DebateMate - Release Report

Version 3.0

Sharp Team

Matthew Weihl | Brianna Frye | Benjamin Brown | Austin Nabors | Semere Tadesse

https://debate-mate.herokuapp.com/

https://github.com/SharpTeam17/DebateMate

1 Project Title, Group Name

The project is entitled “DebateMate”. The name of the team is “Sharp Team”.

2 Project Website and GitHub URL

The project can be accessed at the URL http://debate-mate.herokuapp.com/

3 Version

DebateMate has gone through three sprints worth of development. DebateMate and this report is at version 3.0.

4 Project Summary and Vision

4.1 High Level Overview

DebateMate is a website that allows users to participate in a digital debate forum and discuss various topics. Users can view the topic of the day, join one side of the debate, and post arguments in favor of their side. Users can also vote and score the arguments of each side to determine which side has a better argument.

4.2 Goals

The team’s main goal is to provide users with an environment that is conducive to debates. This product will provide tools enabling users to submit well-thought out arguments, such as requiring debaters to include sources with their arguments. Moderators will have tools, such as the ability to ban users, remove comments, and remove arguments, that help them keep the debate free from abuse and off-topic responses. The product should allow users to better participate in a debate and alleviate issues that arise when trying to debate in the comment sections of other websites. Users are allowed to comment and refute specific arguments in the comments of an argument.

5 Problem Statement

5.1 Original Problem Statement

Debate is a pastime enjoyed by many people across the world. The formality of these debates can range from a simple conversation on the internet to a formal, moderated debate

competition. On the internet, debates are prevalent. The comment sections of news articles, Facebook posts, and even Tweets are commonly transformed into impromptu debates. These media are not designed with debates in mind, so numerous problems can occur when they are used to host debates. The user interface for comment-style sites makes it difficult to trace arguments and rebuttals—the flow of information is not clear. These sites also lack any implicit or explicit agreement that good debate etiquette and structure will be followed.

Because these media are inappropriate for debate, they are prone to abuse, breaking down into echo chambers, and nonproductive squabbles. This is plainly evident when sites such as news outlets close their comment sections due to issues. When this happens, there is almost always backlash. This shows that there is a strong desire among people to debate, but no good site to provide structure and moderation.

In order to solve this problem, our team will create a web application, DebateMate, that provides a structured, moderated, and focused venue for online debate. Through the use of clear rules, moderation, and peer feedback systems, users will be able to engage in constructive debates.

5.2 Changes to Desired Deliverables

Sprint 1

There were no modifications to the problem statement following Sprint 1.

Sprint 2

There were no modifications to the problem statement following Sprint 2.

Sprint 3

There were no modifications to the problem statement following Sprint 3.

6 Team Profile

6.1 Individual Strengths and Interests

6.1.1 Matthew Weihl

Matthew Weihl's strength in this project is a general understanding of Scrum and Agile Development. He currently is an intern with ALSAC/ St. Jude and is a part of their SharePoint team, which is organized via Scrum. He also has a general understanding of many programming languages, including C#, C/C++, JavaScript, Java, and Python. Matthew's ultimate goal is to someday work in a Big-Four technology company, such as Google, Facebook, or Microsoft. His prior experience with working with Scrum and Agile at his workplace

has allowed him to understand better how to run Scrum ceremonies, organize the team, as well as help explain any concepts to any other team members.

6.1.2 Benjamin Brown

Benjamin Brown's main strength is his understanding of various libraries and tools one can implement along with a vast knowledge of programming languages. His interests include game development, compiler theory, and operating system design. As a career path, Ben intends to become a software developer in the gaming industry or a Fortune 500 company where see fit. Benjamin's strength is his prior understanding of the Django framework from past experience which helped others initially get up to speed.

6.1.3 Austin Nabors

Austin Nabors strengths include web application development experience, along with knowledge of various programming languages, like Java, Python, TypeScript, etc. Interests include: Studying programs dealing with self driving cars and traffic flow. His career goal is to become a CIO or something similar.

6.1.4 Brianna Frye

Brianna Frye's relevant strengths include database design, experience with data-processing using Python, and knowledge of query construction. Interests include data/statistical analysis and virtualization technologies like VMWare or Citrix. Brianna's career goal is to become a software developer specializing in desktop applications/utilities.

6.1.5 Semere Tadesse

Semere Tadesse's strengths include [placeholder]. Interests include [placeholder 1], [placeholder 2]...

6.2 Roles and Responsibilities

6.2.1 Product Owner

Matthew Weihl is the Product Owner and Team Lead for "Sharp Team". His responsibilities includes leading Scrum Ceremonies (Standups/ Scrum meetings, Planning, Review, and Retrospective), building out the Product and Sprint Backlog, communicating with the customer (i.e. Professor Yu), validating user stories against acceptance criteria, as well as helping with any team members with any possible bugs that occur during development.

6.2.2 Developers

Benjamin Brown is a developer and is responsible for primarily the front-end of the project, including creating views and forms for the project and developing the UI for the chat interface.

Austin Nabors is a developer and is responsible for front end design, including structures for displaying data, along with ensuring the links in-between pages are correct and intuitive, as well as managing view mapping.

Brianna Frye is a developer and has the primary responsibility of back-end development tasks. These include creating database models, creating queries from form data, and ensuring users can access only the pages and commands appropriate to their role.

Semere Tadesse is a developer and has the primary responsibility of back-end development tasks. These include creating database models, creating queries from form data, and ensuring users can access only the pages and commands appropriate to their role.

6.2.3 Scrum Master

Kathryn Bridson and Dr. James Yu are serving as the co-scrummasters for the team. They have participated and spectated some Scrum meetings, as well as been available in answering questions involving potential roadblocks.

7 Development Process

7.1 Agile Scrum Process

Scrum is an Agile software development framework. It consists of completing work in “sprints” - which are 2-4 weeks. At the beginning of each sprint, items are assigned to be completed at the end of the sprint. By the end of the sprint, there should be a minimum valuable product (MVP) for the customer to see and work with. There are several roles in the Scrum: the product owner, scrum master, and the team. The product owner is responsible for the creation of user stories, which are specific issues or features the product should incorporate. The scrum master’s role is to help the team along in the Scrum process, as well as resolve any roadblocks. The team’s role is actually implement user stories and develop the product. Finally, there are several “ceremonies” in Scrum. These ceremonies include: SCRUM meeting (stand ups), planning, review, and retrospective. Standups are designed to take less than 15 minutes and for the team to inform each other on what they have done, what they are going to do, and if they have any roadblocks or impediments to their work. Planning is where the team and product owner decide and plan the work for the next sprint. Review is where the team reviews the sprint work and demos any work to the customer. Retrospective is where the team reviews and decides internally what went well in a sprint and what the team needs to improve on (process wise)

The team is following the Agile Scrum development process. The team conducts a stand up/ Scrum meeting twice a week, usually before class time on Mondays and Wednesdays. However, the team has also conducted Scrum meetings in-class when time was allotted for that purpose. Before the start of each Sprint, the team conducts a planning meeting, which is usually about an hour. Before planning, the team generally discusses what the team should do for the next Sprint, and the Product Owner revises or creates new stories as appropriate. The team reviews each story in the Product Backlog, makes any needed changes to the story or its acceptance criteria, performs “Planning Poker” to score each story appropriately, and discusses who will be work on each story. Following the end of the sprint and the Sprint Review, the team

conducts a retrospective meeting to discuss what went well in the sprint and what needs to be improved on in the next sprint. This sometimes leads into a brief discussion of refinement of stories, time permitting. This meeting is usually less than 30 minutes long.

7.2 Project Development and Management Platform

Atlassian JIRA is used as the Project Management platform, which is provided via a University of Memphis hosted server. JIRA is where the Product Backlog and Sprint Backlog are hosted for the team. User stories are recorded in JIRA, and during a sprint, stories are moved into columns such as To-Do, In-Progress, In Validation, and Done through a virtual Kanban Board. JIRA also provides functionality that allows the team to track their velocity and sprint progress (through burndown charts).

Additionally, the team uses GitHub to host the project's source code and provide source control. During development, developers work on their local copy of the source code and submit a pull request whenever they wish to push the changes to the master branch of the source code. This allows the team to make sure no breaking changes were introduced into the project before verification. In addition, a separate "Publish" branch is maintained and is pushed at the end of each sprint. Heroku, which is responsible for hosting the project, automatically pulls changes from the Publish branch and deploys them.

7.3 Communication Method

The team primarily communicates through Slack, which is a group messaging application. The team has multiple channels for front-end and back-end developers, as well as a general channel for keeping in touch with the entire team. Confluence is also used through the provided University hosted server. Meeting agendas and notes are posted on Confluence, when possible.

Standups are performed twice a week - before, during, or after class time, where permitted. Planning, retrospective, and review ceremonies are usually conducted on weekends, due to the limited availability of the team during weekdays.

