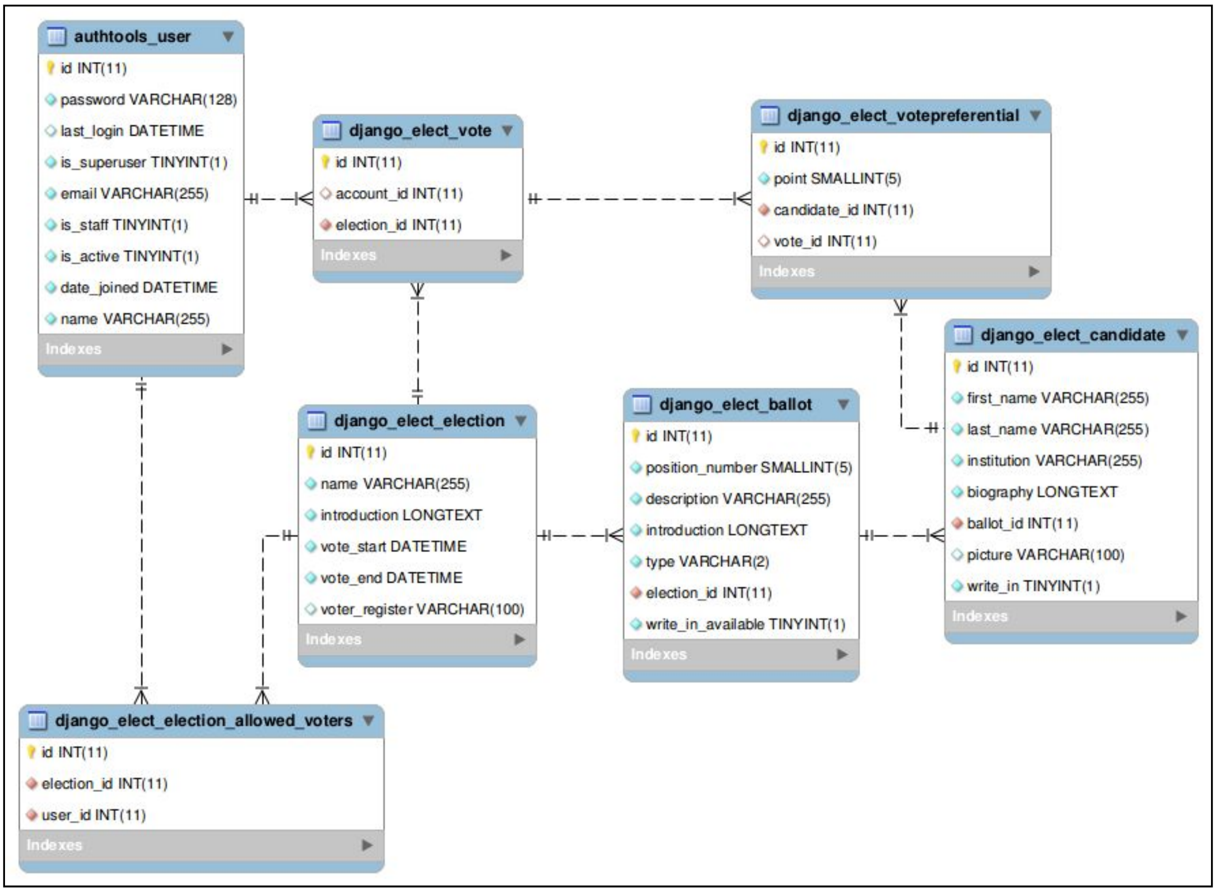


Fig 3.1 EER Diagram

Above in Fig 3.1 shows the enhanced entity relationship for the voting part of the web application.

3.2 Tables

This section contains details on the structure of the tables. Sample output is provided for most of the tables.

