Specification

# Functional Specification

## Overview

myDebate is a service that allows users to participate in a democratic debate or discussion - to achieve a goal of agreement on a particular subject.

The system forces users to vote on a goal at the end of each time period (which is **30 minutes** by default). The goal (an issue that is to be debated) is specified when the debate is created. If all (or a specified proportion) of the participants agree when the vote is cast, then the goal has been achieved and the debate can be ended. Otherwise, the discussion continues for another period.

## User Stories/Scenarios

The user stories or scenarios specified in this specification will help to define how users interact with the system.

### Scenario 1: Jimmy

Jimmy is an 18-year-old college student in London, studying the arts, and has a keen interest in computers. Jimmy is a participant in a number of online communities, where he often has heated discussions, supporting his views on his choice of operating system – Mac OS X. Jimmy enjoys making fun of people that use Microsoft Windows or Linux, and often gets himself into arguments in forums or message boards over the subject.

While browsing his usual online forums, Jimmy came across a thread that simply stated: “Which operating system is better, Windows or OSX? Debate now on myDebate!”. He was intrigued, and clicked the link to the myDebate website.

In order to take part in the debate, Jimmy was required to create an account on the site, which he did. He clicked the link that was sent in an email to his account, and logged in immediately. He then joined the debate about OSX and Windows.

He found that there was a timer counting down in the corner of the screen, and people were putting forward points in an attempt to win the vote of the other participants. When the counter reached zero, a screen appeared asking Jimmy to vote – he picked OSX straight away, and then saw a pie chart that changed as the votes from the other participants arrived. He was impressed at the responsiveness of the system, which seemed to react almost instantly to his clicks and keypresses.

After two periods, the vote was starting to show a majority towards OSX, so Jimmy remained in the debate and put forward his own arguments as to why he likes OSX so much. The next time a poll was taken, 68% of the participants voted in favour of OSX, and the debate was declared finished with a majority.

Jimmy enjoyed putting forward his points on myDebate, and bookmarked the list of debates in his browser so that he could keep checking the latest ones.

## Flowchart

This flow chart gives a high-level overview of how the user interacts with the system, and vice-versa.



## Functional Requirements

1. The system should have a web-accessible interface
2. The system should allow users to login and logout
3. The system should present users with a list of available debates
4. The list of debates should show how many users are currently participating in each
5. The system should allow users to enter a debate, and chat with others in that room in near-real time. This requires the user to be logged-in
6. The system should allow users to see who else is in the debate
7. The system should allow users to leave a debate, and return to the list of available debates
8. The system should use an asynchronous method of communication within the debate (i.e. the page should not refresh when a new message is available.
9. The system should allow users to sign-up for a user account.
   1. Users must specify their full name
   2. Users must choose a username and password
   3. Users must enter their email address
   4. The system should send the user an email, which includes a link that the user must click in order to “activate” their account
   5. Before the user has clicked the link in their email, the account should be inactive - so the user cannot login
10. Users should have a profile, containing their personal information shared with other users.
11. Users should be able to edit the information in their profile.
12. Users shoud be able to change their password.
13. Users should be able to create debates.
14. Users should be required to input a “poll” when creating a debate. This should represent the “objective” of the debate - all members should agree on the issue named in the poll. The poll also requires a number of choices, upon which the users can vote.
15. Users should be able to choose the length of the “period” when creating a debate (a default value of 30 minutes should be provided).
16. Users should be able to choose the time at which a room “locks” - so that no new members can join, before a poll is due. This means that new users cannot “hijack” a debate a few seconds before a vote.
17. The system should switch to a vote at the end of each specified time period.
18. When in the voting mode, users should not be able to enter messages in the conference. Everyone is required to vote on the poll specified at creation time.
    1. One the poll has been completed, the person who is unofficially in charge of the room has the choice over whether to end the debate or go to another period
    2. The person who is in “charge” is decided by who has been in the room the longest
19. The system should allow users to search for debates quickly and easily, so that sorting through a large amount of debates is not necessary.
20. The system should have a Frequently Asked Questions section.

## Non-Functional Requirements

1. The web interface should be accessible from any platform, using either of **two** main browsers – Firefox and Google Chrome.
2. The system should be secure in that unauthenticated users cannot access debates
3. The system should be available
4. The system should respond to requests in a reasonable time period
5. The FAQ should be comprehensive, reflect relevant questions that users would ask, and aid the user in using the system as a whole.