8 Glossary

- Debater: user that submits responses to a particular side of a debate topic
- Spectator: user that only views and votes on arguments in the debate
- Moderator: user that views and moderates the debate and its arguments
- Admin: special user (member of project team) that manages debate topics and has greater privileges that debaters, spectators, and moderators
- Chat: feature and accompanying UI that allows user to submit and views arguments in a debate
- Argument: response to a debate topic
- Django: web framework written in Python that allows web applications to run on the web
- Python: portable and light programming language

- PostgreSQL: database system similar to MySQL but open source.
- MVC: Model View Controller; a pattern that allows an application to be broken down into three different components
- Model: manages data in the MVC pattern
- View: output representation of model in the MVC pattern
- Controller: accepts input in an application for use in model or view in the MVC pattern
- Heroku: cloud provider that supports Django projects
- Git: distributed source code and version control system
- GitHub: online source code repository provider
  Chrome: web browser created by Google
- Firefox: web browser created by Mozilla
- Bootstrap: CSS and JavaScript library that has many built in theming and controls
- Crispy Forms - forms library for Django
- Source: A URL which can be posted alongside an argument in order to support their statement.
- Jumbotron - A bootstrap component which allows text to be placed within a large box element. Used to separate argument posts in a nice and clean fashion.
- Vote - to be able to score/ vote a particular argument in the debate

9 Results

9.1 Requirements Gathering, Analysis and Design

9.1.1 Product Backlog

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	DEBATE-1	This is your first Issue (user story) for this project	Weihl, Matthew	James Yu		DONE	Done	24/Sep/17	02/Oct/17	
	DEBATE-2	Talk , discuss, argue, or debate which one is better	Frye, Brianna	James Yu		DONE	Done	24/Sep/17	26/Sep/17	
	DEBATE-3	As a developer, I want to publish the project to Heroku, so that I can access the website.	Weihl, Matthew	Weihl, Matthew		DONE	Done	02/Oct/17	19/Oct/17	
	DEBATE-4	As a user, I want to be a debater, so I can participate in the debate.	Frye, Brianna	Weihl, Matthew		DONE	Done	02/Oct/17	15/Nov/17	
	DEBATE-5	As a developer, I want to create a Heroku account, so I can host the website.	Weihl, Matthew	Weihl, Matthew		DONE	Done	02/Oct/17	19/Oct/17	
	DEBATE-6	As a user, I want to watch a debate as a spectator.	Nabors, Austin	Weihl, Matthew		DONE	Done	02/Oct/17	15/Nov/17	
	DEBATE-7	As a user, I want to participate as a debate moderator, so I can moderate the debate.	*Unassigned*	Weihl, Matthew		DONE	Done	02/Oct/17	15/Nov/17	
	DEBATE-8	As a developer, I want to go through a Python tutorial, so I can be more familiar with Python.	Tadesse, Semere	Weihl, Matthew		DONE	Done	09/Oct/17	27/Oct/17	
	DEBATE-9	As a product owner, I want to create a story map, so I can better plan future sprints.	Weihl, Matthew	Weihl, Matthew		DONE	Done	10/Oct/17	19/Oct/17	
	DEBATE-11	As a Developer, I want to be more accustomed with the Django Framework	Brown, Benjamin	Brown, Benjamin		DONE	Done	15/Oct/17	19/Oct/17	
	DEBATE-12	As a developer, I want to publish the project to Heroku, so that I can access the website.	Weihl, Matthew	Weihl, Matthew		DONE	Done	19/Oct/17	22/Oct/17	
	DEBATE-13	As a developer, I want to create a Heroku account, so I can host the website.	Weihl, Matthew	Weihl, Matthew		DONE	Done	19/Oct/17	22/Oct/17	
	DEBATE-14	As a product owner, I want to create a story map, so I can better plan future sprints.	Weihl, Matthew	Weihl, Matthew		DONE	Done	19/Oct/17	24/Oct/17	
	DEBATE-15	As a developer, I want to be more accustomed with the Django Framework.	Brown, Benjamin	Weihl, Matthew		DONE	Done	19/Oct/17	24/Oct/17	
	DEBATE-16	As a product owner, I want to schedule a planning meeting time, so we can conduct Sprint Planning.	*Unassigned*	Weihl, Matthew		DONE	Done	19/Oct/17	27/Oct/17	
	DEBATE-17	As a developer, I want to set up CI/ unit testing, so I can track whether the project is building.	Brown, Benjamin	Weihl, Matthew		DONE	Done	19/Oct/17	25/Oct/17	
	DEBATE-18	As a team member, I want to work on our problem statement, so I can finalize our problem statement.	Frye, Brianna	Weihl, Matthew		DONE	Done	19/Oct/17	24/Oct/17	
	DEBATE-19	As a developer, I want to set up the local development environment, so I can develop, compile, and run the project locally.	Brown, Benjamin	Weihl, Matthew		DONE	Done	19/Oct/17	24/Oct/17	
	DEBATE-20	As a debater, I can a choose a side, so I can argue in a particular side of a debate.	Brown, Benjamin	Weihl, Matthew		DONE	Done	30/Oct/17	14/Nov/17	
	DEBATE-23	As a developer, I will create a chat system, that allows posting and viewing of debate responses.	Frye, Brianna	Weihl, Matthew		DONE	Done	30/Oct/17	15/Nov/17	
	DEBATE-24	As an admin, I will manage the debate prompts, so users can engage in debates.	Frye, Brianna	Weihl, Matthew		DONE	Done	30/Oct/17	24/Nov/17	
	DEBATE-37	As a developer, I would like to create a GitHub organizational account, so other developers have more access to configuring GitHub repo.	Weihl, Matthew	Weihl, Matthew		DONE	Done	31/Oct/17	24/Nov/17	
	DEBATE-38	As a debater, I can add a citation in my response, so I can have better sourced responses.	Brown, Benjamin	Weihl, Matthew		DONE	Done	01/Nov/17	29/Nov/17	
	DEBATE-39	As a debater, I can create a team, so I can better organize in a debate.	*Unassigned*	Weihl, Matthew		TO DO	*Unresolved*	01/Nov/17	21/Nov/17	
	DEBATE-40	As a spectator, I can vote on debater responses, so I can provide better feedback for the debate.	*Unassigned*	Weihl, Matthew		TO DO	*Unresolved*	01/Nov/17	10/Dec/17	
	DEBATE-41	As a spectator, I can report debaters, so moderators can remove debaters.	Nabors, Austin	Weihl, Matthew		DONE	Done	01/Nov/17	27/Nov/17	
	DEBATE-42	As a moderator, I can ban users, so I can reduce abuse in a debate.	Frye, Brianna	Weihl, Matthew		DONE	Done	01/Nov/17	10/Dec/17	
	DEBATE-43	As a moderator, I can remove arguments and comments, so I can better control abuse/ on-topic comments in a debate.	Nabors, Austin	Weihl, Matthew		DONE	Done	01/Nov/17	10/Dec/17	
	DEBATE-44	As a moderator, I can request citations from debaters, so debates are better sourced.	*Unassigned*	Weihl, Matthew		TO DO	*Unresolved*	01/Nov/17	10/Dec/17	
	DEBATE-45	As a developer, I would like to investigate a debate rubric, so I can determine the best way to score debates.	Weihl, Matthew	Weihl, Matthew		DONE	Done	01/Nov/17	10/Dec/17	
	DEBATE-46	As a spectator, I can vote on a debate according to a debate rubric, so I can help pick the winner of a debate.	Frye, Brianna	Weihl, Matthew		DONE	Done	01/Nov/17	03/Dec/17	
	DEBATE-47	As a moderator, I can rate limit responses in a debate, so I can help control abuse in a debate.	*Unassigned*	Weihl, Matthew		TO DO	*Unresolved*	01/Nov/17	30/Nov/17	
	DEBATE-48	As a developer, I would like to implement a profanity filter, so profanity does not appear in a debate without moderator manually removing comments.	*Unassigned*	Weihl, Matthew		TO DO	*Unresolved*	01/Nov/17	10/Dec/17	
	DEBATE-52	As a developer, I will make changes to the general design/ CSS of argument cards, so users are able to better distinguish between arguments.	Brown, Benjamin	Weihl, Matthew		DONE	Done	21/Nov/17	29/Nov/17	
	DEBATE-53	As a user, I can add comments to an argument, so I can facilitate discussion about an argument.	Frye, Brianna	Weihl, Matthew		DONE	Done	21/Nov/17	10/Dec/17	
	DEBATE-54	As a developer, I would like to "clean up" the UI, so the product is easier to use (nicer to look at).	Brown, Benjamin	Weihl, Matthew		IN PROGRESS	*Unresolved*	30/Nov/17	05/Dec/17	
	DEBATE-55	As a moderator, I can ban users, so I can reduce abuse in a debate (Enhancements)	*Unassigned*	Weihl, Matthew		TO DO	*Unresolved*	30/Nov/17	30/Nov/17	
	DEBATE-56	As a user, I can read about the rules of the website, so I can better understand how to use the website.	Weihl, Matthew	Weihl, Matthew		DONE	Done	30/Nov/17	03/Dec/17	
	DEBATE-57	As a user,I can access a help page, so I can go for assistance in using the website.	Weihl, Matthew	Weihl, Matthew		DONE	Done	30/Nov/17	05/Dec/17	

The product backlog (PBL) contains a total of 35 stories. These stories were developed through the creation of a story map that laid out the “Happy Path” from Sprint 1 to Sprint 3, as well as various stories conducted in Sprint 0. Additionally, these stories were refined further to

include Acceptance Criteria, and in some cases were split up into separate stories. These stories are left unassigned until Sprint Planning, where they are then assigned or picked up.