3.2.1 authtools_user

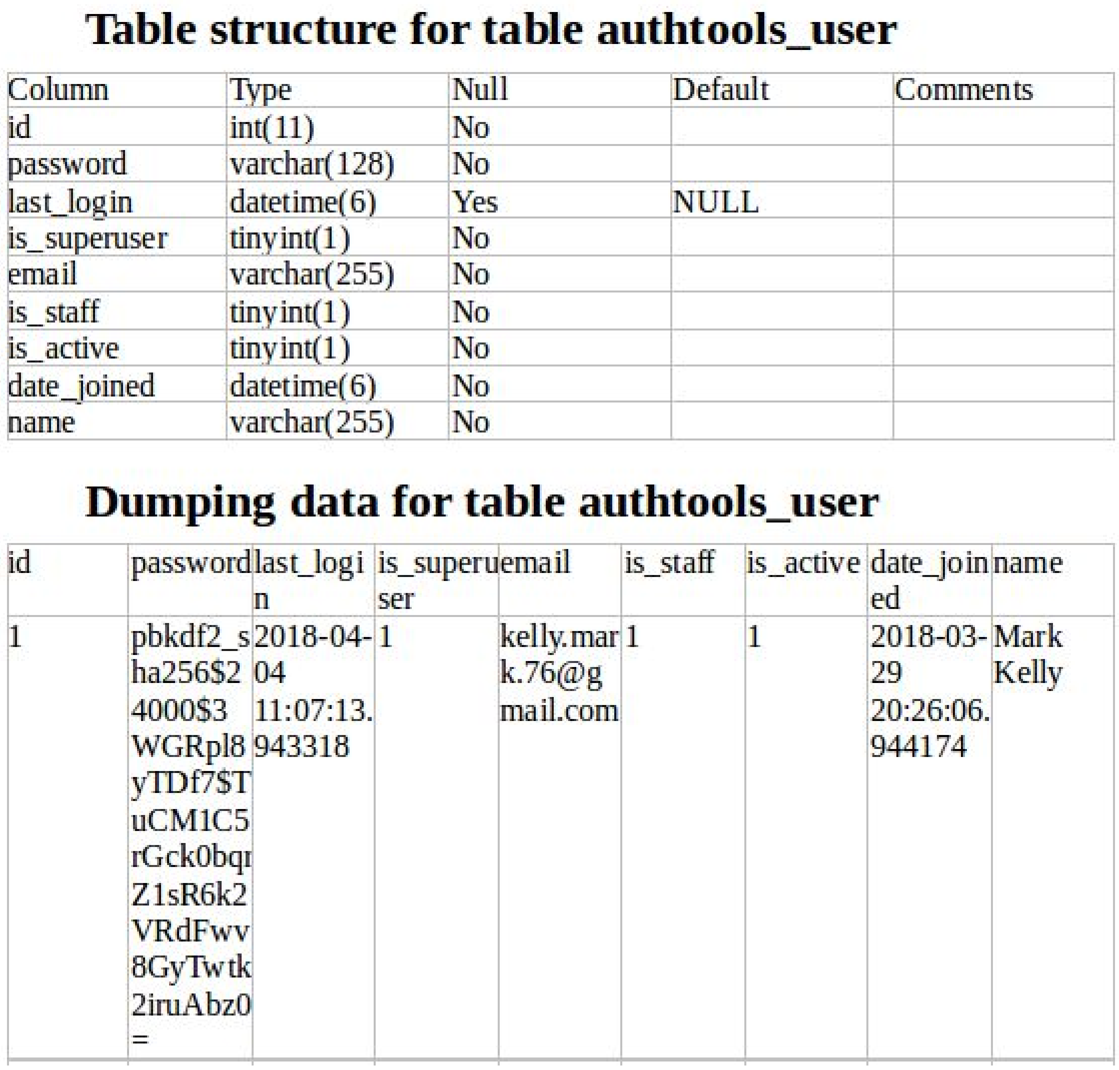


Table structure for table authtools_user

Column	Type	Null	Default	Comments
id	int(11)	No		
password	varchar(128)	No		
last_login	datetime(6)	Yes	NULL	
is_superuser	tinyint(1)	No		
email	varchar(255)	No		
is_staff	tinyint(1)	No		
is_active	tinyint(1)	No		
date_joined	datetime(6)	No		
name	varchar(255)	No		

Dumping data for table authtools_user

id	password	last_login	is_superuser	email	is_staff	is_active	date_joined	name
1	pbkdf2_sha256$24000$3WGRpl8yTDf7$TuCM1C5rGck0bqrZ1sR6k2VRdFwv8GyTwtk2iruAbz0=	2018-04-04 11:07:13.943318	1	kelly.mark.76@gmail.com	1	1	2018-03-29 20:26:06.944174	Mark Kelly

3.2.2 django_elect_election

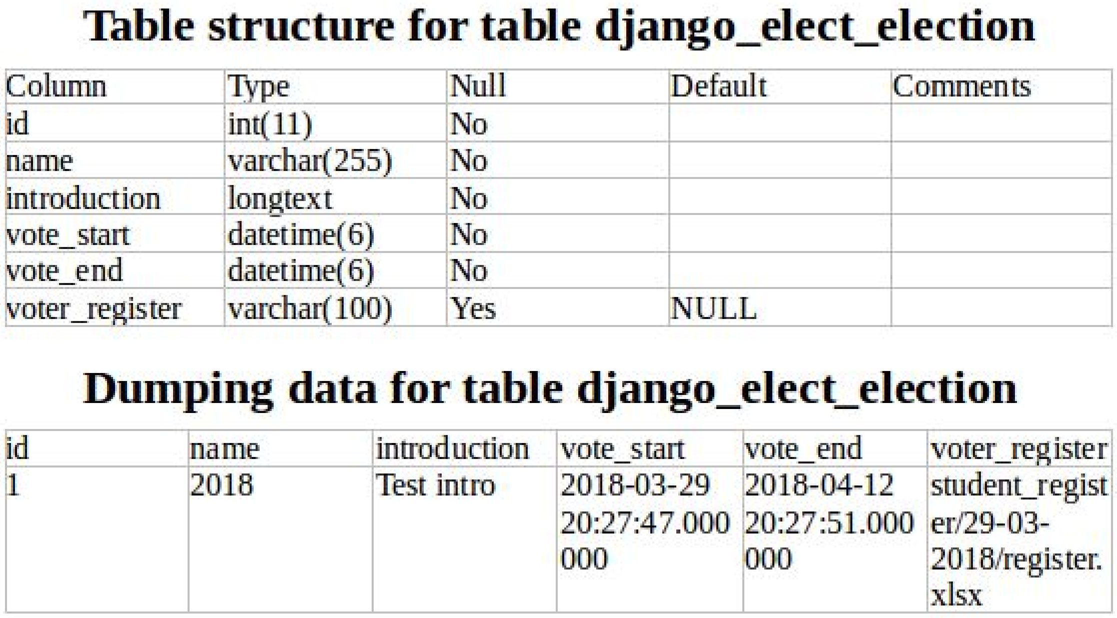


Table structure for table django_elect_election

Column	Type	Null	Default	Comments
id	int(11)	No		
name	varchar(255)	No		
introduction	longtext	No		
vote_start	datetime(6)	No		
vote_end	datetime(6)	No		
voter_register	varchar(100)	Yes	NULL	

Dumping data for table django_elect_election

id	name	introduction	vote_start	vote_end	voter_register
1	2018	Test intro	2018-03-29 20:27:47.000000	2018-04-12 20:27:51.000000	student_register/29-03-2018/register.xlsx

