**ITMD 511**

**Application Development Methodologies**

**Deborah Barndt**

**Erik Escobar-Ocon**

**Spring 2018**

This document is a high level traceability matrix of what will be tested within each requirement. Obviously, one can have many test cases for each requirement, thus ensuring a huge test plan. For this project, we chose to do no more than five test cases per environment. A real traceability matrix would map out the cases to the requirement in a more formal fashion. This means that the requirement will have a certain ID or naming convention to quickly map to the requirement. Since the matrix isn’t in our deliverables, we will provide this high level list of all of the test cases we came up with.

In our test plan, the following format will be used for requirements: ***REQ\_Functionality\_RequirementNumber##.***

For the test cases, the format is: ***TC\_Functionality\_RequirementNumber##\_TestCase##.***

1. **Registration/Login Requirements (REQ\_REG\_##):**
2. Users will register by entering their personal info(street address in format XXXX, 10 digit phone number in xxx-xxx-xxxx format, dob in the format mm/dd/yyyy), voter id number, registration code, password, and district number.

1.1.1 - Enter address in the incorrect format.

1.1.2 - Enter phone number in the incorrect format.

1.1.3 - Enter DOB in the incorrect format.

1.1.4 - Try to register with missing field

1.1.5 - Happy path

1. Users will submit 2 factor authorization to login with the first phase using the username and password, and the second phase by entering a code provided via sms or email with a time limit of 3 minutes to enter the given code

1.2.1 - Enter invalid username format

1.2.2 - Enter valid username, incorrect password

1.2.3 - Enter valid username, valid password, invalid code

1.2.4 - Enter valid username, valid password, surpass the window time for the code

1.2.5 - Happy Path - Enter valid username, valid password, valid code from SMS

1.2.6 - Happy Path - Enter valid username, valid password, valid code from email

1. For security purposes, user session will end after 5 minutes of inactivity and give the user a notification message that the session will end in 2 minutes due to inactivity

1.3.1 - Log in successfully, stay inactive for 4 minutes and 55 seconds, then perform an inaction

1.3.2 - Log in successfully, stay inactive for 5 minutes

1.3.3 - Log in successfully, stay inactive for 2 minutes, receive inactive notification

1.3.4 - Log in successfully, stay inactive for 2 minutes, receive notification, perform action

1. For every new session, the user must re-login with all their credentials again in order to begin a new session (no stored sessions)

1.4.1 - Log in successfully, get kicked out of current session, re-login again

1.4.2 - Log in successfully, become inactive from the app in the background, re-log in

1.4.3 - Log in successfully, kill the app, re-login

**2. Voting Requirements(REQ\_VOTE\_##):**

1. Users can only vote while on the wireless network that is connected to the polling place, and within X feet from the premise (GPS location based)

2.1.1 - Try to submit vote while on the network but outside the GPS boundary

2.1.2 - Try to submit vote while on the network and inside the GPS boundary

2.1.3 - Try to submit vote off of the network but inside the GPS boundary

2.1.4 - Try to submit vote off of the network and outside the GPS boundary

1. Users can select to vote for a party with one-click selection by clicking on the red colored elephant labeled republican or the blue colored donkey labeled democrat

2.2.1 - Click to vote for all republicans, submit - verify votes

2.2.2 - Click to vote for all democrats, submit - verify votes

2.2.3 - Click to vote for all republicans, edit vote, click to vote for all democrats

2.2.4 - Click to vote for all democrats, edit vote, choose 1-2 republicans

2.2.5 - Try to submit empty ballot

1. Users can save vote selections for the current ballot, but can not submit multiple times and the information will be dropped after the polling window closes if it is not submitted

2.3.1 - Submit ballot, try to submit another ballot

2.3.2 - Fill ballot, save ballot, edit ballot, save ballot, submit ballot, verify

2.3.3 - Fill ballot, save ballot, submit ballot, verify

2.3.4 - Fill ballot, save ballot, try to submit outside of polling window

2.3.5 - Fill ballot, edit ballot, save ballot, close app, return to app, verify persistence

