**User Requirements**

1. The policy voter app will allow users to access their account and view policies currently available for voting.
2. Users will be able to post a vote to any policies currently available for them for voting.
3. Users will be able to propose new policies for voting.
4. The policy voter app will notify all users when a new policy is available for voting
5. Users will be able to view voting results of previous policies

**System Requirements**

1. All interactions between the mobile client and server is facilitated by REST API
2. Only authenticated user will have access to the mobile application
3. Server will not fulfill request from a mobile client not signed in by an authenticated user
4. When query for current policies is made only current policies not voted by user yet is sent
5. All details for a new policy, title and description must be provided for a valid request to be made
6. When a new policy is entered into the Firestore DB a notification is sent to all mobile clients through Firebase Cloud Messaging
7. When the collective votes on a policy is equal to the number of total users the status for the policy will be changed to COMPLETE
8. When a policy status is set to COMPLETE a notifications is sent to all mobile clients through Firebase Cloud Messaging

**Risk Analysis**

1. Unauthorized user gains access to accounts and post fraudulent votes. Mitigation: Verify each request are made from authorized user account
2. Not all team members are experienced in tech stack. Mitigation: Those who are experienced pass knowledge and assist team members
3. If tasks are not completed in a timely manner it can slow down development for the whole team. Mitigation: Create schedule and deadlines for when tasks should be completed to keep everything on track.