3.2.3 django_elect_ballot

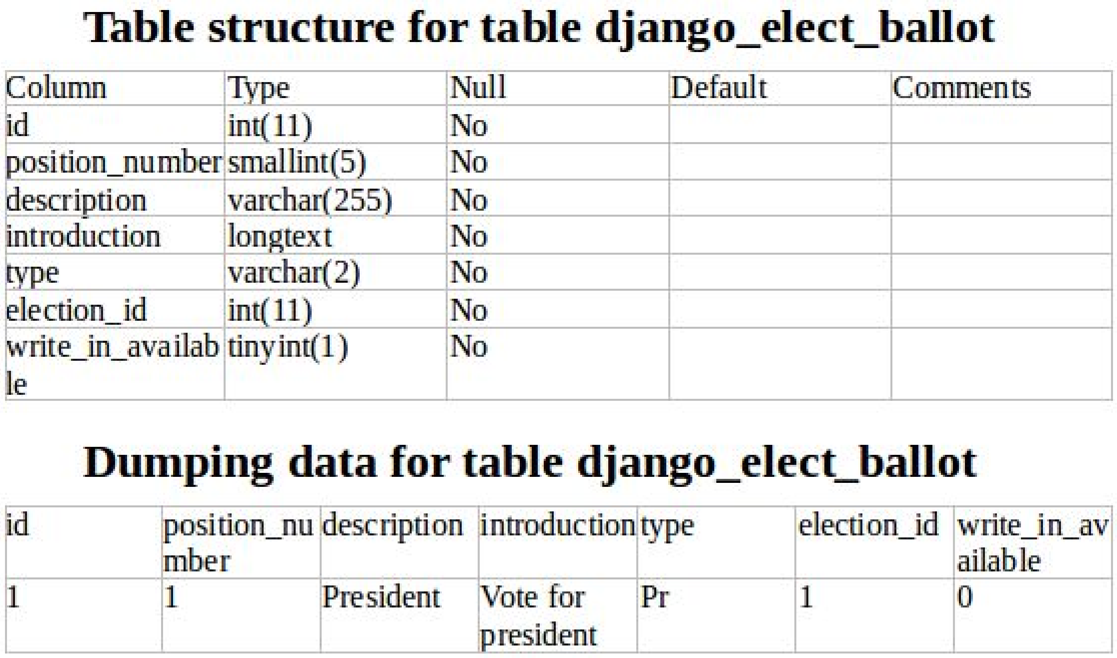


Table structure for table django_elect_ballot

Column	Type	Null	Default	Comments
id	int(11)	No		
position_number	smallint(5)	No		
description	varchar(255)	No		
introduction	longtext	No		
type	varchar(2)	No		
election_id	int(11)	No		
write_in_available	tinyint(1)	No		

Dumping data for table django_elect_ballot

id	position_number	description	introduction	type	election_id	write_in_available
1	1	President	Vote for president	Pr	1	0

3.2.4 django_elect_candidate

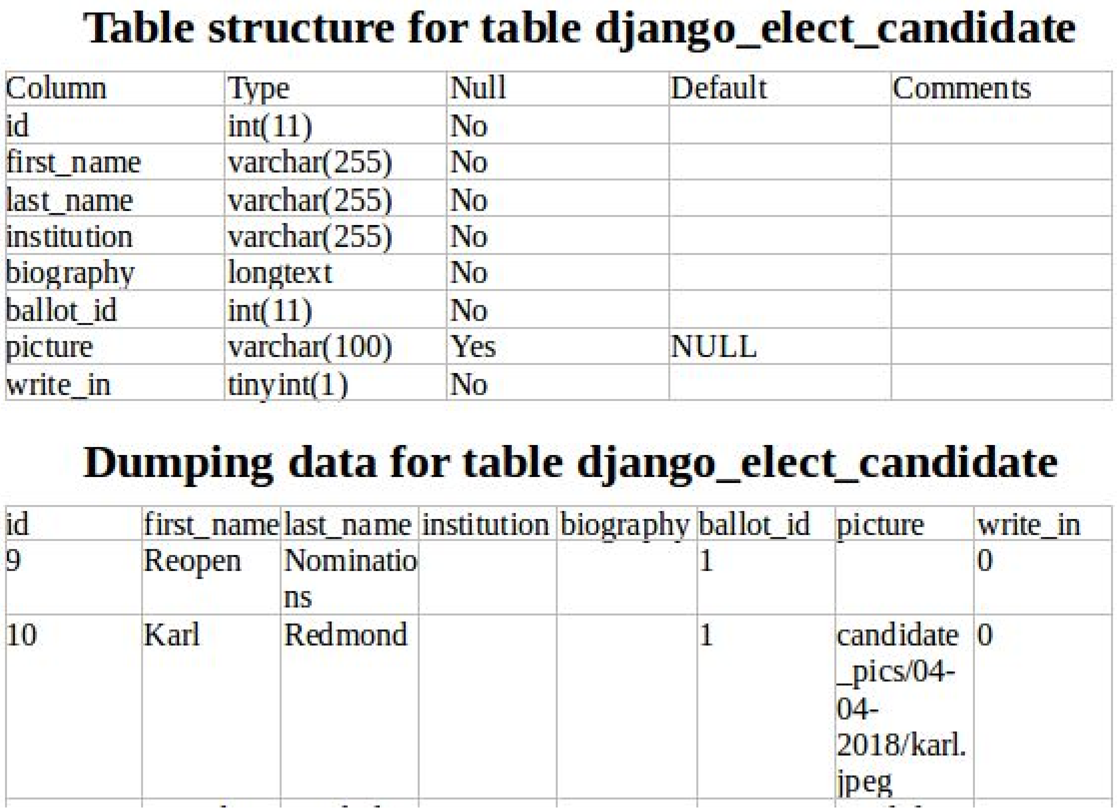


Table structure for table django_elect_candidate

Column	Type	Null	Default	Comments
id	int(11)	No		
first_name	varchar(255)	No		
last_name	varchar(255)	No		
institution	varchar(255)	No		
biography	longtext	No		
ballot_id	int(11)	No		
picture	varchar(100)	Yes	NULL	
write_in	tinyint(1)	No		

Dumping data for table django_elect_candidate

id	first_name	last_name	institution	biography	ballot_id	picture	write_in
9	Reopen	Nominations			1		0
10	Karl	Redmond			1	candidate_pics/04-04-2018/karl.jpeg	0