9.1.2 Functional Requirements

The project should allows users to conduct a debate virtually. It should contain similar rules and roles to those present in a real-life debate. Users should be able to submit responses and new arguments to a debate topic. If users are not participating in adding new debate arguments, they should spectate the debate and be able to vote on which side is more persuasive and has the most well-structured arguments. Additionally, there should be a way for some types of users to be able to moderate the debate and to prevent abuse and off-topic discussion.

Sprint 1 (User Stories)

DEBATE-4	As a user, I want to be a debater, so I can participate in the debate.
DEBATE-6	As a user, I want to watch a debate as a spectator.
DEBATE-7	As a user, I want to participate as a debate moderator, so I can moderate the debate.
DEBATE-20	As a debater, I can a choose a side, so I can argue in a particular side of a debate.
DEBATE-23	As a developer, I will create a chat system, that allows posting and viewing of debate responses.
DEBATE-24	As an admin, I will manage the debate prompts, so users can engage in debates.

Sprint 2 (User Stories)

DEBATE-38	As a debater, I can add a citation in my response, so I can have better sourced responses.
DEBATE-41	As a spectator, I can report debaters, so moderators can remove debaters.
DEBATE-42	As a moderator, I can ban users, so I can reduce abuse in a debate.
DEBATE-43	As a moderator, I can remove arguments and comments, so I can better control abuse/ on-topic comments in a debate.
DEBATE-52	As a developer, I will make changes to the general design/ CSS of argument cards, so users are able to better distinguish between arguments.
DEBATE-53	As a user, I can add comments to an argument, so I can facilitate discussion about an argument.

Sprint 3 (User Stories)

DEBATE-58	As a user, I can view the history of previous debates, so I can see the viewpoints from older debates.
DEBATE-59	As a spectator, I can view each side's score, so I can see who is winning the debate.
DEBATE-46	As a spectator, I can vote on a debate according to a debate rubric, so I can help pick the winner of a debate.

Basic Functions of DebateMate

- As a user, I should be able to register a user account.
- As a user, I should be able to participate as a debater and submit new arguments.
- As a user, I should be able to participate as a spectator and vote on arguments in a debate.
- As a user, I should be able to moderate a debate as a moderator and help keep the debate on topic.
- As an admin, I should able to change the debate topic and ban users, if necessary.
- As a user, I am able to add a source to my argument.
- As a user, I am able to comment on other users' arguments.
- As a user, I am allowed to report other arguments for abuse or spam.
- As a user, I can vote on debate arguments.

Activity diagram

The following activity diagrams show the general flow of how users interact with DebateMate. The first diagram covers all actions that are available in any role and accessible from every page via the navigation bar. The subsequent diagrams cover all actions that are specific to a certain role.

