M.A.C.K.
Memory Leak
Project Plan

Andrew Palmer, Christian Young, Kody Kimberl, and Mike Bauer

Capstone Project: Spring 2022

The University of West Florida

4 February 2022

CIS4595: Capstone Project

Dr. Bernd Owsnicki Klewe

Abstract

This work outlines the fundamental parts of our plan that we will follow throughout the development of our project. Memory Leak will provide a much needed service to students of UWF's Department of Computer Science that aims to solve many of the issues with traditional tutoring on campus. This service will be delivered in the form of a web application, on which questions can be posed by students, and answered by students. In a sense, it is community-driven. The nature of this service, and our commitment to maximizing usability, security, and business logic, requires many essential use cases, possibly acting as an obstacle, among other things, like disagreements among the group. Our group is composed of various leads, and each one of us recognizes the responsibilities these roles bring. By analyzing the group mechanics, scheduling, and initial requirements of the project, the plan will ensure the organization and success of the Memory Leak project.

Table of Contents

1 Team and Project Information

Our team, M.A.C.K., is composed of Andrew Palmer, Christian Young, Kody Kimberl, and Mike Bauer. We will be working on a project entitled “Memory Leak”.

2 Executive Summary

Our project will provide the UWF Computer Science Department student community with a more organized, accessible, and helpful virtual platform for posing and answering questions related to course content. The only virtual platform native to UWF that allows students to ask questions is the Computer Science Discord server. However, from the perspective of both students and tutors, there are three main problems. The first is that students are hesitant to ask questions in such a public format, especially when their instructors or peers may be able to see what they asked. The second is that students who do not have some degree of status within the department (i.e. tutors, student assistants, officers, SGA members, or students with more seniority) either do not feel comfortable answering questions or feel that they do not have permission. The last is that some questions go entirely unanswered if someone posts a question immediately after, no one answers it quickly, or someone’s attempted answer does not work. Our project will attempt to solve all of these problems by making a more anonymous and open community platform that allows students of any status to ask and answer questions. The underlying concept of a ticketing system will help to ensure that all questions get answered and that it is easy for anyone to find questions that have been answered previously. Our platform will also allow for community moderation in order to ensure that answers are accurate and follow certain guidelines (such as being kind, appropriate, and free of potential academic misconduct issues).

3 Expected Project Requirements

3.1 User Stories and Use Cases

The following is a list of the user stories we developed for Memory Leak. Due to the size of the Use Case diagrams, it does not make sense to include them in this section. However, they will be included within Appendix A.

1. As a moderator, I would like to remove an answer that does not meet community guidelines.
2. As a moderator, I would like to remove a question that does not meet the community guidelines.
3. As a user, I would like to log in to my account.
4. As a user, I would like to edit my profile information.
5. As a user, I would like to view my profile information.
6. As a user, I would like to flag a question or answer that I think does not meet community guidelines.
7. As a moderator, I would like to view a list of flagged questions and answers.
8. As a user, I would like to ask a question.
9. As a user, I would like to be able to view a list of currently unanswered questions.
10. As a user, I would like to search through questions.
11. As a user, I would like to be able to comment on a question or answer.

12. As a user, I would like to view a selected question.
13. As a user, I would like to be able to answer current unanswered questions.
14. As a user, I would like to be able to mark questions I asked as answered.
15. As a user, I would like to be able to navigate the site.
16. As a user, I would like to be able to upvote or downvote.
17. As a user, I would like to be able to create an account.

3.2 System Components

The main component of our system will be a central server that serves each of the pages along with a database that will store limited user and question information. The application will be accessible to the client over the web. The server will be running Node.js and Angular and will be hosted on Firebase, with our data being stored in Firestore. With respect to protocols, HTTP will be the primary communication protocol considering the application is web-based. The aforementioned components are able to be linked within the production code by implementing credentials. Any necessary media content, such as images, will be stored on the server.

4 Anticipated Development/Deployment Process Model

We will be utilizing the Agile development methodology, more specifically the SCRUM process. This model was chosen because each member of the team is familiar with it already from Software Engineering. It also works very well for a web development setting that contains many modular tasks. The overall characteristic of agile that solidified our decision is that agile prioritizes responding to changes and working software over extensive plans and documentation. While planning and documentation are important, we wanted to ensure that the ability for change and growth is present as we all continue to develop the idea together.

5 Preliminary Roles

There are two sets of roles that we have identified. The first set of roles are those necessary for SCRUM development: Product Owner, SCRUM Master, and Development team. In addition, we have also designated leads for each important area of development.

5.1 SCRUM Roles

The SCRUM Master role will be taken on by Mike Bauer. Since he is our Software Engineering Lead, which will be further explained below, many of the responsibilities overlap. In general, he will be responsible for planning and reviewing sprints. He will also ensure that agile methodologies are followed by all members of the development team.

The Product Owner role will be taken on by Kody Kimberl. While there is no defined customer, it is important for the needs of users to be considered during each stage of the process. Since Kody is also the Testing lead, he can ensure that user's needs are met during the acceptance testing process.

All members of the team will be part of the development team. Every individual will be given various tasks to work on during the sprint planning phase. Each lead will generally focus on tasks in their respective areas, but there may be some overlap that causes leads to oversee others' tasks.

5.2 Lead Roles

Andrew Palmer will be our Security Lead. The responsibilities of this position include setting up security specifications, ensuring the integration of security into the development process, monitoring the coding process, implementing security tools to test the project, and performing and documenting security tests and evaluations.

Christian Young will be our Coding Lead. The responsibilities of this position include ensuring the quality of the codebase, integrating the code with the existing testing specifications, tracking commits to the central repository, and resolving any merge conflicts.

Kody Kimberl will be our Testing Lead and Project Lead. The responsibilities of the Testing Lead will include setting the testing specifications, monitoring the testability of the codebase, evaluating tests, and documenting the testing process. Additionally, the Project Lead is responsible for acting as the point of contact for the instructor, ensuring consistent workflow, completion of deliverables, and performance of responsibilities and resolving conflicts that may arise.

Mike Bauer will be our Software Engineering Lead. The responsibilities of this position include scheduling work, monitoring the development process, and selecting and provisioning development tools. Additionally, the position will involve ensuring that Agile methodologies are implemented and that the Sprint process is effectively planned and followed.

6 Anticipated Evaluation Strategies and Tools

First and foremost, our project intends on being as usable as possible. In order to evaluate this aspect, we will implement Aria in order to assess accessibility for disabled or handicapped users. In addition, we will ensure that the application undergoes numerous acceptance tests and quality assurance tests to continually assess the general user experience of the application.

The project will prioritize security by ensuring that login credentials are properly handled, user input is cleaned, and any publicly accessible user data does not breach confidentiality. In order to accomplish this aspect, the testing lead will be monitoring code development closely. Additionally, they will periodically conduct red team assessments of the application in order to test the exploitability of the application.

The project will also be evaluated by correctness of business logic by ensuring that all code is tested and working as intended. Tests will be written for JavaScript and TypeScript code using Jest. Tests involving larger user stories will be written in Gherkin style. In order to achieve this task, the testing lead will be closely monitoring the development process to ensure that it is test driven. Additionally, they will be conducting periodic tests of various scales to ensure that all aspects of the application are working as intended.

7 Preliminary Project Timeline and Tracking

Week	Dates	Project Goal	Course Milestones and Deliverables
1	01/10/22 - 01/16/22	Group Formation and Project Planning	
2	01/17/22 - 01/23/22	Further Planning	Submission of Group Members and Name
3	01/24/22 - 01/30/22	Project Idea Completion User Story Development	Project Idea

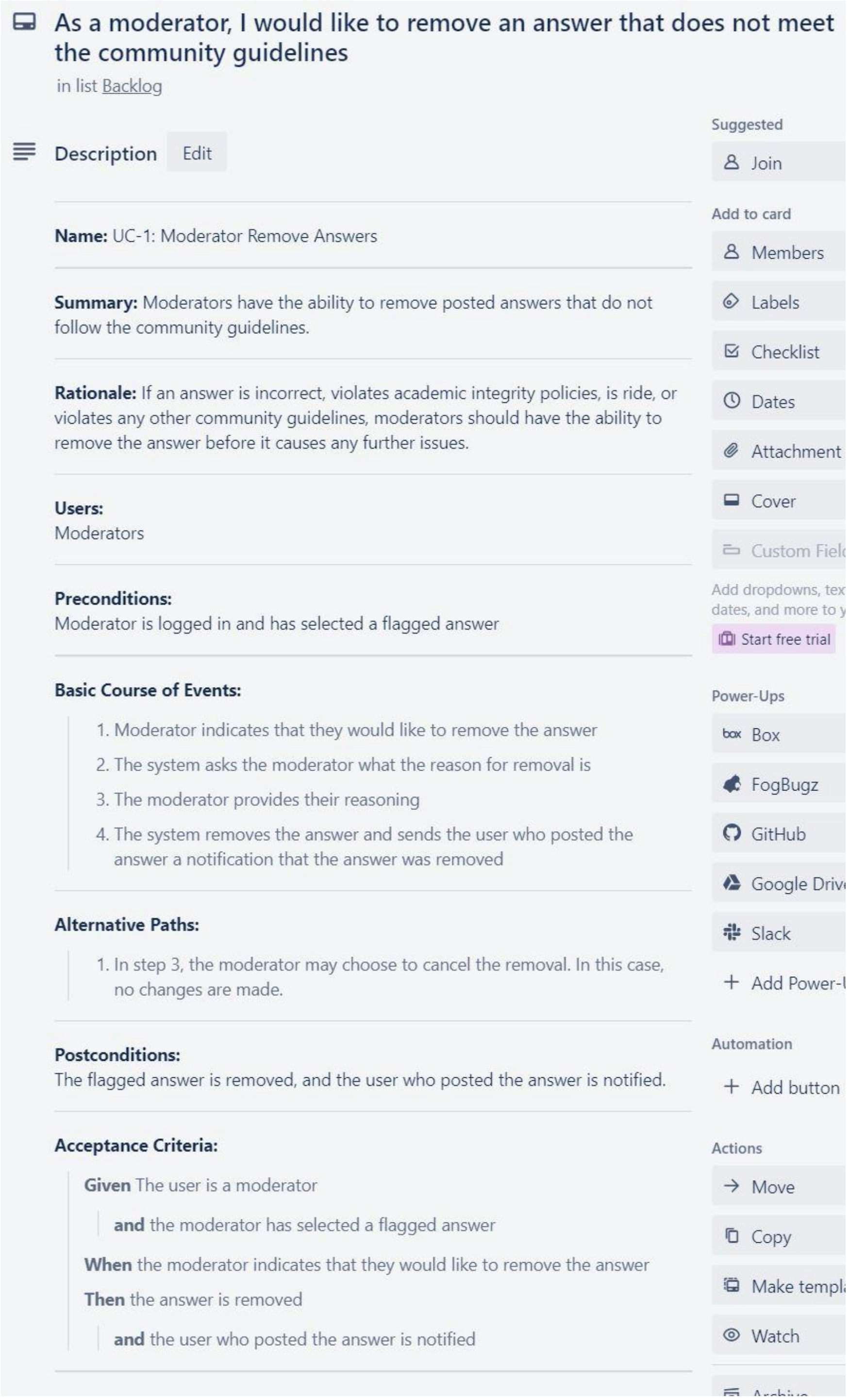
4	01/31/22 - 02/05/22	User Story Completion Planning Poker Sprint 1 Planning Initial Development Meeting Project Plan Completion Group Contract Completion Sprint 1 Begins	Project Plan Group Contract
5	02/06/22 - 02/12/22	Completion of Presentation 1	Presentation 1
6	02/13/22 - 02/19/22	Group Meeting 1 Sprint 1 Ends	Individual Plan Group Meeting
7	02/20/22 - 02/26/22	Sprint 2 Planning Sprint 2 Begins Technical Document Completion	Technical Document
8	02/27/22 - 03/05/22	Work on Presentation 2 Work on Group Report 1	
9	03/06/22 - 03/12/22	Completion of Presentation 2 Completion of Group Report 2 Sprint 2 Ends	Group Report 1 Presentation 2
10	03/13/22 - 03/19/22	Spring Break	
11	03/20/22 - 03/26/22	Sprint 3 Planning Sprint 3 Begins Group Meeting 2 Completion of Group Report 2	Group Report 2 Group Meeting 2
12	03/27/22 - 04/02/22	Completion of Group Report 3	Group Report 3
13	04/03/22 - 04/09/22	Sprint 3 Ends Completion of Group Report 4	Group Report 4
14	04/10/22 - 04/16/22	Sprint 4 Begins	Individual Presentation