3.2.5 django_elect_election_allowed_voters

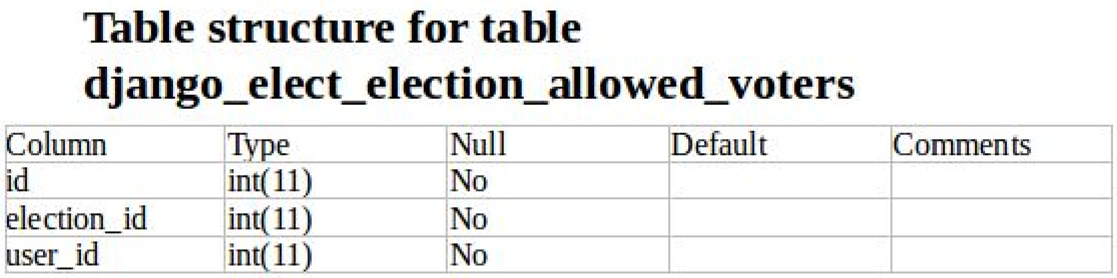


Table structure for table django_elect_election_allowed_voters

Column	Type	Null	Default	Comments
id	int(11)	No		
election_id	int(11)	No		
user_id	int(11)	No		

3.2.6 django_elect_election_vote

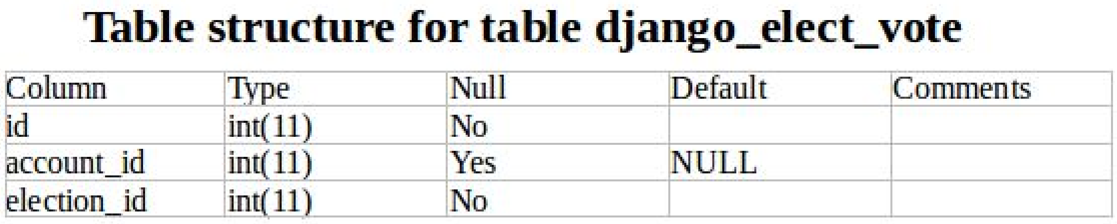


Table structure for table django_elect_vote

Column	Type	Null	Default	Comments
id	int(11)	No		
account_id	int(11)	Yes	NULL	
election_id	int(11)	No		

3.2.7 django_elect_election_votepreferential

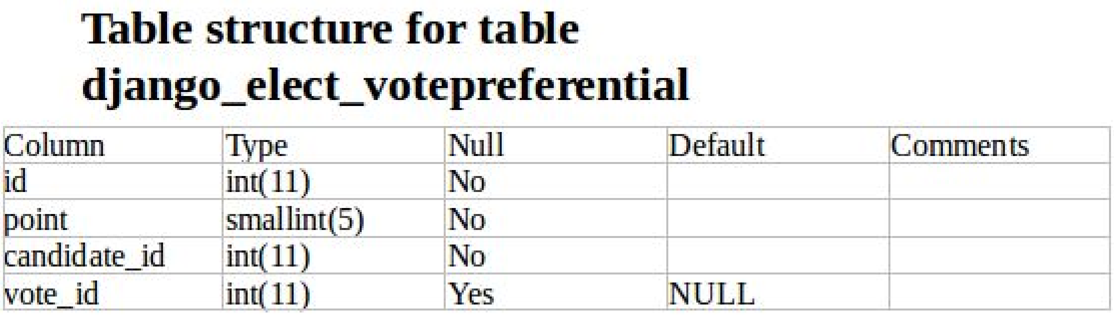


Table structure for table django_elect_votepreferential

Column	Type	Null	Default	Comments
id	int(11)	No		
point	smallint(5)	No		
candidate_id	int(11)	No		
vote_id	int(11)	Yes	NULL	

4. Use Case Diagram

The use case diagram below, Fig 4.1, was documented in the functional specification and is added here for reference.