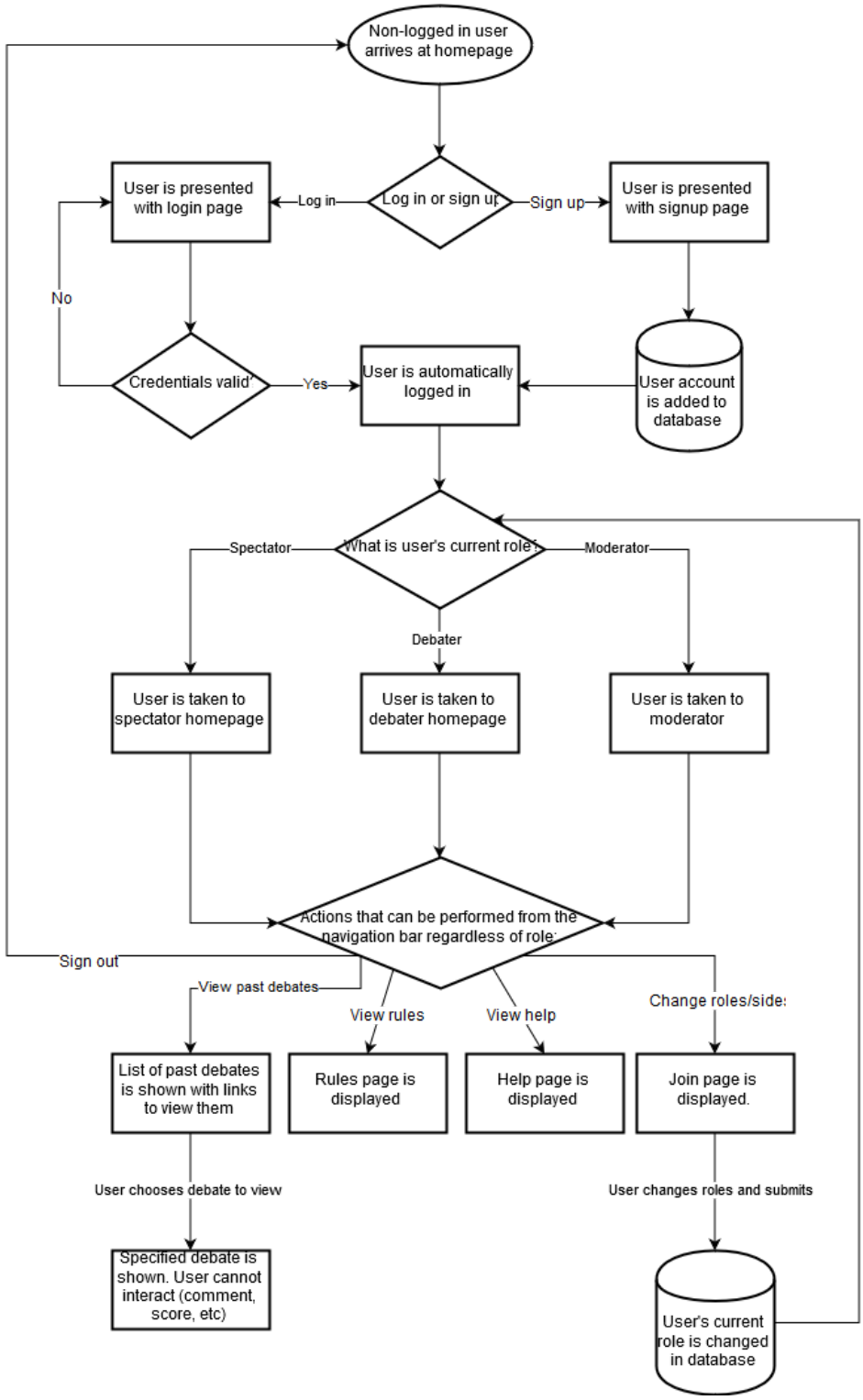


Figure 1 - General activities

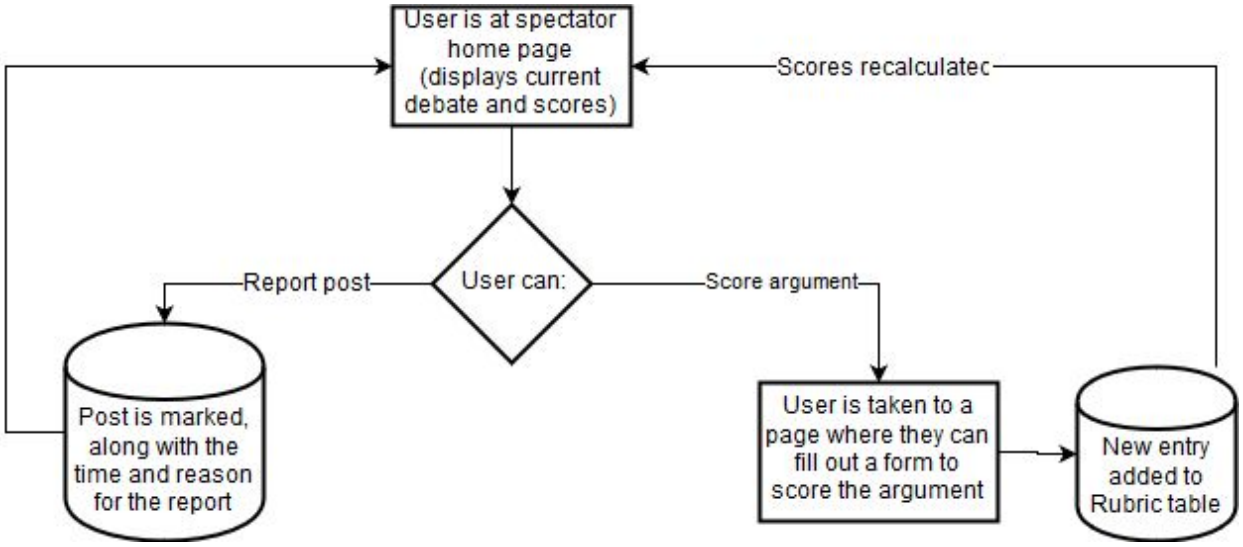


Figure 2 - Activities that spectators can perform from their homepage

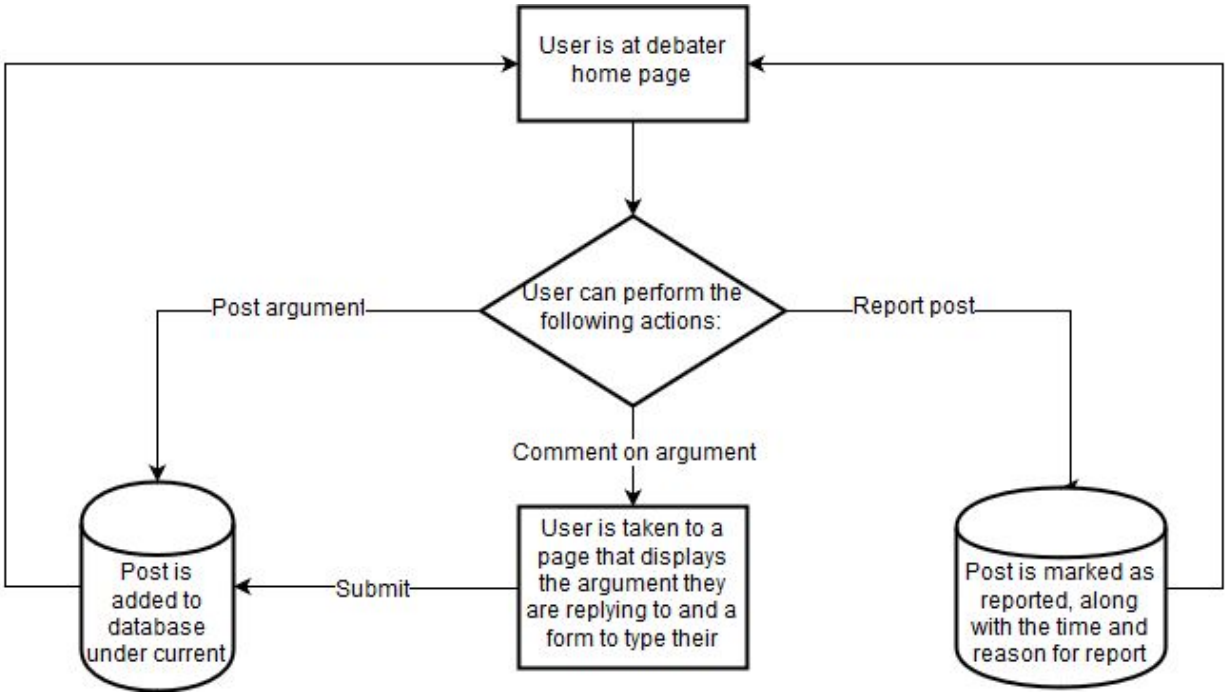


Figure 3 - Activities that debaters can perform from their homepage

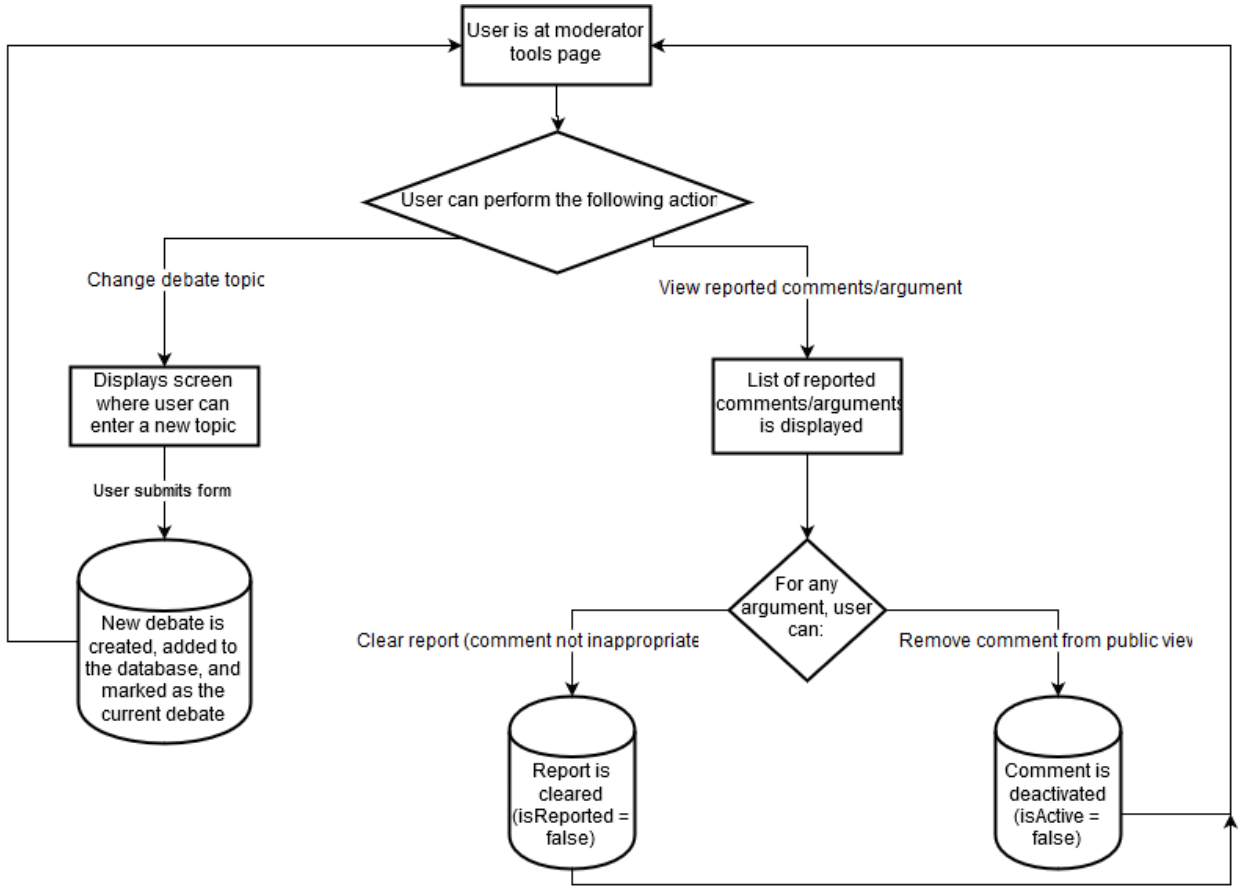


Figure 4 - Activities that moderators can perform from their homepage

9.1.3 Non-Functional Requirements

DebateMate will be developed in Python using the Django MVC library. Additionally, the back-end (i.e. the database) will be implemented in PostgreSQL. Finally, the project will be hosted in the Heroku Cloud. Numerous libraries are used in the project, such as Bootstrap and Crispy Forms.

Sprint 1 (User Stories)

DEBATE-37	As a developer, I would like to create a GitHub organizational account, so other developers have more access to configuring GitHub repo.