15	04/17/22 - 04/23/22		Individual Report
16	04/24/22 - 04/30/22	Sprint 4 Ends Completion of Final Presentation	Final Presentation and Deliverables
17	05/01/22 - 05/07/22	Finals Week	Department Questionnaire

8 Project Risk/Success

The project outcomes as detailed in the use cases outlined in Section 3.2 can be defined as successful once all of their respective acceptance criteria are met. We anticipate that we will encounter obstacles that may impact project success. One such obstacle could be running out of time and being unable to complete all use cases, to which we would respond by prioritizing the remaining use cases by their assigned priority. Another potential obstacle could be merge conflicts, in which case the Coding Lead would be responsible for resolving merge conflicts. Additionally, disagreements regarding design implementation could impact project success. Our plan to address this would be to meet as a group and come to a consensus on how to implement design features or functions, with the Project lead resolving any conflicts and contacting the instructor if necessary.

Appendix A: Use Case Diagrams



As a moderator, I would like to remove an answer that does not meet the community guidelines

in list Backlog

Description

Name: UC-1: Moderator Remove Answers

Summary: Moderators have the ability to remove posted answers that do not follow the community guidelines.

Rationale: If an answer is incorrect, violates academic integrity policies, is ride, or violates any other community guidelines, moderators should have the ability to remove the answer before it causes any further issues.

Users:
Moderators

Preconditions:
Moderator is logged in and has selected a flagged answer

Basic Course of Events:

1. Moderator indicates that they would like to remove the answer
2. The system asks the moderator what the reason for removal is
3. The moderator provides their reasoning
4. The system removes the answer and sends the user who posted the answer a notification that the answer was removed

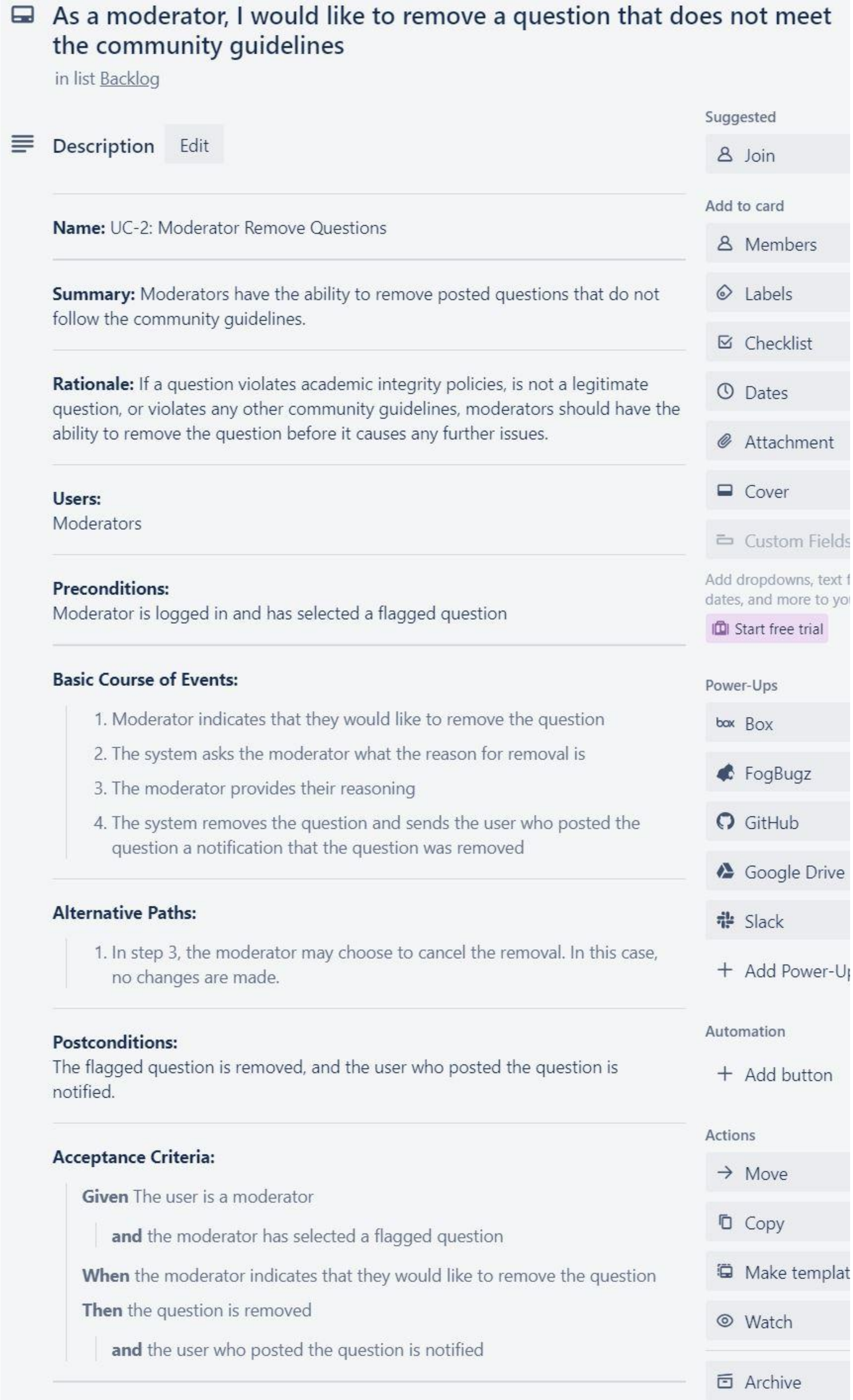
Alternative Paths:

1. In step 3, the moderator may choose to cancel the removal. In this case, no changes are made.

Postconditions:
The flagged answer is removed, and the user who posted the answer is notified.

Acceptance Criteria:

> **Given** The user is a moderator
>
> > **and** the moderator has selected a flagged answer
>
> **When** the moderator indicates that they would like to remove the answer
>
> **Then** the answer is removed
>
> > **and** the user who posted the answer is notified



As a moderator, I would like to remove a question that does not meet the community guidelines

in list Backlog

Description

Name: UC-2: Moderator Remove Questions

Summary: Moderators have the ability to remove posted questions that do not follow the community guidelines.

Rationale: If a question violates academic integrity policies, is not a legitimate question, or violates any other community guidelines, moderators should have the ability to remove the question before it causes any further issues.