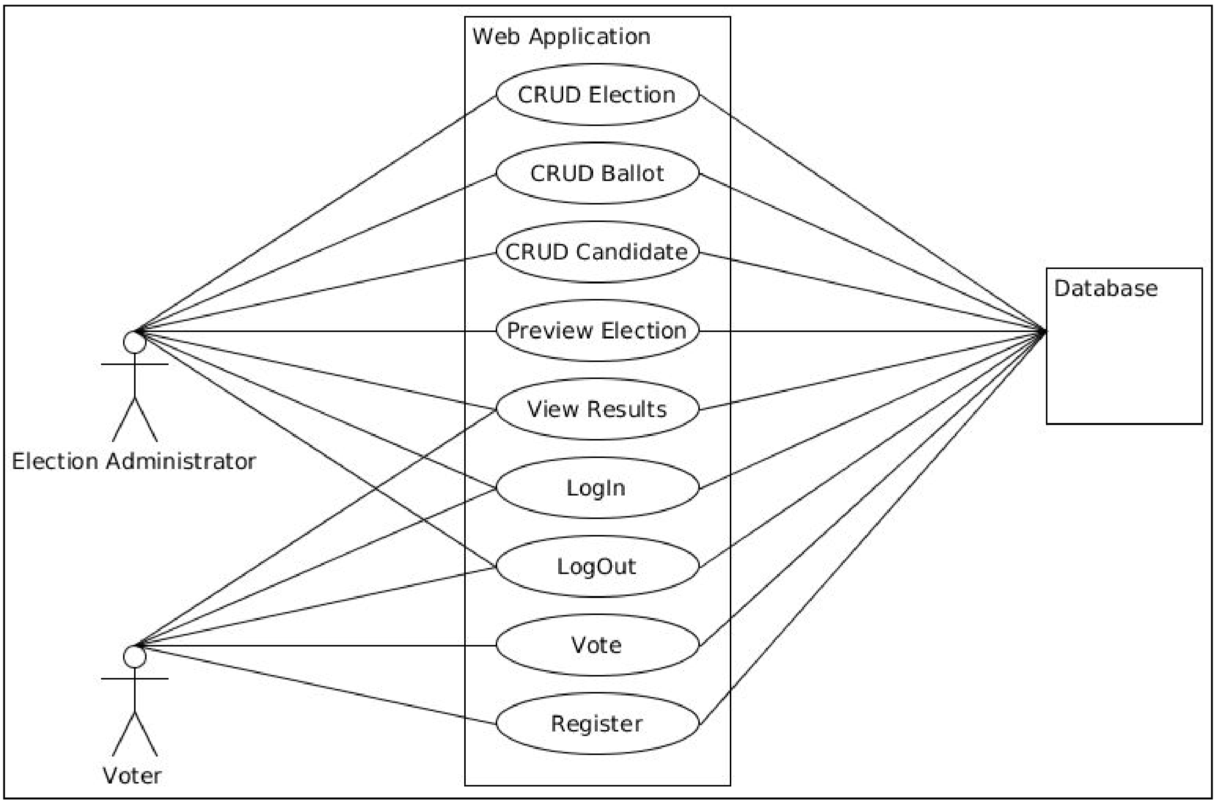


Fig. 4.1 Use Case Diagram

5. Detailed use cases

This section explains the brief use cases in more detail.

5.1 Detailed use case - CRUD Election (Election Admin)

Name: CRUD Election.
Actors: Election Administrator, Database.
Description: This use case begins when the election administrator wishes to create an election. The election administrator selects “Add Election” button. The system displays the “Election” page. The election administrator enters the details of the election and clicks “Save”. The details of an election include a name, introduction, start date, end date, and allowed voters. The system validates the details.

Main Success Story:

1. The election administrator clicks Add Election.
2. The system displays the Add Election page.
3. The election administrator enters the details of the election (Name, Introduction, Election start, Election end, Allowed voters).
4. The system validates the entered details.
5. The system stores the details of the election in the database.
6. The system indicates to the election administrator that the election was created successfully.
7. The system displays the Election page.

Alternatives:

4a. The details entered are invalid: display error and go back to step 3

5.2 Detailed use case - CRUD Ballot (Election Admin)

Name: CRUD Ballot.
Actors: Election Administrator, Database.
Description: This use case begins when the election administrator wishes to create a ballot. The election administrator selects the “Add Ballot” button. The system displays the “Add Ballot” page. The election administrator enters the details of the ballot and selects the “Save” button. The details of a ballot include the election for the ballot, a description, an introduction. The system validates the details.

Main Success Story:

1. The election administrator clicks Add Ballot.
2. The system displays the Add Ballot screen.
3. The system displays a list of available elections.
4. The election administrator chooses the election they want to add the ballot to from the list.

5. The election administrator enters the details of the ballot (Position number, Description, Introduction, Type).
6. The system validates the entered details.
7. The system stores the ballot details in the database.
8. The system indicates to the election administrator that the ballot was created successfully.
9. The system displays the Ballot page.

Alternatives:

3a. The list is empty: display message to add an election and end use case.

6a. The details entered are invalid: display error and go back to step 5

5.3 Detailed use case - CRUD Candidate (Election Admin)

Name: CRUD Candidate.

Actors: Election Administrator, Database.

Description: This use case begins when the election administrator wishes to add a candidate for an election.. The election administrator selects the “Add Candidate” button. The system displays the “Add Candidate” page. The election administrator enters the details of the candidate and selects the “Save” button. The details of an option include, the ballot, first name, last name, institution and profile picture. The system validates the details.

Main Success Story:

1. The election administrator clicks Add Candidate.
2. The system displays the Add Candidate screen.
3. The system displays a list of available ballots.
4. The election administrator chooses the ballot they want to add the candidate to from the list.
5. The election administrator enters the details of the candidate (First name, Last name, Institution, Profile picture).
6. The system validates the entered details.
7. The system stores the candidate details in the database.
8. The system indicates to the election administrator that the candidate was created successfully.
9. The system displays the Candidate page.

Alternatives:

3a. The list is empty: display message to add a ballot and end use case.

6a. The details entered are invalid: display error and go back to step 5

5.4 Detailed use case - Preview Election (Election Admin)

Name: Preview Election.

Actors: Election Administrator, Database.

Description: This use case begins when the election administrator selects the “Preview

Election” button. The system retrieves the details of the election from the database and displays them on the screen.

Main Success Story:

1. The election administrator clicks Preview Election.
2. The system displays the Preview Election screen.
3. The system displays the latest election.

Alternatives:

3a. There are no elections: display info message and end use case.

5.5 Detailed use case - View Results (Election Admin)

Name: View Results.
Actors: Election Administrator, Database.
Description: This use case begins when the election administrator selects the “View Results” button. The system retrieves the results of the election from the database. The system displays the results on the screen.

Main Success Story:

1. The election administrator clicks View Results.
2. The system displays the Results screen.
3. The system displays the results of past elections.

Alternatives:

3a. There are no elections: display info message and end use case.

5.6 Detailed use case - View Results (Voter)

Name: View Results.
Actors: Voter, Database.
Description: This use case begins when the voter selects the “View Results” button. The system retrieves the results of the election from the database. The system displays the results on the screen.

Main Success Story:

1. The voter clicks View Results.
2. The system displays the Results screen.
3. The system displays the results of past elections.

Alternatives:

3a. There are no elections: display info message and end use case.

5.7 Detailed use case - Login (Election Admin)

Name: Login.
Actors: Election Administrator, Database.
Description: This use case begins when the election administrator wishes to login to the system. The election administrator enters their login credentials. Credentials include email and password. The system validates the credentials with the database.

Main Success Story:

1. The Election administrator selects login.
2. The system displays the Login screen.
3. The election administrator enters their login details (Email, Password)
4. The system validates the login details
5. The system displays the home page with access to the admin interface.

Alternatives:

4a. The details entered are invalid: display error and go back to step 3.

5.8 Detailed use case - Login (Voter)

Name: Login.
Actors: Voter, Database.
Description: This use case begins when the Voter wishes to login to the system. The Voter enters their login credentials. Credentials include email and password. The system validates the credentials with the database.