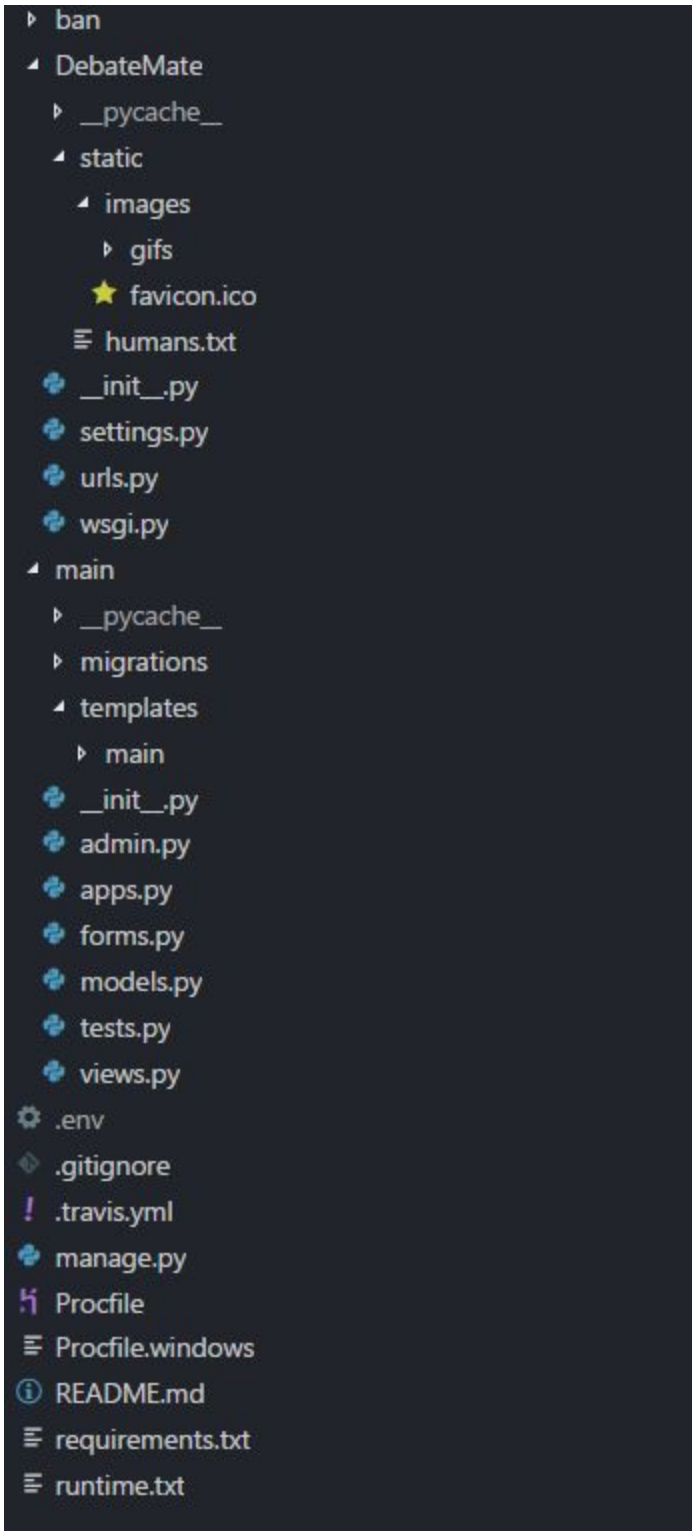
Sprint 2 (User Stories)

DEBATE-45	As a developer, I would like to investigate a debate rubric, so I can determine the best way to score debates.

Sprint 3 (User Stories)

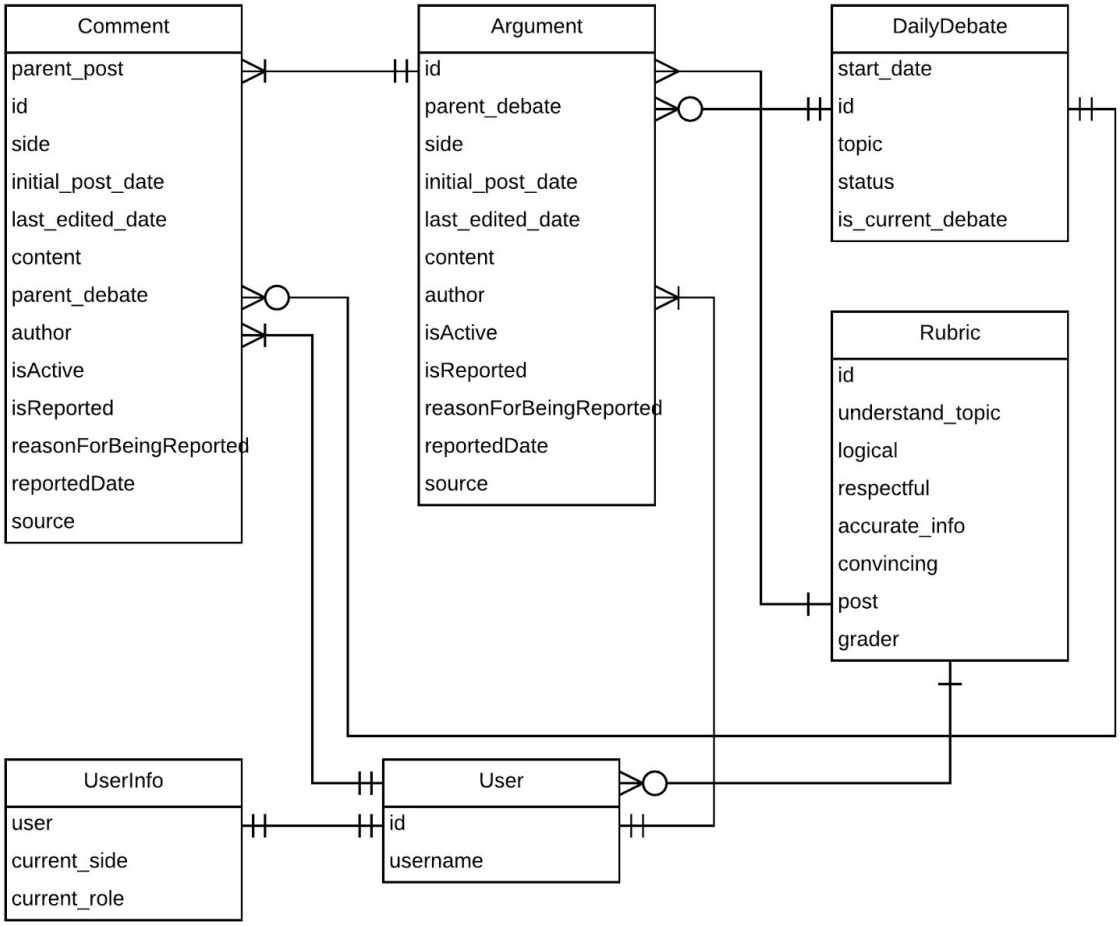
DEBATE-56	As a user, I can read about the rules of the website, so I can better understand how to use the website.
DEBATE-57	As a user,I can access a help page, so I can go for assistance in using the website.
DEBATE-54	As a developer, I would like to "clean up" the UI, so the product is easier to use (nicer to look at).

Source Code Organizational Tree



Entity-relationship diagram of database

The following diagram shows the relationships of the different tables in the database used by DebateMate.



Builds and Deployment

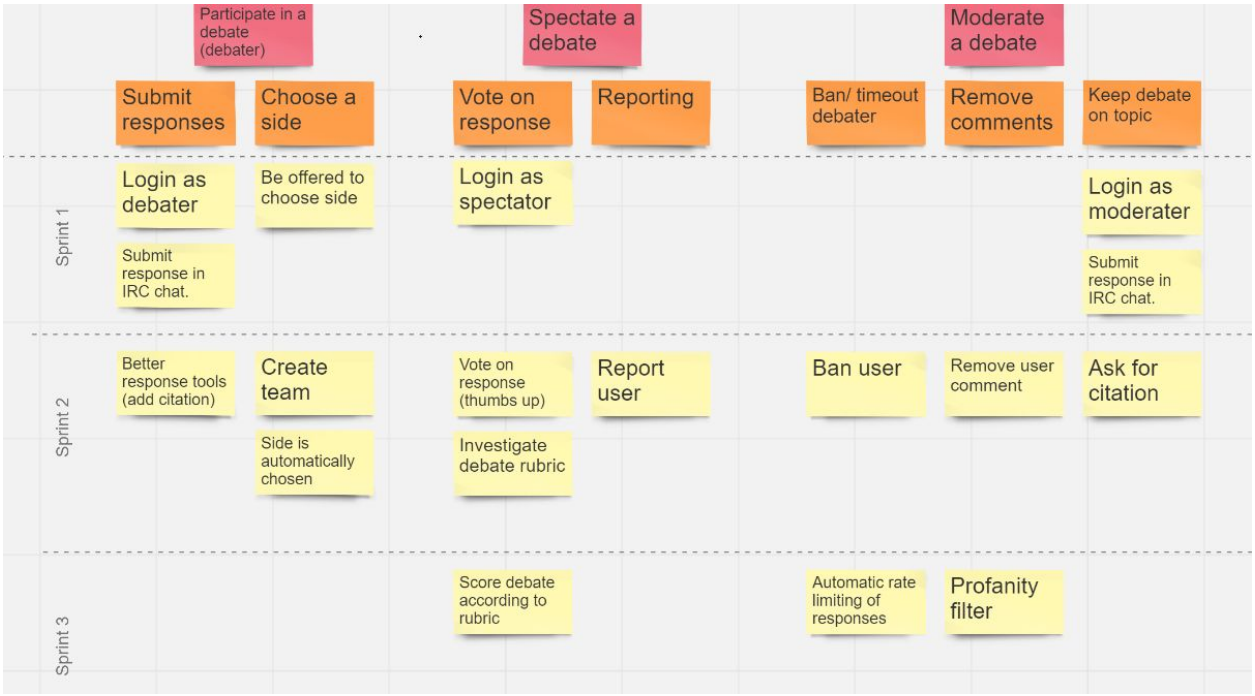
When the project is ready to be deployed after the end of a sprint, the master branch of the Git repository is merged into a separate Publish branch. Heroku, the host for the website, will then automatically push the changes from the Publish branch to the website. When the team is running the website locally, the command “python manage.py runserver 127.0.0.1:5000” is run, which starts the Django project locally and allows the team to access the website.