Users:
Moderators

Preconditions:
Moderator is logged in and has selected a flagged question

Basic Course of Events:

> 1. Moderator indicates that they would like to remove the question
> 2. The system asks the moderator what the reason for removal is
> 3. The moderator provides their reasoning
> 4. The system removes the question and sends the user who posted the question a notification that the question was removed

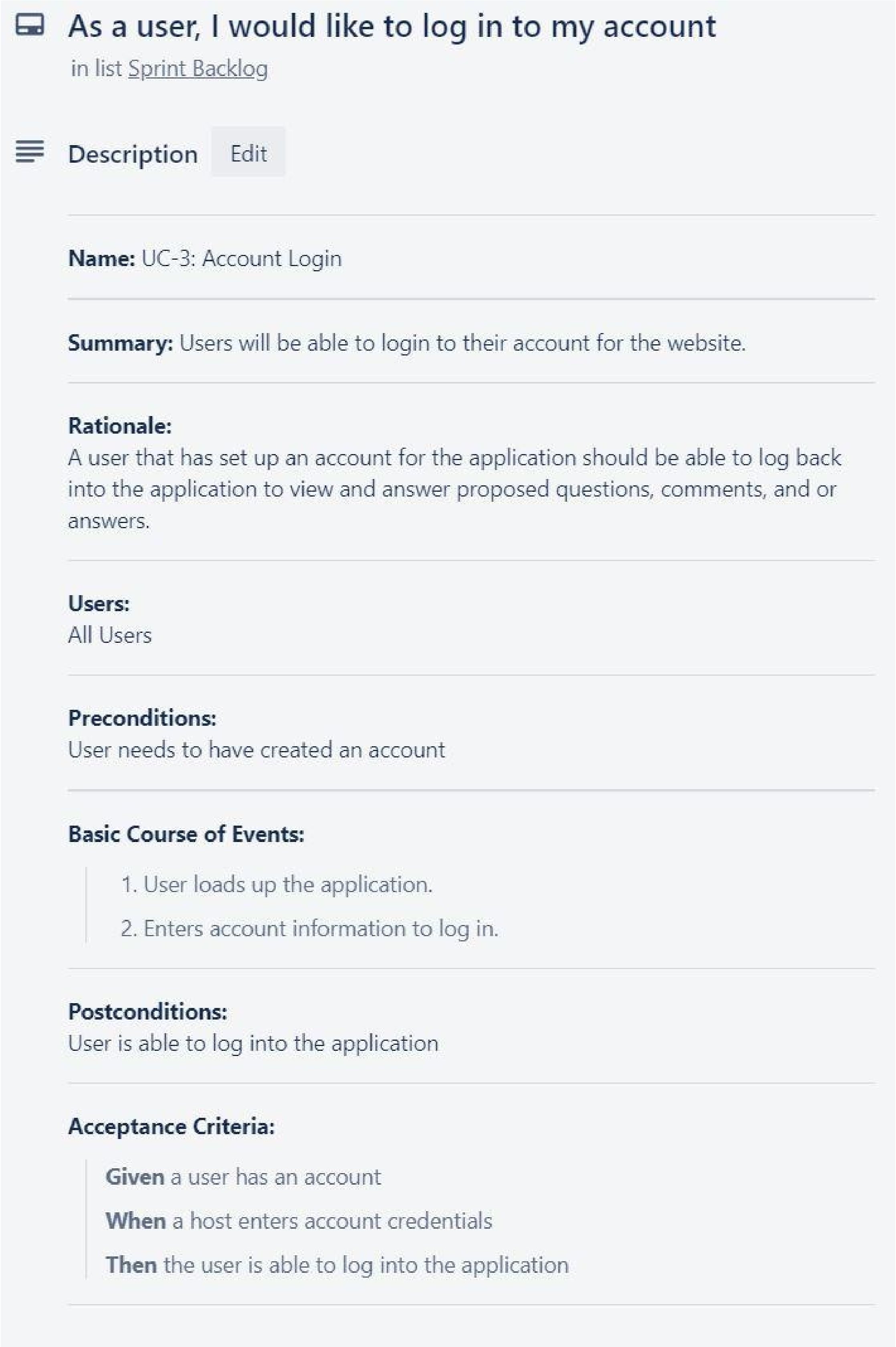
Alternative Paths:

> 1. In step 3, the moderator may choose to cancel the removal. In this case, no changes are made.

Postconditions:
The flagged question is removed, and the user who posted the question is notified.

Acceptance Criteria:

> **Given** The user is a moderator
>
> > **and** the moderator has selected a flagged question
>
> **When** the moderator indicates that they would like to remove the question
>
> **Then** the question is removed
>
> > **and** the user who posted the question is notified



As a user, I would like to log in to my account

in list Sprint Backlog

Description

Name: UC-3: Account Login

Summary: Users will be able to login to their account for the website.

Rationale:
A user that has set up an account for the application should be able to log back into the application to view and answer proposed questions, comments, and or answers.

Users:
All Users

Preconditions:
User needs to have created an account

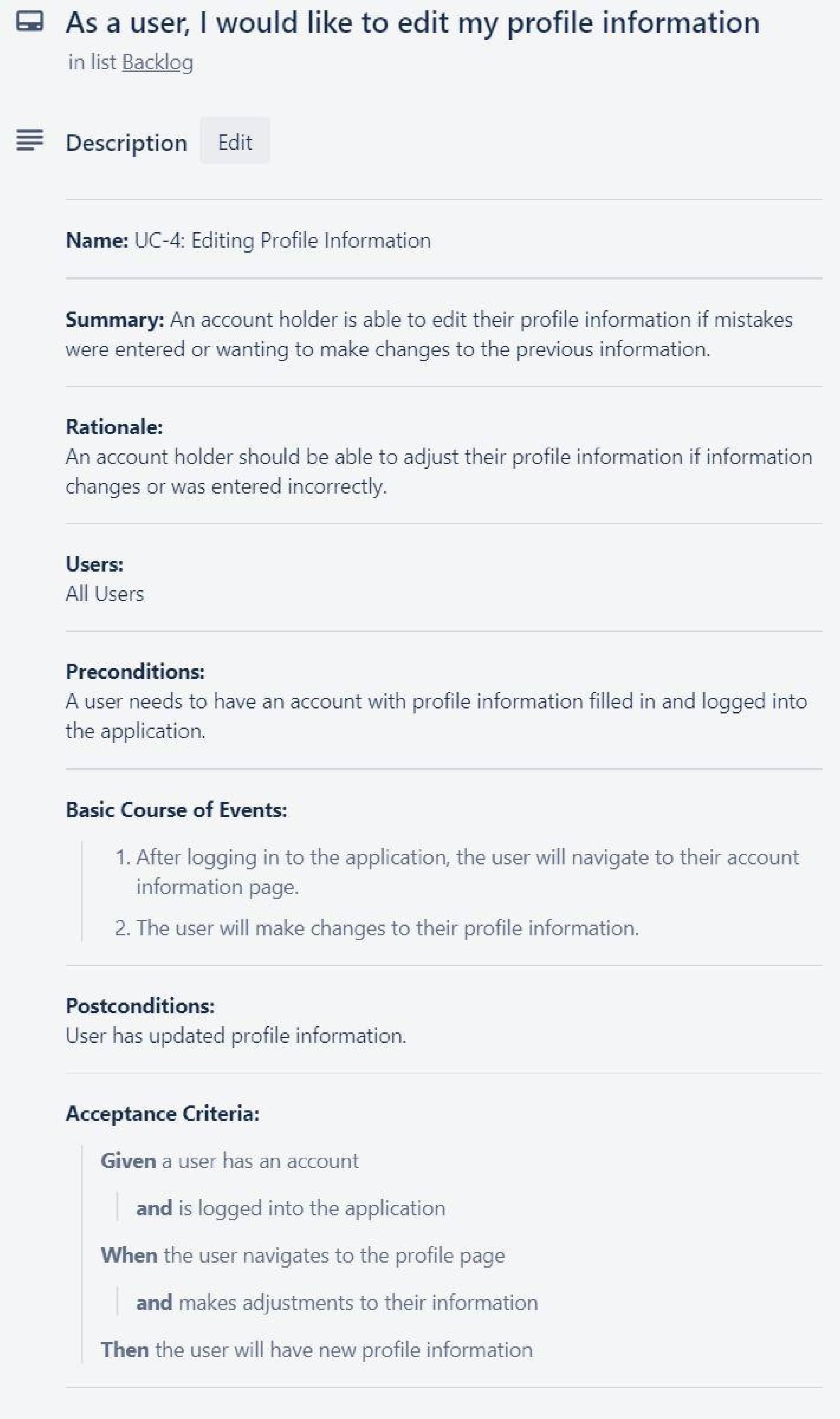
Basic Course of Events:

> 1. User loads up the application.
> 2. Enters account information to log in.

Postconditions:
User is able to log into the application

Acceptance Criteria:

> **Given** a user has an account
>
> **When** a host enters account credentials
>
> **Then** the user is able to log into the application



As a user, I would like to edit my profile information

in list Backlog

Name: UC-4: Editing Profile Information

Summary: An account holder is able to edit their profile information if mistakes were entered or wanting to make changes to the previous information.

Rationale:
An account holder should be able to adjust their profile information if information changes or was entered incorrectly.

Users:
All Users

Preconditions:
A user needs to have an account with profile information filled in and logged into the application.

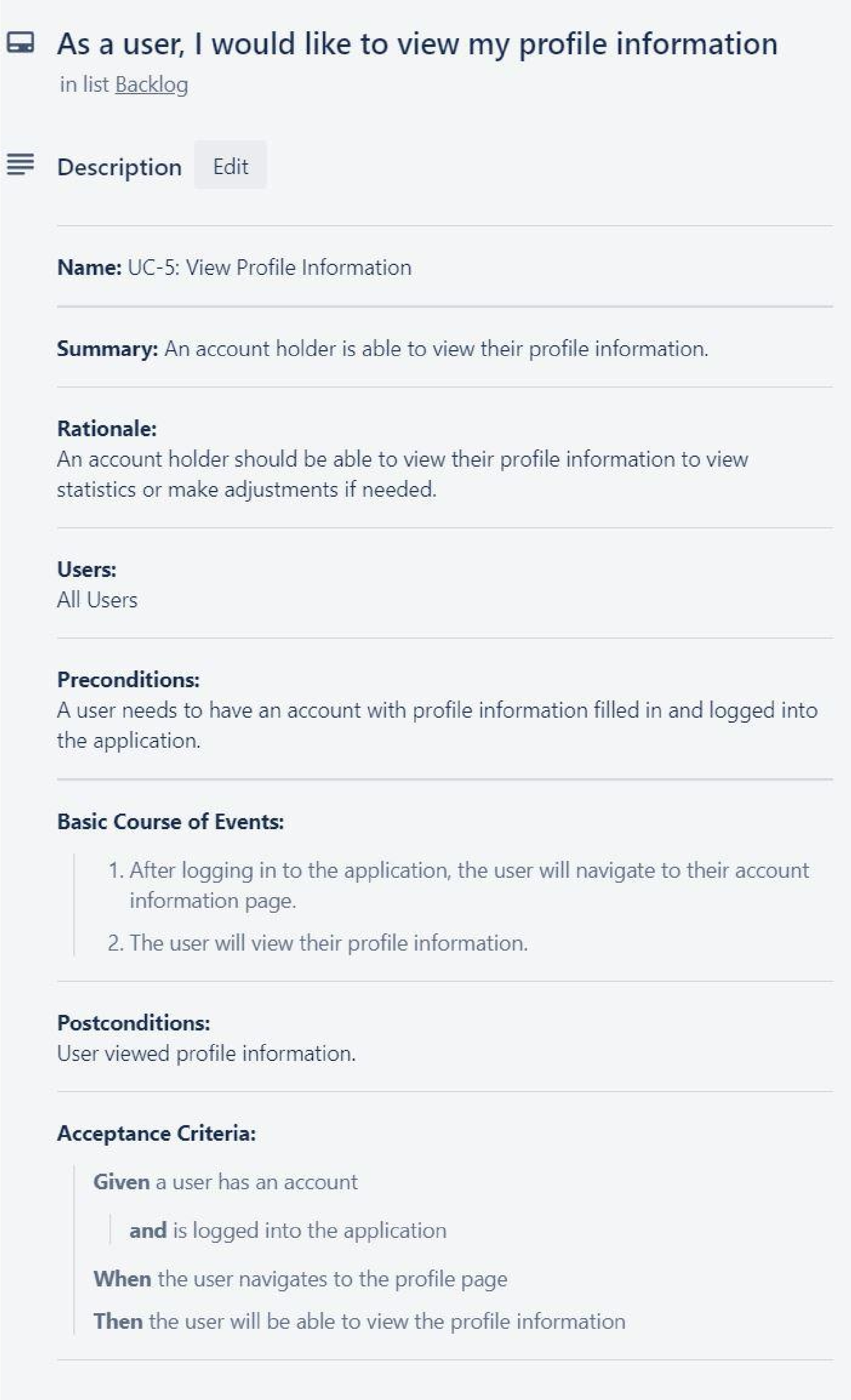
Basic Course of Events:

> 1. After logging in to the application, the user will navigate to their account information page.
> 2. The user will make changes to their profile information.

Postconditions:
User has updated profile information.

Acceptance Criteria:

> **Given** a user has an account
>
> > **and** is logged into the application
>
> **When** the user navigates to the profile page
>
> > **and** makes adjustments to their information
>
> **Then** the user will have new profile information



As a user, I would like to view my profile information

in list Backlog

Description

Name: UC-5: View Profile Information

Summary: An account holder is able to view their profile information.

Rationale:
An account holder should be able to view their profile information to view statistics or make adjustments if needed.

Users:
All Users

Preconditions:
A user needs to have an account with profile information filled in and logged into the application.

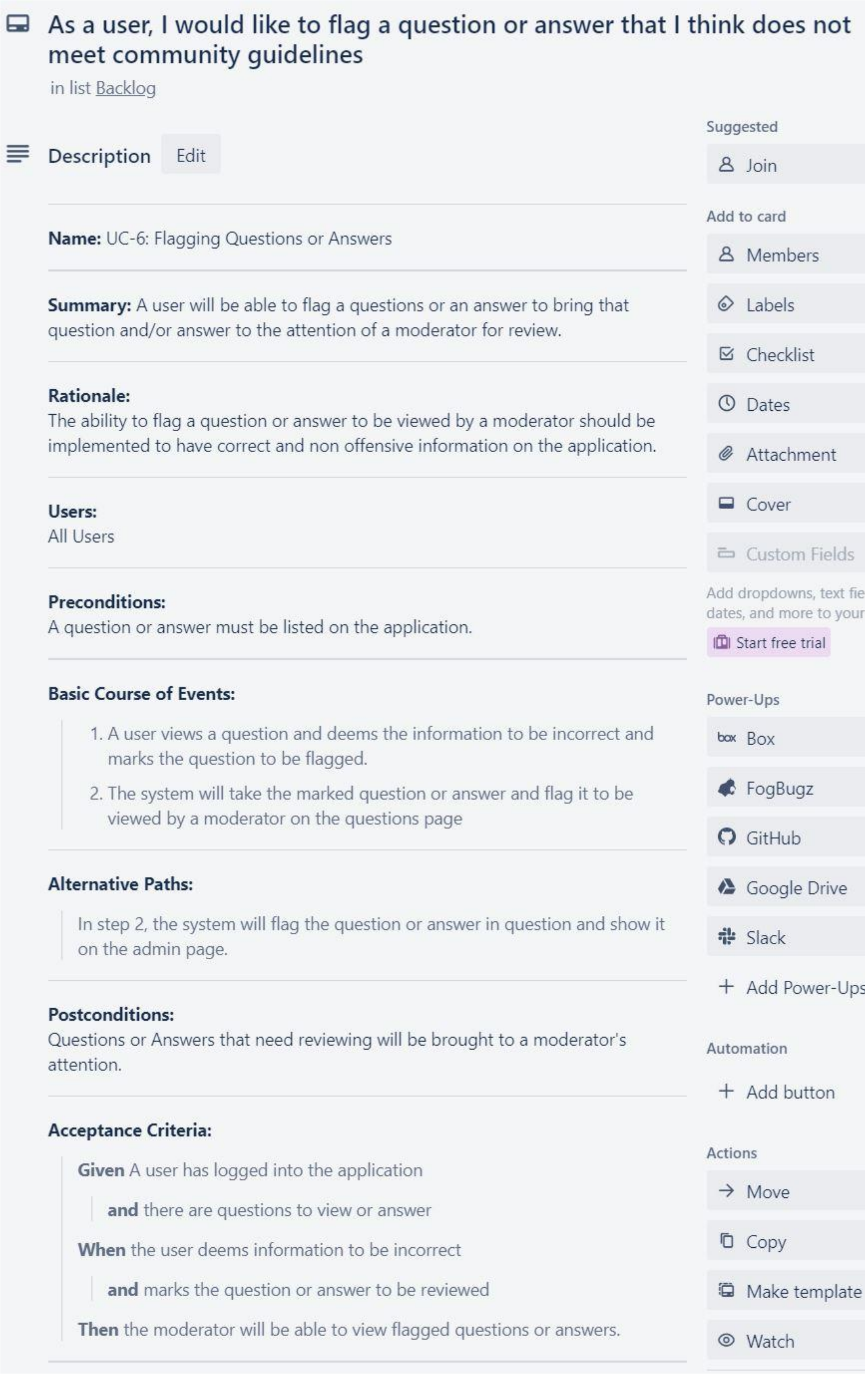
Basic Course of Events:

> 1. After logging in to the application, the user will navigate to their account information page.
> 2. The user will view their profile information.

Postconditions:
User viewed profile information.

Acceptance Criteria:

> **Given** a user has an account
>
> > **and** is logged into the application
>
> **When** the user navigates to the profile page
>
> **Then** the user will be able to view the profile information



As a user, I would like to flag a question or answer that I think does not meet community guidelines

in list Backlog

Description Edit

Name: UC-6: Flagging Questions or Answers

Summary: A user will be able to flag a questions or an answer to bring that question and/or answer to the attention of a moderator for review.

Rationale:
The ability to flag a question or answer to be viewed by a moderator should be implemented to have correct and non offensive information on the application.

Users:
All Users

Preconditions:
A question or answer must be listed on the application.

Basic Course of Events:

> 1. A user views a question and deems the information to be incorrect and marks the question to be flagged.
> 2. The system will take the marked question or answer and flag it to be viewed by a moderator on the questions page

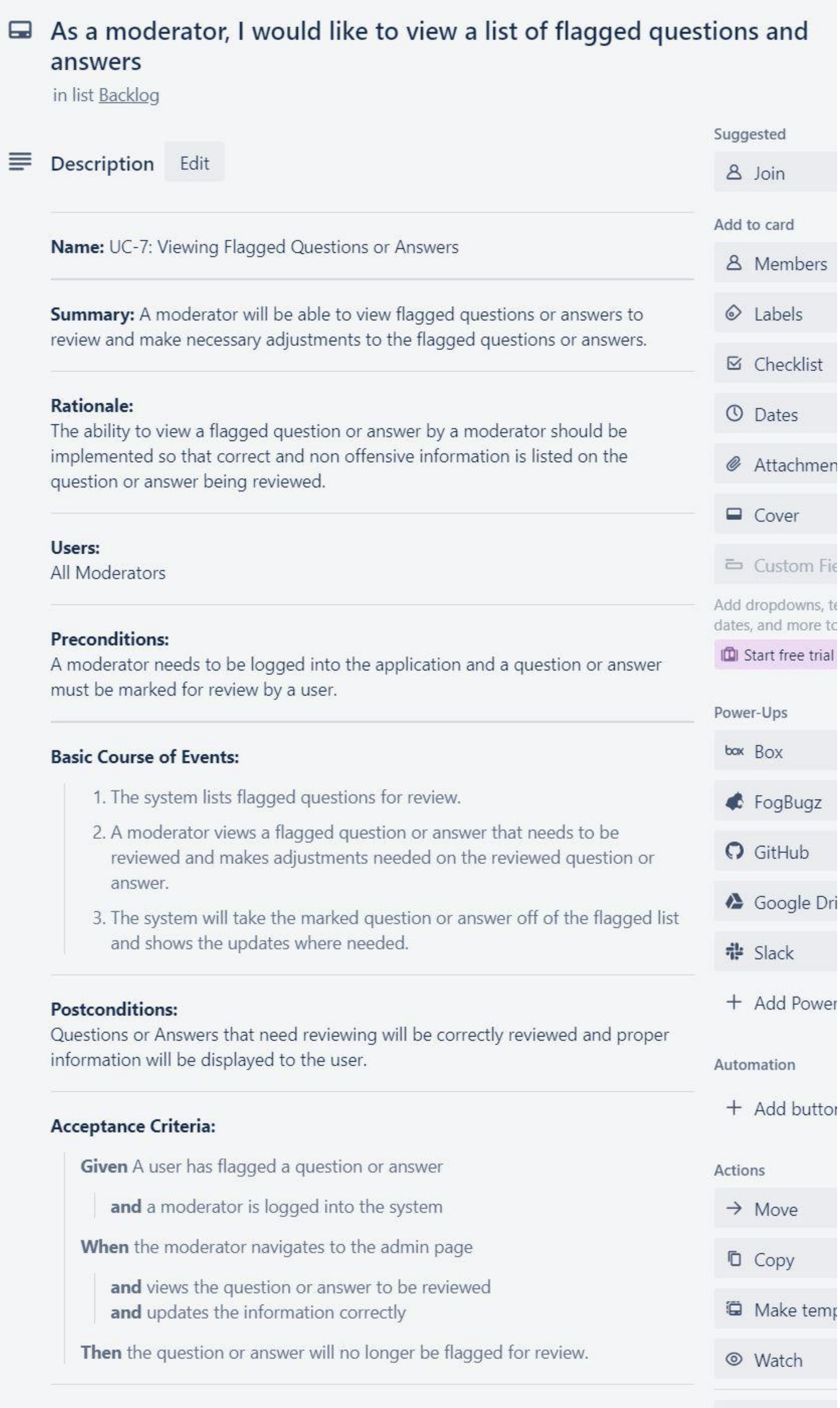
Alternative Paths:

> In step 2, the system will flag the question or answer in question and show it on the admin page.