Main Success Story:

1. The Voter selects login.
2. The system displays the Login screen.
3. The Voter enters their login details (Email, Password)
4. The system validates the login details
5. The system displays the home page with access to the Voter interface.

Alternatives:

4a. The details entered are invalid: display error and go back to step 3.

5.9 Detailed use case - Logout (Election Admin)

Name: Logout.
Actors: Election Administrator, Database.
Description: The use case begins when the election administrator wishes to logout of the system. The election administrator clicks the “logout” button. The web application logs the election administrator out.

Main Success Story:

1. The Election administrator selects Logout.
2. The system clears the session.
3. The system displays the home page.

5.10 Detailed use case - Logout (Voter)

Name: Logout.
Actors: Voter, Database.
Description: The use case begins when the Voter wishes to logout of the system. The Voter clicks the “logout” button. The web application logs the Voter out.

Main Success Story:

1. The Voter selects Logout.
2. The system clears the session.
3. The system displays the home page.

5.11 Detailed use case - Vote (Voter)

Name: View Results.
Actors: Voter, Database.
Description: This use case begins when a voter wishes to vote on an election. The system displays the Vote page. The voter fills in the ballot. The system validates the ballot.

Main Success Story:

1. The voter selects Vote.
2. The system displays the Vote screen.
3. The voter fills in the ballot by selecting choices from drop down list.
4. The system asks for confirmation.
5. The voter selects yes.
6. The system validates the entered choices.
7. The system displays the Home page.
8. The system displays a thank you for voting message.

Alternatives:

6a. No choices were made: display error message and go back to step 3.
6b. The same choice was made for more than one candidate: display error message and go back to step 3.

5.12 Detailed use case - Register (Voter)

Name: Login.
Actors: Voter, Database.
Description: This use case starts when the voter clicks on the registration link on the “Login” page. The system displays the registration page. The voter enters their details. Details for the registration include a password. The system checks if voter is on allowed voters list. The

system validates the details. A confirmation email is sent to the voter. The voter confirms the email.

Main Success Story:

1. The Voter selects register.
2. The system displays the register screen.
3. The Voter enters their details (Email, Password, Confirm Password).
4. The system validates the details.
5. The system checks if the voters email address is on the allowed voters list for the current election.
6. The system checks if the voter is registering in the correct time frame (e.g the election hasn't started).
7. The system stores the voters details in the database with a flag to say their account is inactive.
8. The system displays a message telling the voter to check their emails for a confirmation email.
9. The system sends the voter an email with a link to complete the registration, with a token attached.
10. The system receives the token.
11. The system updates the voters details in the database, setting the flag to active.
12. The system displays the home page with a success message.

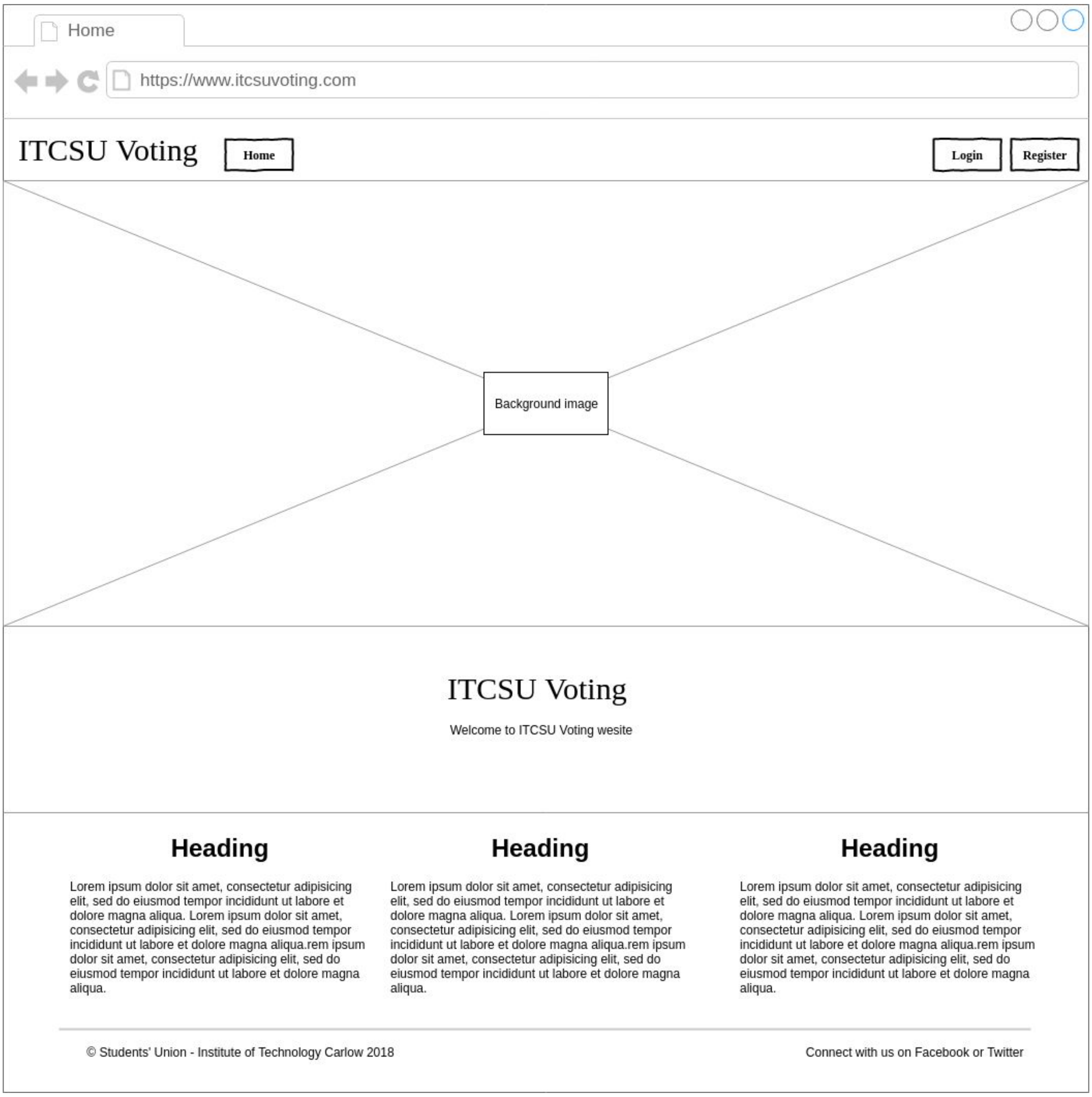
Alternatives:

4a. The details entered are invalid: display error and go back to step 3.
5a. The voters email address is not on the allowed voters list: display error message and end use case.
6a. Registration is closed for the current election: display error message and end use case.