9.1.4 Other Specific Requirements

The database is implemented in PostgreSQL, which is hosted by Heroku. Since DebateMate is a website, it is supported by any device that has support for the web. However, DebateMate has been primarily been tested on Chrome and Firefox on the desktop.

9.2 Agile Development Planning Results

9.2.1 Sprint Planning Summary



9.2.2 Roadmap from Sprint 1 to Sprint 3

Sprint 1 Roadmap - Bare bones

The main focus for Sprint 1 included getting the “bare bones” of the project off the ground. This included: implementation of the different roles in a debate (debater, spectator, and moderator), a basic chat system that allows users to submit arguments or responses to a debate topic, and allowing admins to set a debate topic.

Sprint 1 User Stories

Key	Summary	Issue Type	Priority	Status	Story Points (26)
DEBATE-4	As a user, I want to be a debater, so I can participate in the debate.	Story	Medium	DONE	5
DEBATE-6	As a user, I want to watch a debate as a spectator.	Story	Medium	DONE	3
DEBATE-7	As a user, I want to participate as a debate moderator, so I can moderate the debate.	Story	Medium	DONE	5
DEBATE-20	As a debater, I can a choose a side, so I can argue in a particular side of a debate.	Story	Medium	DONE	3
DEBATE-23	As a developer, I will create a chat system, that allows posting and viewing of debate responses.	Story	Medium	DONE	5
DEBATE-24	As an admin, I will manage the debate prompts, so users can engage in debates.	Story	Medium	DONE	3
DEBATE-37	As a developer, I would like to create a GitHub organizational account, so other developers have more access to configuring GitHub repo.	Story	Medium	DONE	2

Sprint 2 Roadmap - Moderation and Debate tooling

Whereas Sprint 1's theme was centered around getting a basic implementation of the "chat"/ argument system up and running, Sprint 2's theme was centered around improving the moderation and debater tools.

Sprint 2 User Stories

Key	Summary	Story Points (24)
DEBATE-38	As a debater, I can add a citation in my response, so I can have better sourced responses.	3
DEBATE-41	As a spectator, I can report debaters, so moderators can remove debaters.	3
DEBATE-42	As a moderator, I can ban users, so I can reduce abuse in a debate.	5
DEBATE-43	As a moderator, I can remove arguments and comments, so I can better control abuse/ on-topic comments in a debate.	5
DEBATE-45	As a developer, I would like to investigate a debate rubric, so I can determine the best way to score debates.	2
DEBATE-52	As a developer, I will make changes to the general design/ CSS of argument cards, so users are able to better distinguish between arguments.	3
DEBATE-53	As a user, I can add comments to an argument, so I can facilitate discussion about an argument.	3

Sprint 3 User Stories

Key	Summary	Story Points (15)
DEBATE-46	As a spectator, I can vote on a debate according to a debate rubric, so I can help pick the winner of a debate.	5
DEBATE-56	As a user, I can read about the rules of the website, so I can better understand how to use the website.	2
DEBATE-57	As a user,I can access a help page, so I can go for assistance in using the website.	2
DEBATE-58	As a user, I can view the history of previous debates, so I can see the viewpoints from older debates.	3
DEBATE-59	As a spectator, I can view each side's score, so I can see who is winning the debate.	3

Issues Not Completed

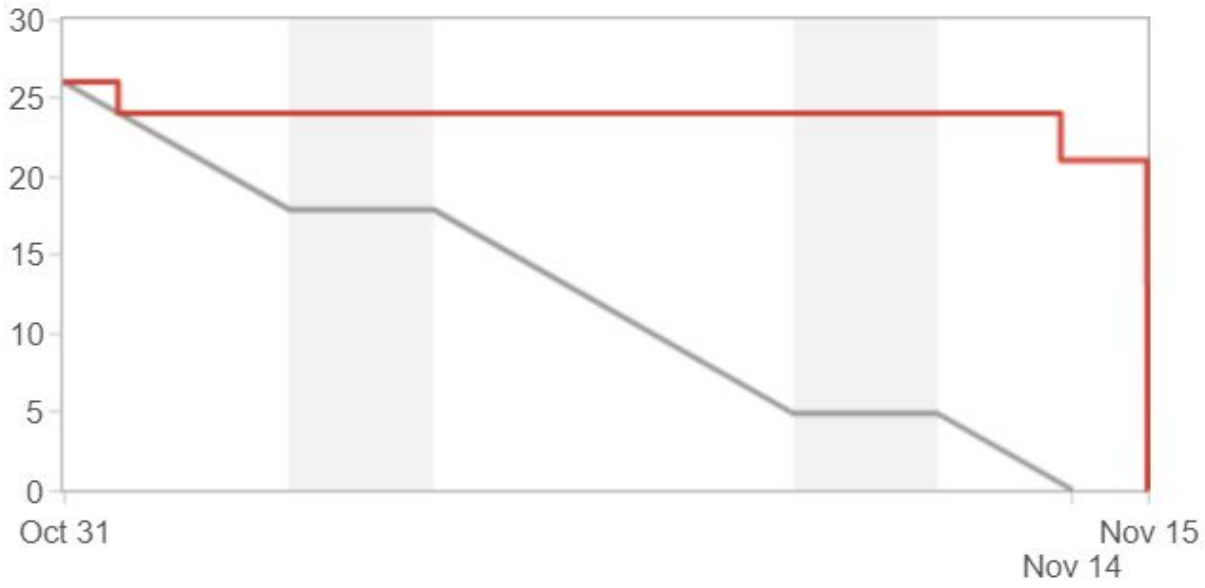
Key	Summary	Story Points (3)
DEBATE-54	As a developer, I would like to "clean up" the UI, so the product is easier to use (nicer to look at).	3

Sprint 3 Roadmap - Voting and Scoring

Sprint 3's theme was around implementing the voting and scoring system for debates. The sprint also included some fit and finish work to smoothen some rough edges of the project.