Postconditions:
Questions or Answers that need reviewing will be brought to a moderator's attention.

Acceptance Criteria:

> **Given** A user has logged into the application
>
> > **and** there are questions to view or answer
>
> **When** the user deems information to be incorrect
>
> > **and** marks the question or answer to be reviewed
>
> **Then** the moderator will be able to view flagged questions or answers.



As a moderator, I would like to view a list of flagged questions and answers

in list Backlog

Description Edit

Name: UC-7: Viewing Flagged Questions or Answers

Summary: A moderator will be able to view flagged questions or answers to review and make necessary adjustments to the flagged questions or answers.

Rationale:
The ability to view a flagged question or answer by a moderator should be implemented so that correct and non offensive information is listed on the question or answer being reviewed.

Users:
All Moderators

Preconditions:
A moderator needs to be logged into the application and a question or answer must be marked for review by a user.

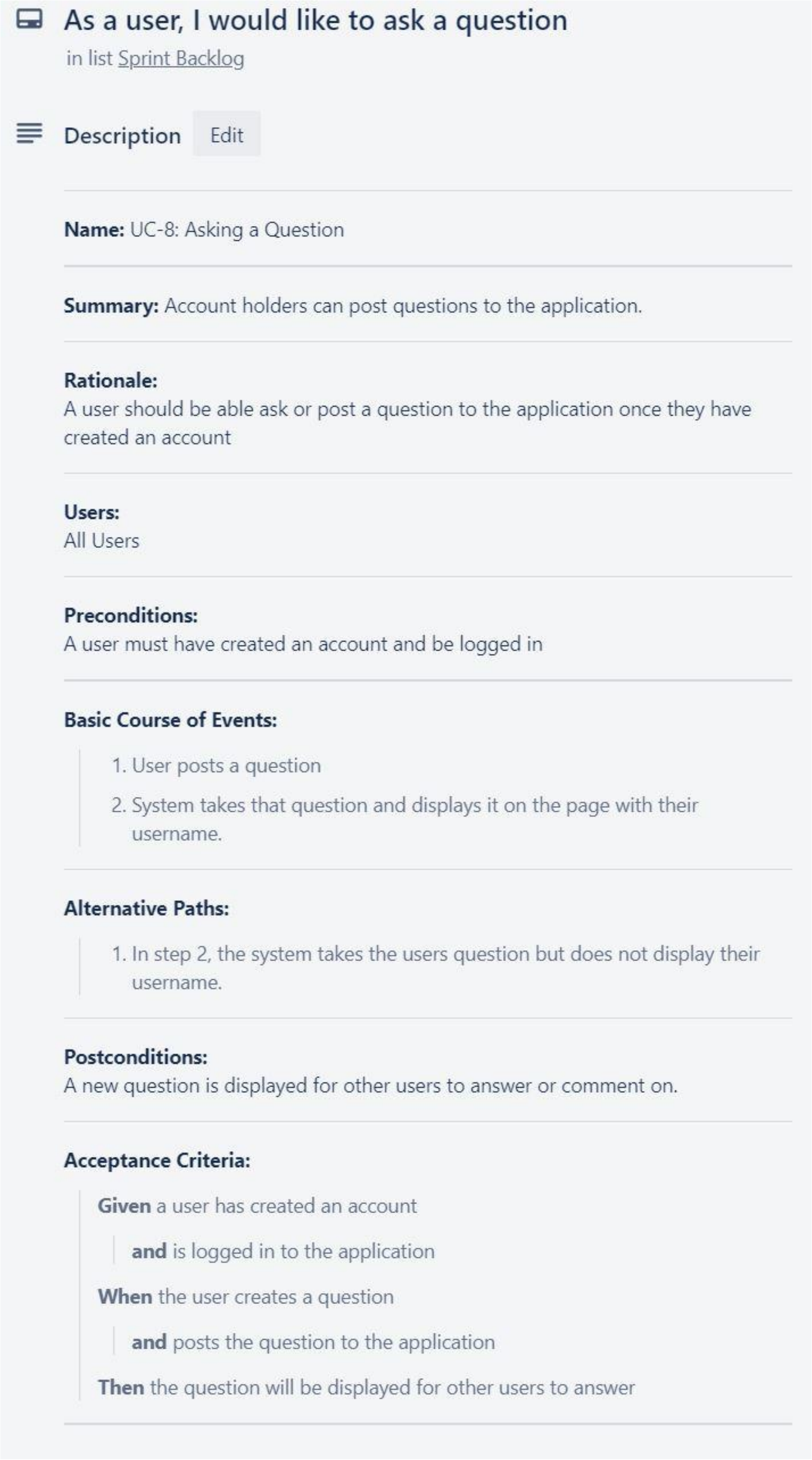
Basic Course of Events:

> 1. The system lists flagged questions for review.
> 2. A moderator views a flagged question or answer that needs to be reviewed and makes adjustments needed on the reviewed question or answer.
> 3. The system will take the marked question or answer off of the flagged list and shows the updates where needed.

Postconditions:
Questions or Answers that need reviewing will be correctly reviewed and proper information will be displayed to the user.

Acceptance Criteria:

> **Given** A user has flagged a question or answer
>
> > **and** a moderator is logged into the system
>
> **When** the moderator navigates to the admin page
>
> > **and** views the question or answer to be reviewed
> > **and** updates the information correctly
>
> **Then** the question or answer will no longer be flagged for review.



As a user, I would like to ask a question

in list Sprint Backlog

Description

Name: UC-8: Asking a Question

Summary: Account holders can post questions to the application.

Rationale:
A user should be able ask or post a question to the application once they have created an account

Users:
All Users

Preconditions:
A user must have created an account and be logged in

Basic Course of Events:

> 1. User posts a question
> 2. System takes that question and displays it on the page with their username.

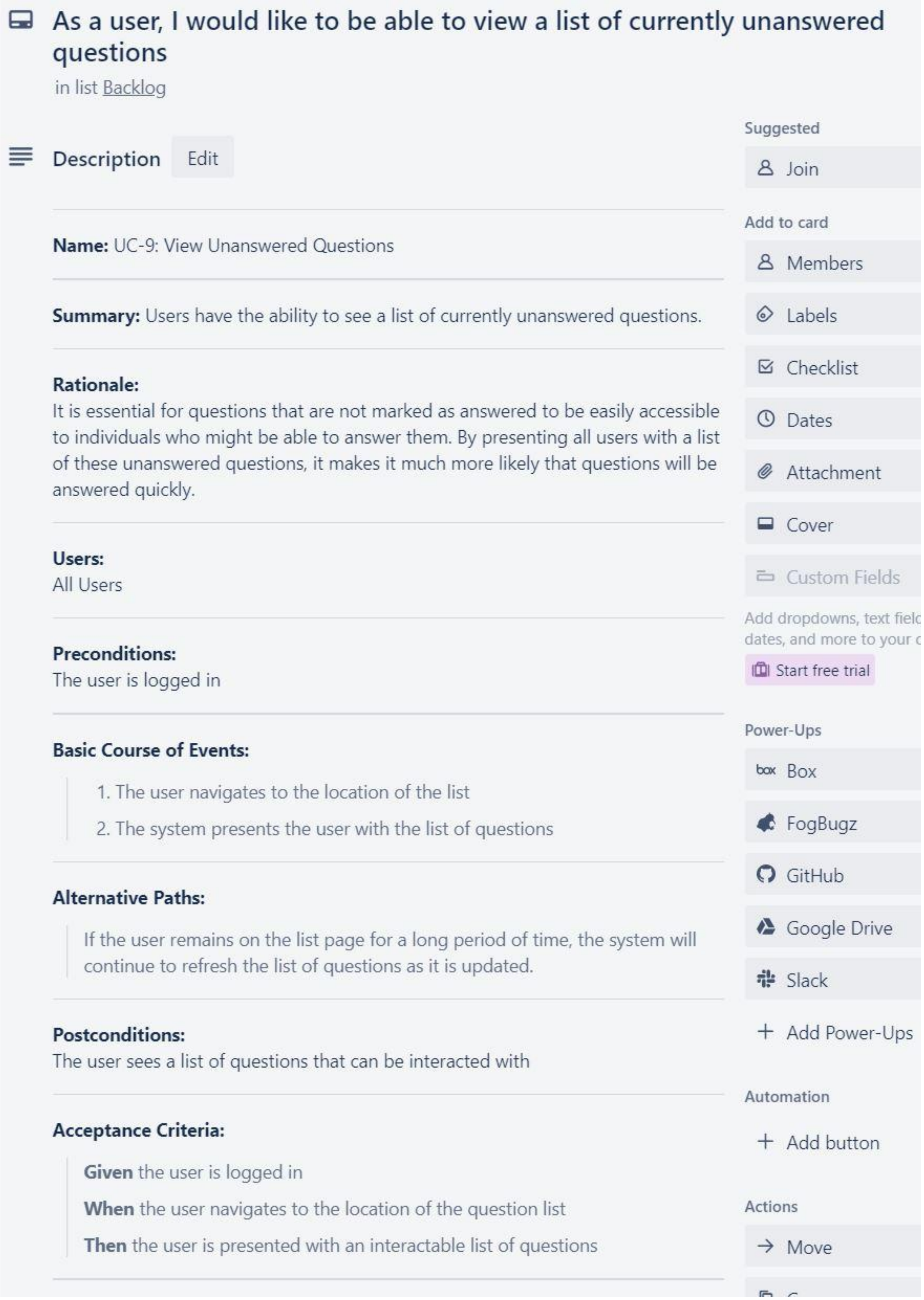
Alternative Paths:

> 1. In step 2, the system takes the users question but does not display their username.