6. User Interface Design

This section contains a selection of wireframes loosely based on how we want the user interface to look like. We shall only create wireframes for the main pages. The wireframes are a great help when it comes to implementing the pages in code.

6.1 Home page wireframe



ITCSU Voting

Welcome to ITCSU Voting wesite

Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.rem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.rem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Heading

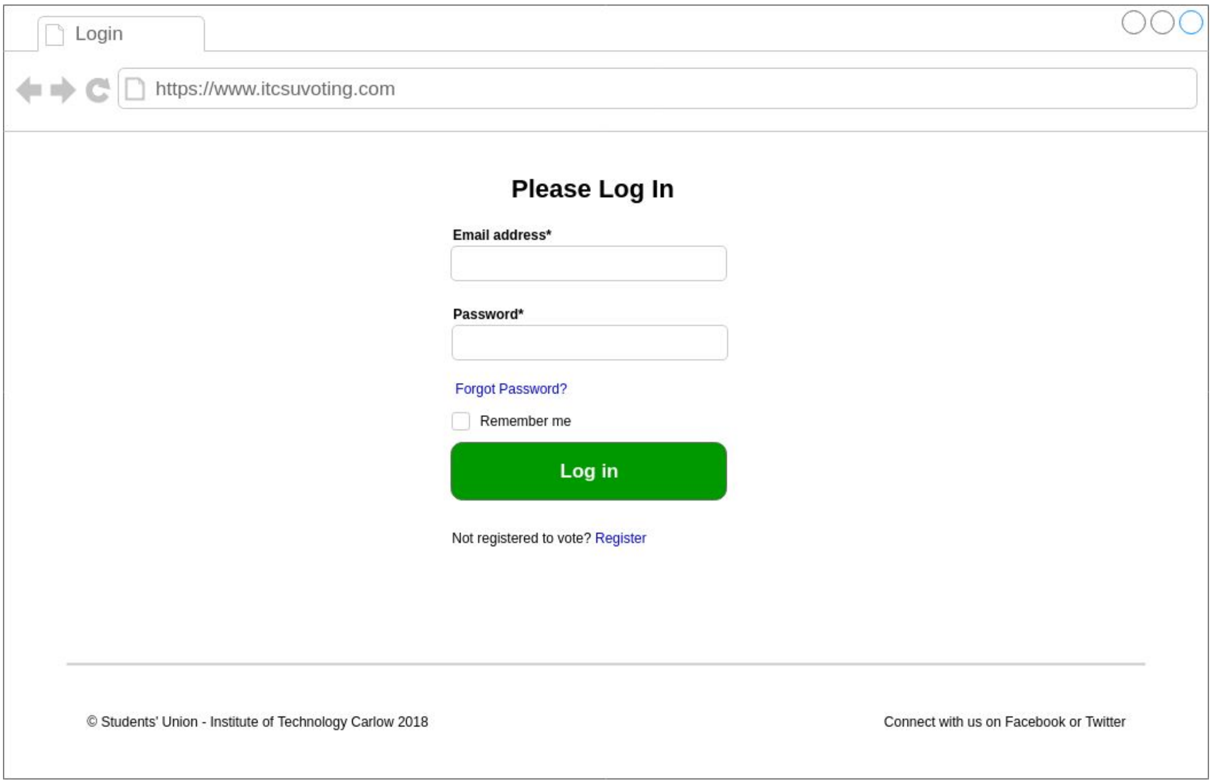
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.rem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

© Students' Union - Institute of Technology Carlow 2018

Connect with us on Facebook or Twitter

Fig. 6.1 Home page wireframe

6.2 Log in page wireframe



Please Log In

Email address*

Password*

Forgot Password?