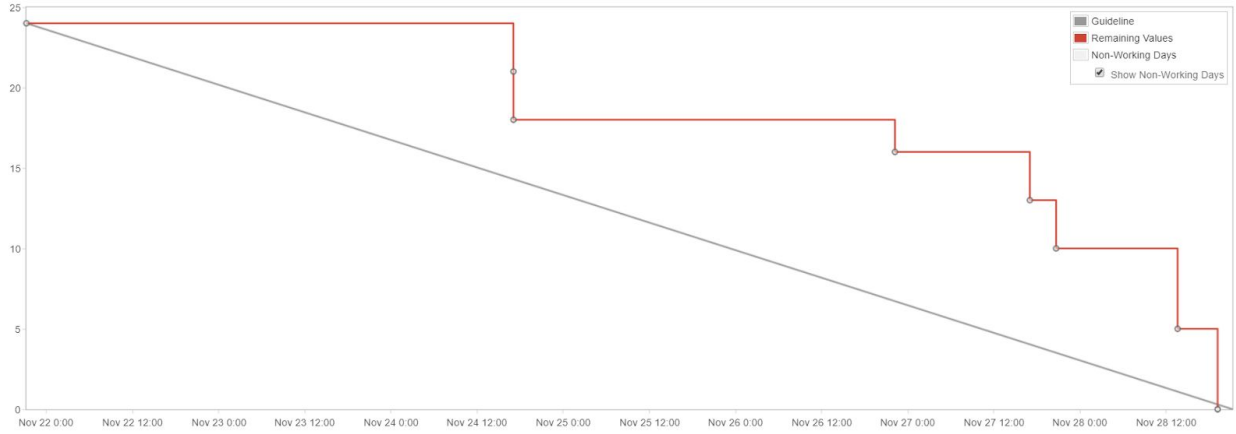
9.3 Sprint Burndown Chart Analysis

9.3.1 Sprint 1 Burndown Chart



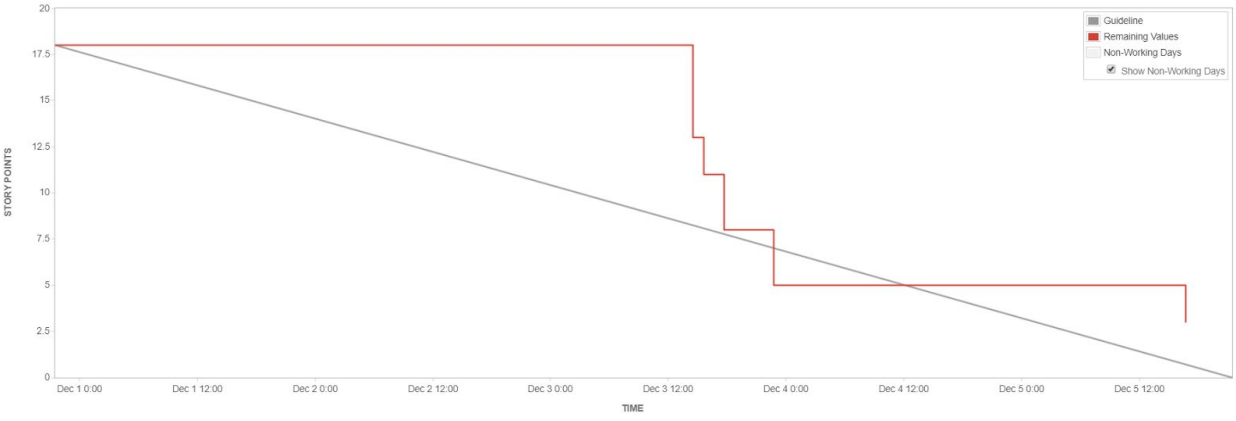
During Sprint 1, the team discovered during the sprint that many of the user stories depended on one another - e.g. a story about the implementation of the debater role relied on the story about implementing the chat system. Therefore, many stories were stuck in “In Progress” while the chat system was being developed. The chat system was not successfully implemented until the end of the sprint, and then the other stories were able to be closed. Therefore, the burndown chart shows the line staying at the top until the very end, where it suddenly drops. The team also additionally did not do a good job on estimating stories, and as a result, many stories turned out to be more complicated than originally planned.

9.3.2 Sprint 2 Burndown Chart



Sprint 2 went much more smoothly versus Sprint 1, as shown in the above burndown chart. The line in the chart gradually goes down as work as completed during the sprint. However, the chart shows progress was not completely smooth. For example, look at the very beginning of the chart. There is a multiple day period where the line stays straight. This is a visualization of Thanksgiving Break, and as a result some team members were unable to complete work as quickly as they would have if Thanksgiving Break did not occur. Many of the stories in Sprint 2 were more independent, which allowed the team to complete them and not necessarily having to keep a story in progress while waiting on another feature to completed or “done done”.

9.3.1 Sprint 3 Burndown Chart

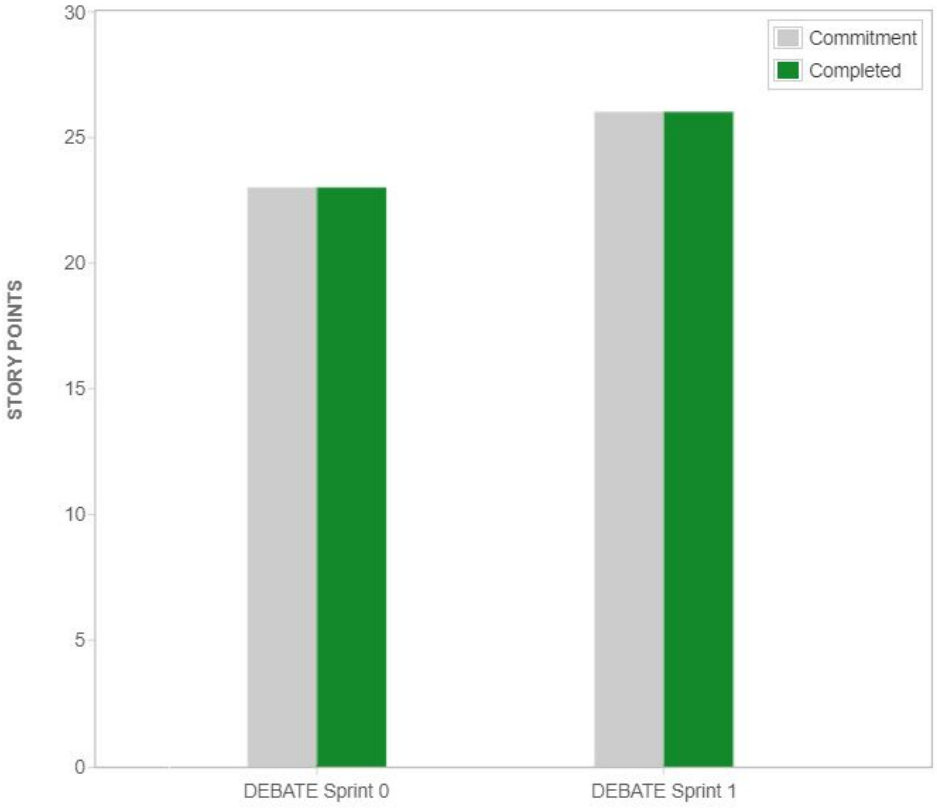


Sprint 3 was shorter versus Sprint 2 (5 days versus 7 days, respectively). And some team members were busy preparing for tests in other classes, and therefore were unable to work on the project until the weekend. This is the cause of the straight line at the beginning of the chart. On Sunday December 8 and Monday December 4, the team was able to finish multiple stories on one day, which caused the chart to jump down quickly. The chart never goes

down to 0 in terms of story points, since one story (DEBATE-54 - UI changes) was not completed during the sprint.

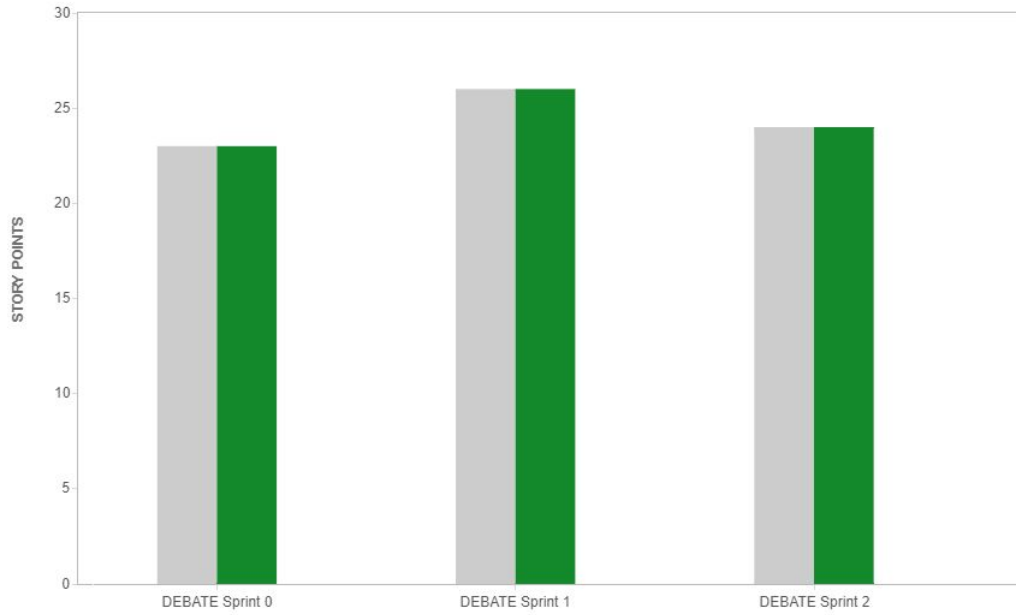
9.4 Sprint Velocity Chart Analysis

9.4.1 Sprint 1 Velocity Chart



The sprint velocity for Sprint 1 was 26 points. Although the team was able to complete everything, the team had to go through “crunch mode” and scramble to finish all committed stories in the sprint. Some stories in the sprint took more time than originally thought during planning, which caused the team to scramble. So while the chart may seem the team was able to complete everything on time, the chart is misleading, since it does not show the scramble and death march towards the end of the sprint.