Postconditions:
A new question is displayed for other users to answer or comment on.

Acceptance Criteria:

> **Given** a user has created an account
>
> > **and** is logged in to the application
>
> **When** the user creates a question
>
> > **and** posts the question to the application
>
> **Then** the question will be displayed for other users to answer



As a user, I would like to be able to view a list of currently unanswered questions

in list Backlog

Description Edit

Name: UC-9: View Unanswered Questions

Summary: Users have the ability to see a list of currently unanswered questions.

Rationale:
It is essential for questions that are not marked as answered to be easily accessible to individuals who might be able to answer them. By presenting all users with a list of these unanswered questions, it makes it much more likely that questions will be answered quickly.

Users:
All Users

Preconditions:
The user is logged in

Basic Course of Events:

> 1. The user navigates to the location of the list
> 2. The system presents the user with the list of questions

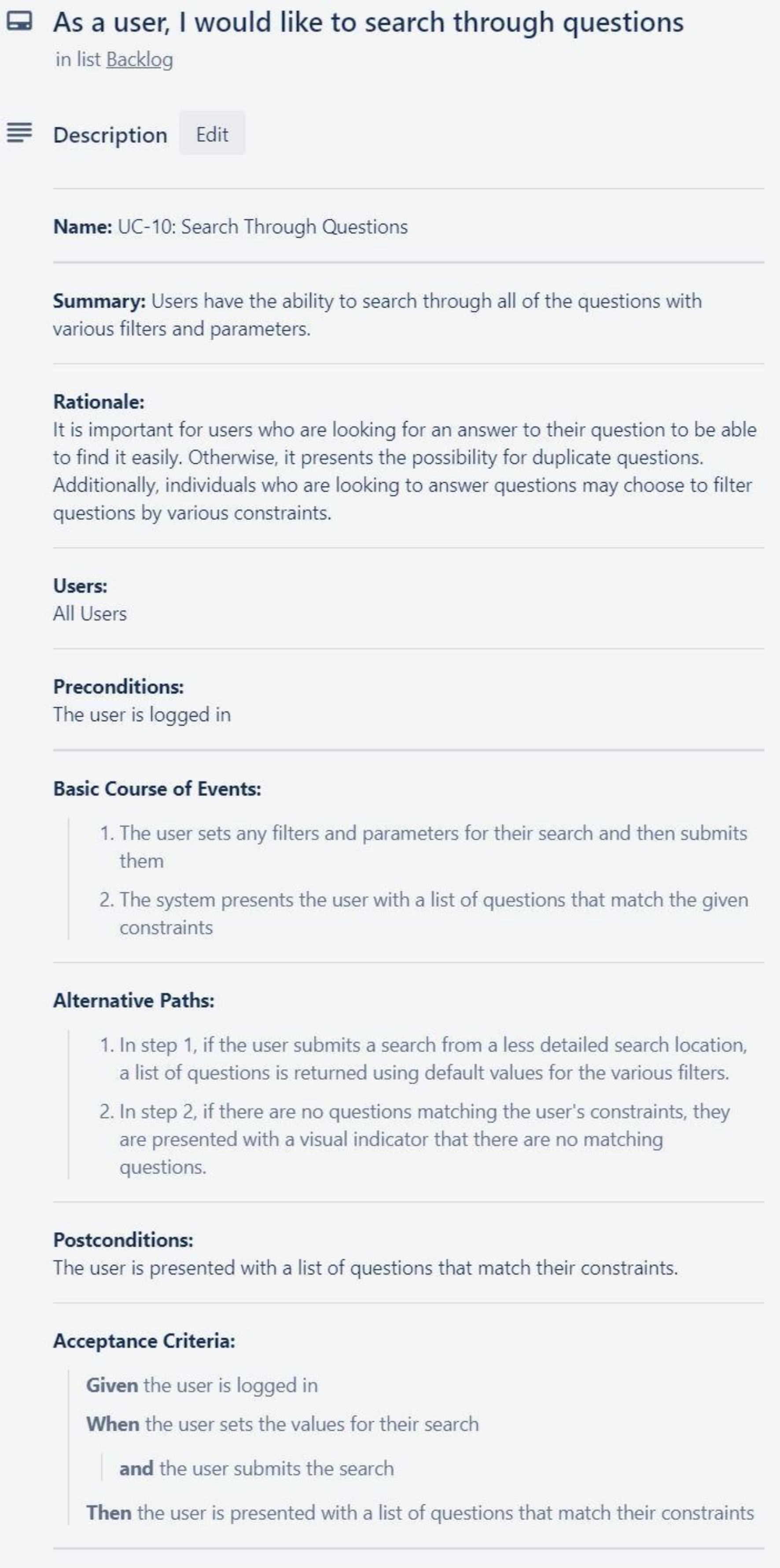
Alternative Paths:

> If the user remains on the list page for a long period of time, the system will continue to refresh the list of questions as it is updated.

Postconditions:
The user sees a list of questions that can be interacted with

Acceptance Criteria:

> **Given** the user is logged in
>
> **When** the user navigates to the location of the question list
>
> **Then** the user is presented with an interactable list of questions



As a user, I would like to search through questions

in list Backlog

Description

Name: UC-10: Search Through Questions

Summary: Users have the ability to search through all of the questions with various filters and parameters.

Rationale:
It is important for users who are looking for an answer to their question to be able to find it easily. Otherwise, it presents the possibility for duplicate questions. Additionally, individuals who are looking to answer questions may choose to filter questions by various constraints.

Users:
All Users

Preconditions:
The user is logged in

Basic Course of Events:

> 1. The user sets any filters and parameters for their search and then submits them
> 2. The system presents the user with a list of questions that match the given constraints

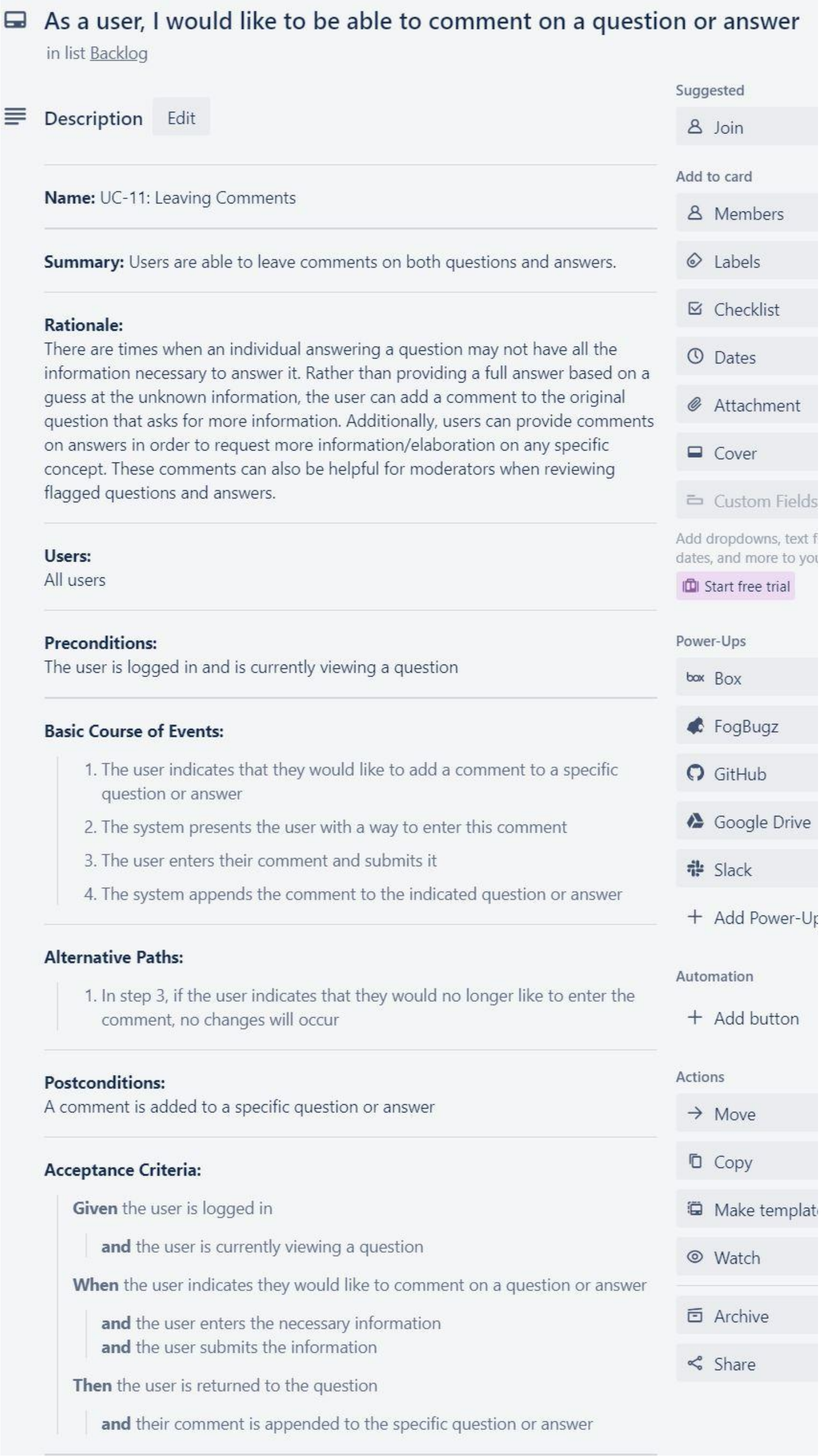
Alternative Paths:

> 1. In step 1, if the user submits a search from a less detailed search location, a list of questions is returned using default values for the various filters.
> 2. In step 2, if there are no questions matching the user's constraints, they are presented with a visual indicator that there are no matching questions.