1. Ballot selections will be multiple choice with oval-shaped icons, and write-in selections will be text-based in Arial font

2.4.1 - Verify ballot selections are oval shaped

2.4.2 - Verify user can write-in selections

2.4.3 - Verify written-in ballots are persisted

2.4.4 - Verify written-in ballots are submit successfully

2.4.5 - Verify written-in ballots are persisted between edits

1. User will receive notifications at 2 hours, 1 hour, and 15 minutes until closing of the poll window

2.5.1 - Verify user receives notification with 2 hours left in window

2.5.2 - Verify user receives notification with 1 hour left in window

2.5.3 - Verify user receives notification with 15 minutes left in window

2.5.4 - Verify user receives notification when window opens

2.5.5 - Verify user receives notification when window closes

1. User will receive a window to submit their poll selections (look up polling times for real elections)

2.6.1 - Verify user is informed of polling window

2.6.2 - Verify user is informed of the next polling window

2.6.3 - Change phone time zone, verify user is informed of polling window for their local poll

2.6.4 - Verify poll window is based on system time for the election and not based on user phone time

**3. Submission Requirements(REQ\_SUB\_##):**

1. If users leave the premise (network or outside of GPS range), the submission will be forfeited and they will not be allowed to start another session until they get back on the network

3.1.1 - Fill ballot out, go outside of GPS boundary, try to submit vote

3.1.2 - Fill ballot out, disconnect from network, try to submit vote

3.1.3 - Fill ballot out, go outside of GPS boundary, try to submit vote, go back within range, submit vote, verify vote

3.1.4 - Fill ballot out, disconnect from network, try to submit vote, go back on network, submit vote, verify vote

1. Users will receive confirmation/verification message upon successful vote submission, and receive a digital badge that will be added to their account

3.2.1 - Submit valid vote, verify confirmation

3.2.2 - Submit valid vote, verify verification message

3.2.3 - Submit valid vote, verify badge received

3.2.4 - Submit valid vote, verify no new ballots can be submitted

3.2.5 - Submit vote, deny, confirmation, submit re-vote, verify confirmation, verify verification message, verify submission

1. Users will confirm vote selections before submitting.

3.3.1 - Submit ballot, verify confirmation

3.3.2 - Submit ballot, deny confirmation, re-submit new ballot, verify new confirmation

3.3.3 - Submit ballot, deny confirmation, re-submit new ballot, deny new confirmation, re-submit new ballot

3.3.4 - Submit ballot, deny confirmation, re-submit same ballot, verify new confirmation, verify successful vote entry

1. Users will only be allowed to submit 1 ballot.

3.4.1 - Submit ballot, verify no new ballot can be submitted

3.4.2 - Submit ballot, deny ballot, submit same ballot, verify no new ballot can be submitted

3.4.3 - Submit ballot, deny ballot, submit new ballot, verify no new ballot can be submitted

3.4.4 - Submit ballot, close app, verify no new ballot can be submitted

3.4.5 - Submit ballot, get kicked out (inactivity), verify no new ballot can be submitted

**4. Error Handling/Security Requirements(REQ\_ERR\_##):**

1. Upon frozen app or crashed app, new session will be restarted and error report will be sent

4.1.1 - Verify when app freezes, new session is started

4.1.2 - Verify when app crashes, new session is started

4.1.3 - Verify error report is sent when app freezes

4.1.4 - Verify error report is sent when app crashed

4.1.5 - Verify user is informed of crashed/frozen app on new session

1. Encrypts all the stored data to ensure data integrity and security

4.2.1 - Verify user registration info in encrypted

4.2.2 - Verify user credentials are encrypted

4.2.3 - Verify user ballots are encrypted

4.2.4 - Verify no leakage of information in logs

1. Application will not allow users to share on social media profiles

4.3.1 - Verify user is not allowed to share ballot on social media

4.3.2 - Verify user is not allowed to access their old ballot