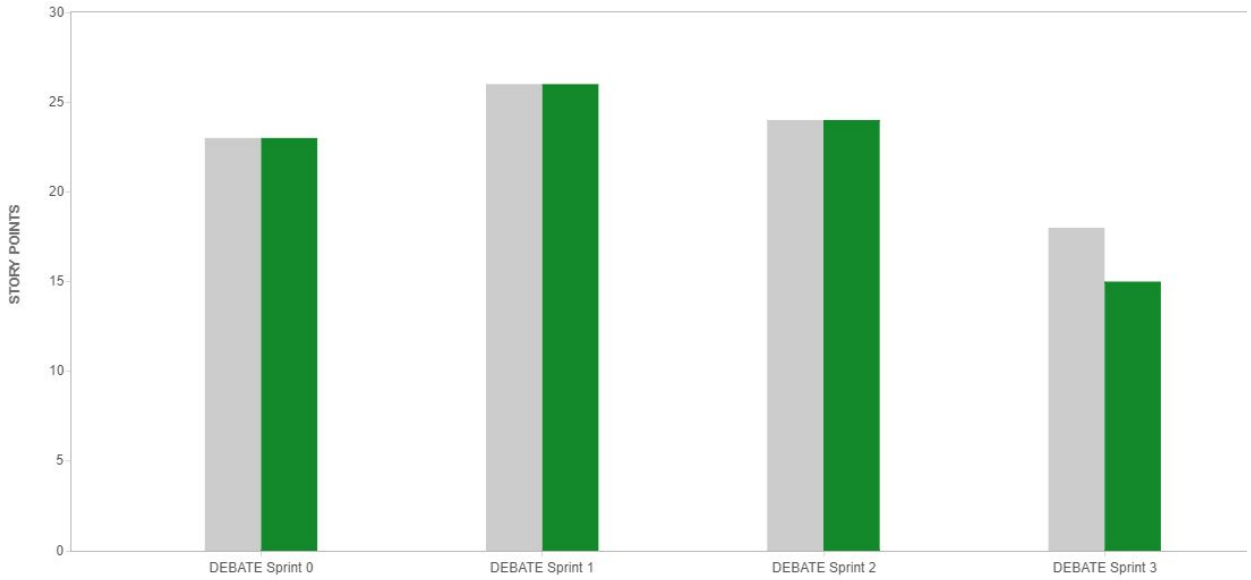
9.4.2 Sprint 2 Velocity Chart



Sprint	Commitment	Completed
DEBATE Sprint 0	23	23
DEBATE Sprint 1	26	26
DEBATE Sprint 2	24	24

The sprint velocity for Sprint 2 was 24 points. If you notice, even though the sprint was 1 week shorter versus Sprint 1, the team was able to complete a similar amount of work, with some exceptions. The chart is a bit misleading again, since some acceptance criteria was modified mid-sprint, due to issues in completing a story. This is not visible on the velocity chart, since the story is still marked as completed.

9.4.3 Sprint 3 Velocity Chart



Sprint	Commitment	Completed
DEBATE Sprint 0	8	8
DEBATE Sprint 0	23	23
DEBATE Sprint 1	26	26
DEBATE Sprint 2	24	24
DEBATE Sprint 3	18	15

For Sprint 3, the sprint velocity was a few points lower versus Sprint 2, due to the shorter sprint length (5 days). While there was a two day difference between Sprint 2 and Sprint 3, this decrease in length combined with some team members being busy with other classes, this contributed to the decrease in the velocity. Additionally, our team was unable to complete one story (DEBATE-54) in the sprint.

9.5 Risk Tracking Tables

9.5.1 Sprint 1 Risk Tracking Table

Possible Risk	**Severity of Risk**	**Likelihood of Risk**	**Steps Taken to Avoid Risk**
Lack of time to complete sprint	High	Medium	Reducing scope in Sprint 1 of features (“barebones”)
Inexperience in development frameworks	Medium	Medium	Reading through Django and Python documentation

9.5.2 Sprint 2 Risk Tracking Table

Possible Risk	Severity of Risk	Likelihood of Risk	Steps Taken to Avoid Risk
Thanksgiving Break may impede progress	Medium	High	Availability during break discussed during planning and accommodations made, if necessary
Smaller team size may impede progress	Medium	High	Reducing size of Sprint

9.5.3 Sprint 3 Risk Tracking Table

Possible Risk	Severity of Risk	Likelihood of Risk	Steps Taken to Avoid Risk
Work in other classes	Medium	High	Work completed on weekends
Shorter length	Medium	High	Less stories committed

9.6 Retrospectives

9.6.1 Sprint 1

For Sprint 1, the team learned many lessons. One lesson learned was the need for better written, more detailed user stories. As stated previously, some user stories in Sprint 1 were dependent on one another, which caused many user stories to stay in progress while waiting on other user stories. The team underestimated the difficulty of many stories in the sprint, which caused stories to be marked as done at the last minute since they were not able to be completed until the end of the sprint. Additionally, the team ran into some mid-sprint design changes (e.g. the chat system no longer being realtime).

In future sprints, the team will focus on developing stories in greater detail - by including more acceptance criteria and discussing some technical details during planning. Additionally, sub-tasks will not be used, since they can negatively affect the burndown chart.

9.6.2 Sprint 2

For Sprint 2, we learned some lessons. The team learned that by only including acceptance criteria and using it as the basis for our definition of done, the team was able to be more flexible in terms of the implementation of stories. The team learned that modifying acceptance criteria allows stories to be completed if issues occur, however these changes negatively affect the velocity chart since they are not visible on the chart. During the review/ demo, it was commented that the demo/ project does appear to have a good flow (moving from one task to another), which was a side effect of the stories worked on being enhancements to the core product, rather than must-have features.

In future sprints, the team will focus on reducing acceptance criteria changes, as well improving demo flows.

9.6.3 Sprint 3

For Sprint 3, the team learned a few lessons. The team learned that even just a two day difference in sprint length (5 vs 7 days) can affect the team's velocity. The team also learned that features that seem easy to implement during planning can transform to problems that are harder to solve and result in not being to complete a user story. The team encountered this when trying to implement the side by side argument acceptance criteria in DEBATE-54 (UI changes). During planning, the team initially thought that since Bootstrap (the UI framework DebateMate users) supports grids and columns, the UI could easily be adapted to show each side's arguments side by side. However, this turned out not to be the case. The team also had difficulty about thinking how the UI would like in a mobile view (DebateMate is designed to be as "responsive" as possible).

9.7 Scrum Meeting Notes

9.7.1 Sprint 1 Sample

DebateMate - Scrum Meeting - 11/13/2017

Matthew Weihl:

1. What have I done? - Worked on team report.
2. What will I do? - Will continue to work on team report, start building out/ planning Sprint 2.
3. Roadblocks? - None.

Austin Nabors:

1. What have I done? - Worked on templates, front end.
2. What will I do? - Continue to work on templates, front-end.
3. Roadblocks? - None.

Benjamin Brown:

1. What have I done? - Worked on Join page
2. What will I do? - Continue working on Join page
3. Roadblocks? - None

Brianna Frye:

1. What have I done? - Worked on models, chat
2. What will I do? - Will continue working on chat.
3. Roadblocks? - None.

Semere Tadesse:
Not present.

9.7.2 Sprint 2 Sample

11/27 - Sprint 2 Stand-up/ SCRUM meeting
Template:

1. What have I done?
2. What will I do?
3. Roadblocks?

Team Members Present: Ben, Brianna, Austin, Matthew

Ben:

1. What have I done?: Source, formatting of /debate
2. What will I do?: Nothing
3. Roadblocks?: None

Brianna:

1. What have I done?: Working on comments - parent id. Some initial banning work done
2. What will I do?: Trying to implement bans through moderator view
3. Roadblocks?: None

Austin:

1. What have I done?: Spectator - report arguments.
2. What will I do?: Reporting comments
3. Roadblocks?: None

Matthew:

1. What have I done?: Rubric
2. What will I do?: Team report
3. Roadblocks?: None

9.7.3 Sprint 3 Sample

Standup meeting - 12/4
Team Members Present: Ben, Austin, Brianna, Matthew

Ben:

1. What have I done?: Nothing, busy with Models of Computation
2. What will I do?: will finish CSS - will be done tomorrow
3. Roadblocks: None

Austin:

1. What have I done?: Old Debate page working
2. What will I do?: test as many pages possible
3. Roadblocks: None

Brianna:

1. What have I done?: DB models scoring, spectators can score posts, debate pages show average score, each post shows score, altered MakeContext
2. What will I do?: Breakdown of criteria
3. Roadblocks: None

Matthew:

1. What have I done?: Rules
2. What will I do?: Help page
3. Roadblocks: None

10 Conclusions

10.1 Sprint 1

In conclusion, the first sprint was moderately successful. The team ran into numerous issues regarding the planning and creation of stories, which negatively affected the team's velocity and the sprint burndown chart. The team also ran into some unexpected issues related to inexperience with the project's development frameworks. Overall, this lead to the team barely finishing the sprint in time for the review and demo. However, the team believes they have a foundation where additional improvements can be made.

10.2 Sprint 2

In conclusion, Sprint 2 was a smoother experience for the team versus Sprint 1. We learned many lessons along the way, such as demo and product flows as well as when to change acceptance criteria. Thanksgiving Break did impact the productivity of the team, however the team was able to complete everything planned, with some minor changes to acceptance criteria that occurred because of some design changes in the implementation of certain stories (e.g. ban interface is not as intuitive as where the team would want it to be).

10.2 Sprint 3

In conclusion, the team encountered some challenges and difficulties in completing stories. This included encountering technical issues with Bootstrap, as well as general issues related to the shorter sprint length. However,by the end of Sprint 3, the team was able to complete all functionality necessary to meet the solution originally stated in the project's problem statement.

11 References

Bootstrap library documentation - http://bootstrapdocs.com/v3.0.3/docs/
Django Python documentation - https://docs.djangoproject.com/en/2.0/
Heroku Cloud provider - https://www.heroku.com/
Source for Debate voting rubric - https://www.niu.edu/facdev/_pdf/guide/strategies/classroom_debate_rubric.pdf