Postconditions:
The user is presented with a list of questions that match their constraints.

Acceptance Criteria:

> **Given** the user is logged in
>
> **When** the user sets the values for their search
>
> > **and** the user submits the search
>
> **Then** the user is presented with a list of questions that match their constraints



As a user, I would like to be able to comment on a question or answer

in list Backlog

Description

Edit

Name: UC-11: Leaving Comments

Summary: Users are able to leave comments on both questions and answers.

Rationale:
There are times when an individual answering a question may not have all the information necessary to answer it. Rather than providing a full answer based on a guess at the unknown information, the user can add a comment to the original question that asks for more information. Additionally, users can provide comments on answers in order to request more information/elaboration on any specific concept. These comments can also be helpful for moderators when reviewing flagged questions and answers.

Users:
All users

Preconditions:
The user is logged in and is currently viewing a question

Basic Course of Events:

> 1. The user indicates that they would like to add a comment to a specific question or answer
> 2. The system presents the user with a way to enter this comment
> 3. The user enters their comment and submits it
> 4. The system appends the comment to the indicated question or answer

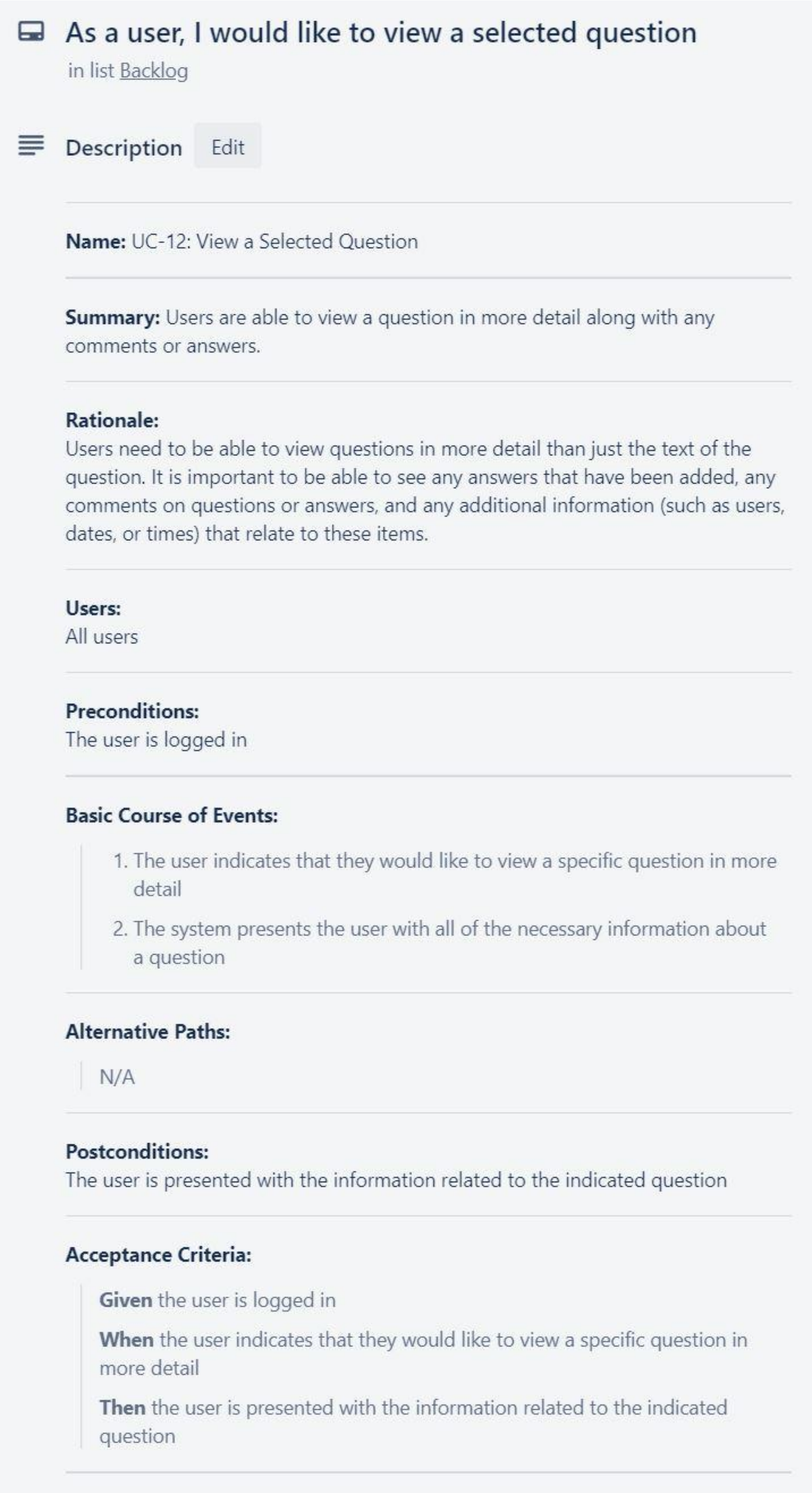
Alternative Paths:

> 1. In step 3, if the user indicates that they would no longer like to enter the comment, no changes will occur

Postconditions:
A comment is added to a specific question or answer

Acceptance Criteria:

> **Given** the user is logged in
>
> > **and** the user is currently viewing a question
>
> **When** the user indicates they would like to comment on a question or answer
>
> > **and** the user enters the necessary information
> > **and** the user submits the information
>
> **Then** the user is returned to the question
>
> > **and** their comment is appended to the specific question or answer



As a user, I would like to view a selected question

in list Backlog

Description

Name: UC-12: View a Selected Question

Summary: Users are able to view a question in more detail along with any comments or answers.

Rationale:
Users need to be able to view questions in more detail than just the text of the question. It is important to be able to see any answers that have been added, any comments on questions or answers, and any additional information (such as users, dates, or times) that relate to these items.

Users:
All users

Preconditions:
The user is logged in

Basic Course of Events:

> 1. The user indicates that they would like to view a specific question in more detail
> 2. The system presents the user with all of the necessary information about a question

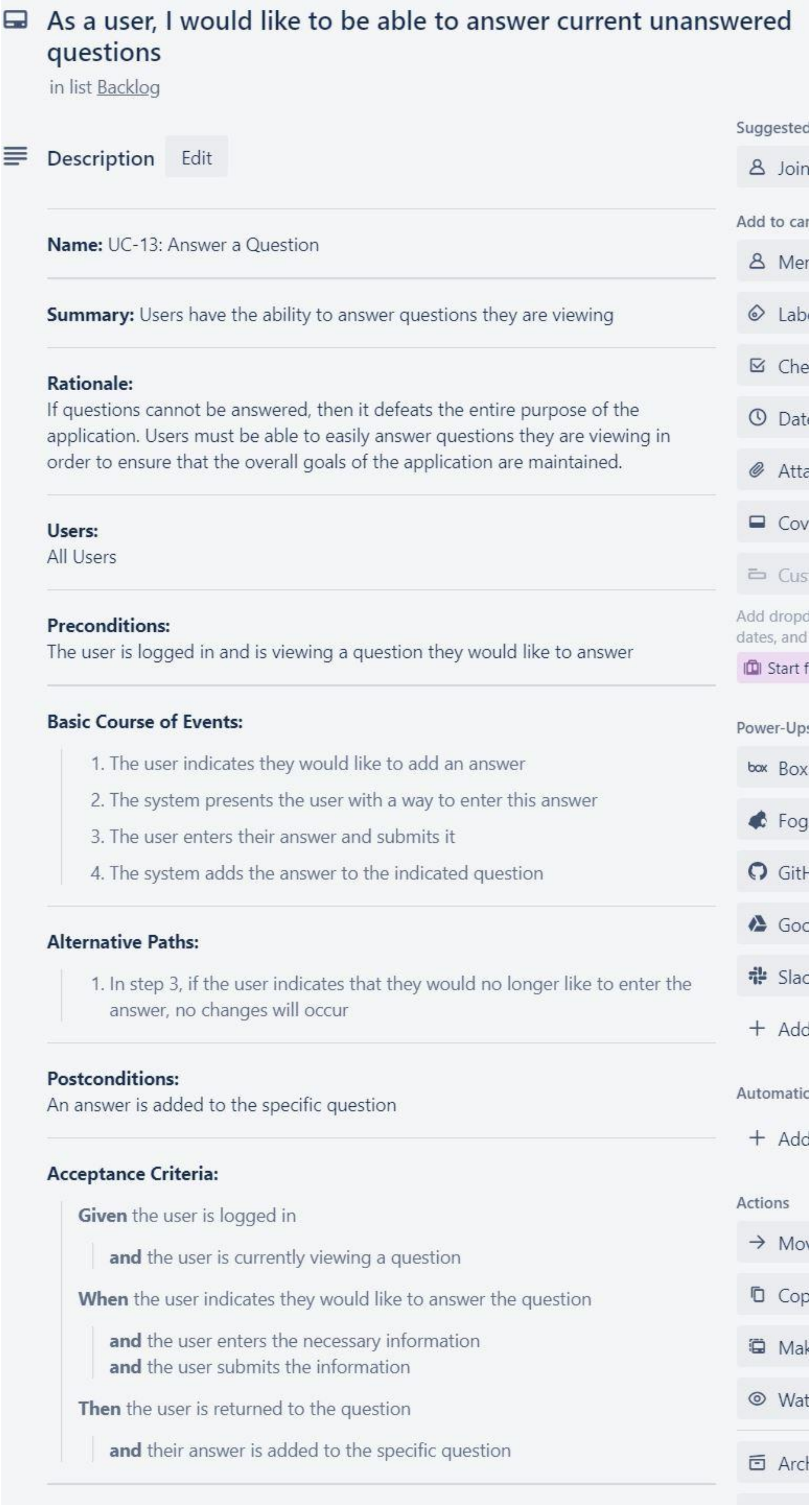
Alternative Paths:

> N/A

Postconditions:
The user is presented with the information related to the indicated question

Acceptance Criteria:

> **Given** the user is logged in
>
> **When** the user indicates that they would like to view a specific question in more detail
>
> **Then** the user is presented with the information related to the indicated question



As a user, I would like to be able to answer current unanswered questions

in list Backlog

Description Edit

Name: UC-13: Answer a Question

Summary: Users have the ability to answer questions they are viewing

Rationale:
If questions cannot be answered, then it defeats the entire purpose of the application. Users must be able to easily answer questions they are viewing in order to ensure that the overall goals of the application are maintained.