□ Remember me

Not registered to vote? Register

© Students' Union - Institute of Technology Carlow 2018

Connect with us on Facebook or Twitter

Fig. 6.2 Log In wireframe

6.3 Register page wireframe

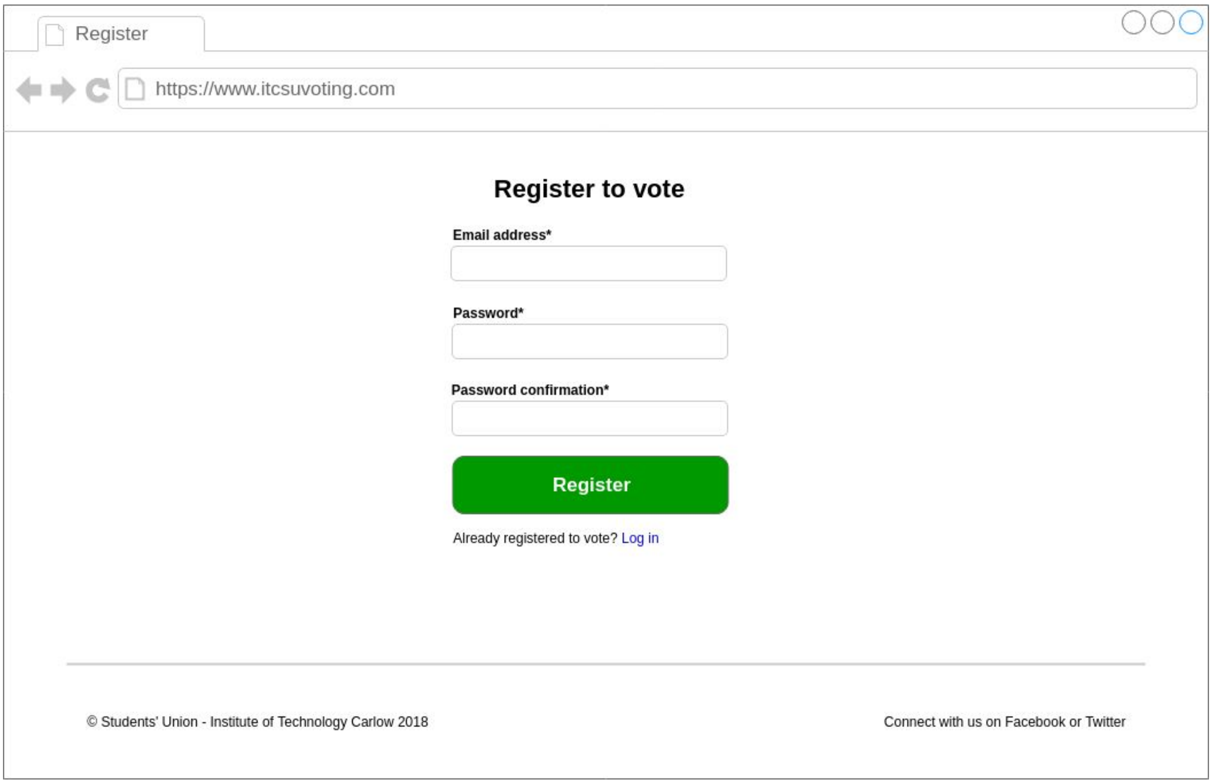
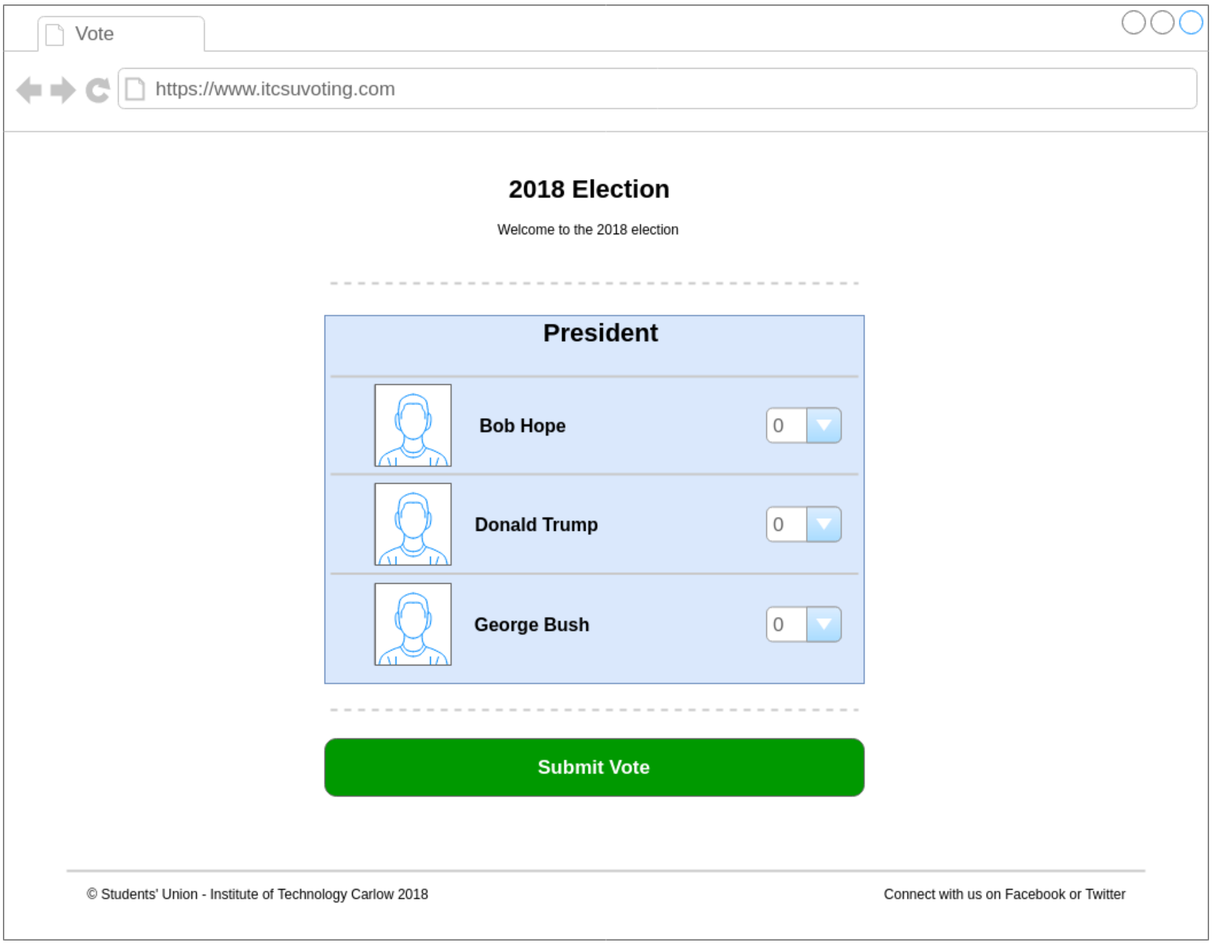

Register
https://www.itcsuvoting.com
Register to vote
Email address*
Password*
Password confirmation*
Register
Already registered to vote? Log in
© Students' Union - Institute of Technology Carlow 2018
Connect with us on Facebook or Twitter

Fig. 6.3 Register wireframe

6.4 Vote page wireframe



2018 Election

Welcome to the 2018 election

President

Bob Hope	0
Donald Trump	0
George Bush	0

© Students' Union - Institute of Technology Carlow 2018

Connect with us on Facebook or Twitter

Fig. 6.4 Vote wireframe