Users:
All Users

Preconditions:
The user is logged in and is viewing a question they would like to answer

Basic Course of Events:

> 1. The user indicates they would like to add an answer
> 2. The system presents the user with a way to enter this answer
> 3. The user enters their answer and submits it
> 4. The system adds the answer to the indicated question

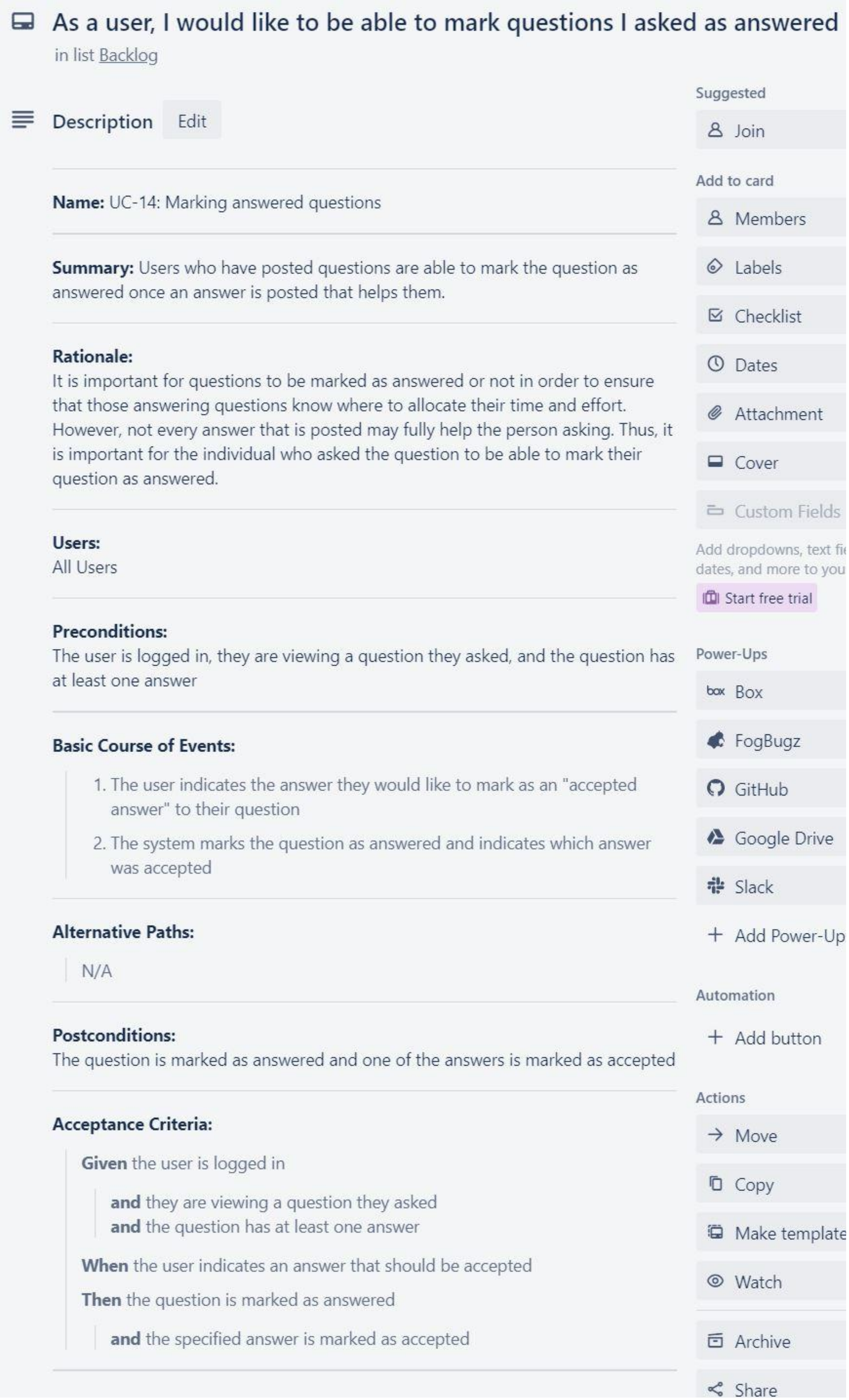
Alternative Paths:

> 1. In step 3, if the user indicates that they would no longer like to enter the answer, no changes will occur

Postconditions:
An answer is added to the specific question

Acceptance Criteria:

> **Given** the user is logged in
>
> > **and** the user is currently viewing a question
>
> **When** the user indicates they would like to answer the question
>
> > **and** the user enters the necessary information
> > **and** the user submits the information
>
> **Then** the user is returned to the question
>
> > **and** their answer is added to the specific question



As a user, I would like to be able to mark questions I asked as answered

in list Backlog

Description Edit

Name: UC-14: Marking answered questions

Summary: Users who have posted questions are able to mark the question as answered once an answer is posted that helps them.

Rationale:
It is important for questions to be marked as answered or not in order to ensure that those answering questions know where to allocate their time and effort. However, not every answer that is posted may fully help the person asking. Thus, it is important for the individual who asked the question to be able to mark their question as answered.

Users:
All Users

Preconditions:
The user is logged in, they are viewing a question they asked, and the question has at least one answer

Basic Course of Events:

> 1. The user indicates the answer they would like to mark as an "accepted answer" to their question
> 2. The system marks the question as answered and indicates which answer was accepted

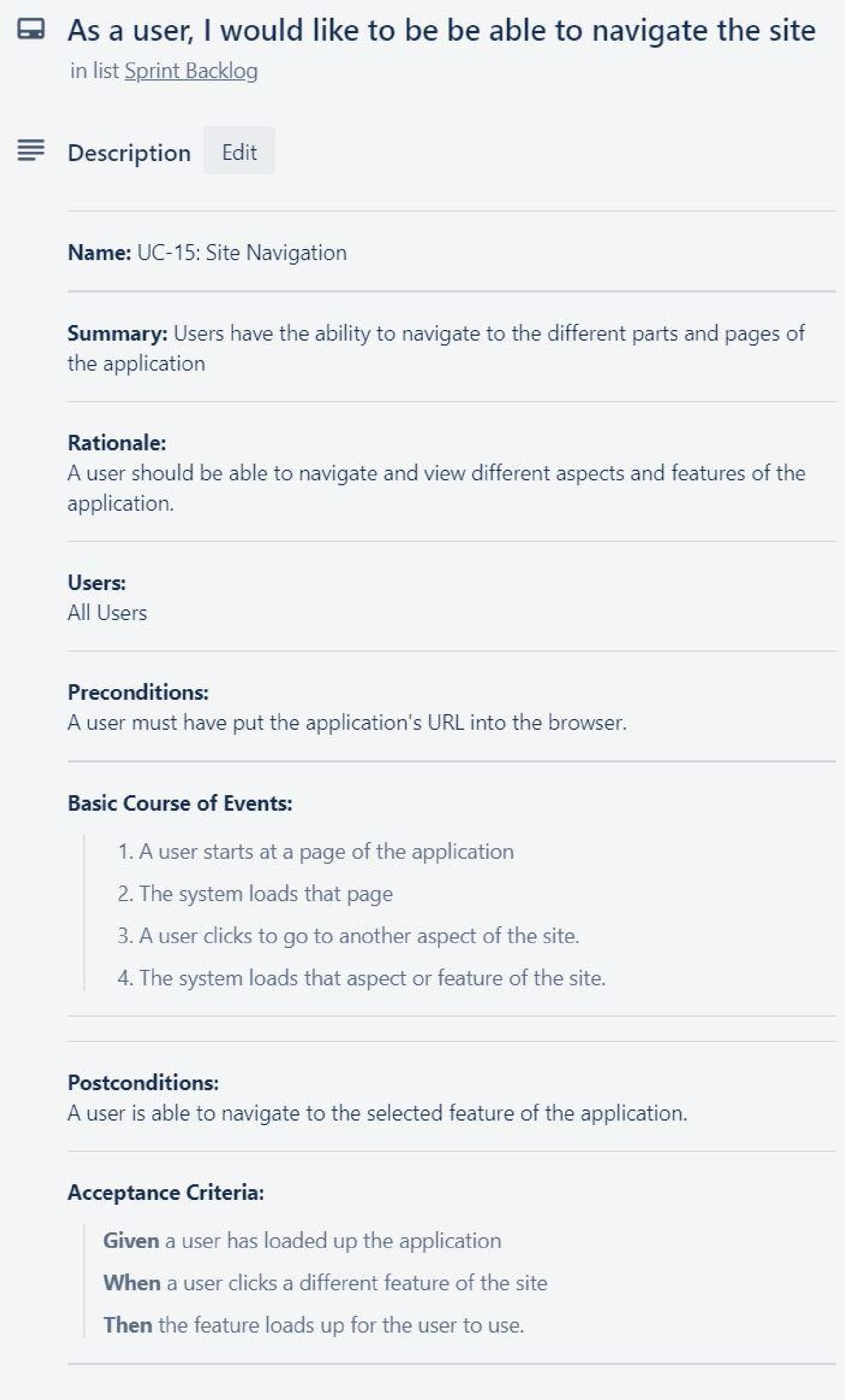
Alternative Paths:

> N/A

Postconditions:
The question is marked as answered and one of the answers is marked as accepted

Acceptance Criteria:

> **Given** the user is logged in
>
> > **and** they are viewing a question they asked
> > **and** the question has at least one answer
>
> **When** the user indicates an answer that should be accepted
>
> **Then** the question is marked as answered
>
> > **and** the specified answer is marked as accepted



As a user, I would like to be be able to navigate the site

in list Sprint Backlog

Description

Name: UC-15: Site Navigation

Summary: Users have the ability to navigate to the different parts and pages of the application

Rationale:
A user should be able to navigate and view different aspects and features of the application.

Users:
All Users

Preconditions:
A user must have put the application's URL into the browser.

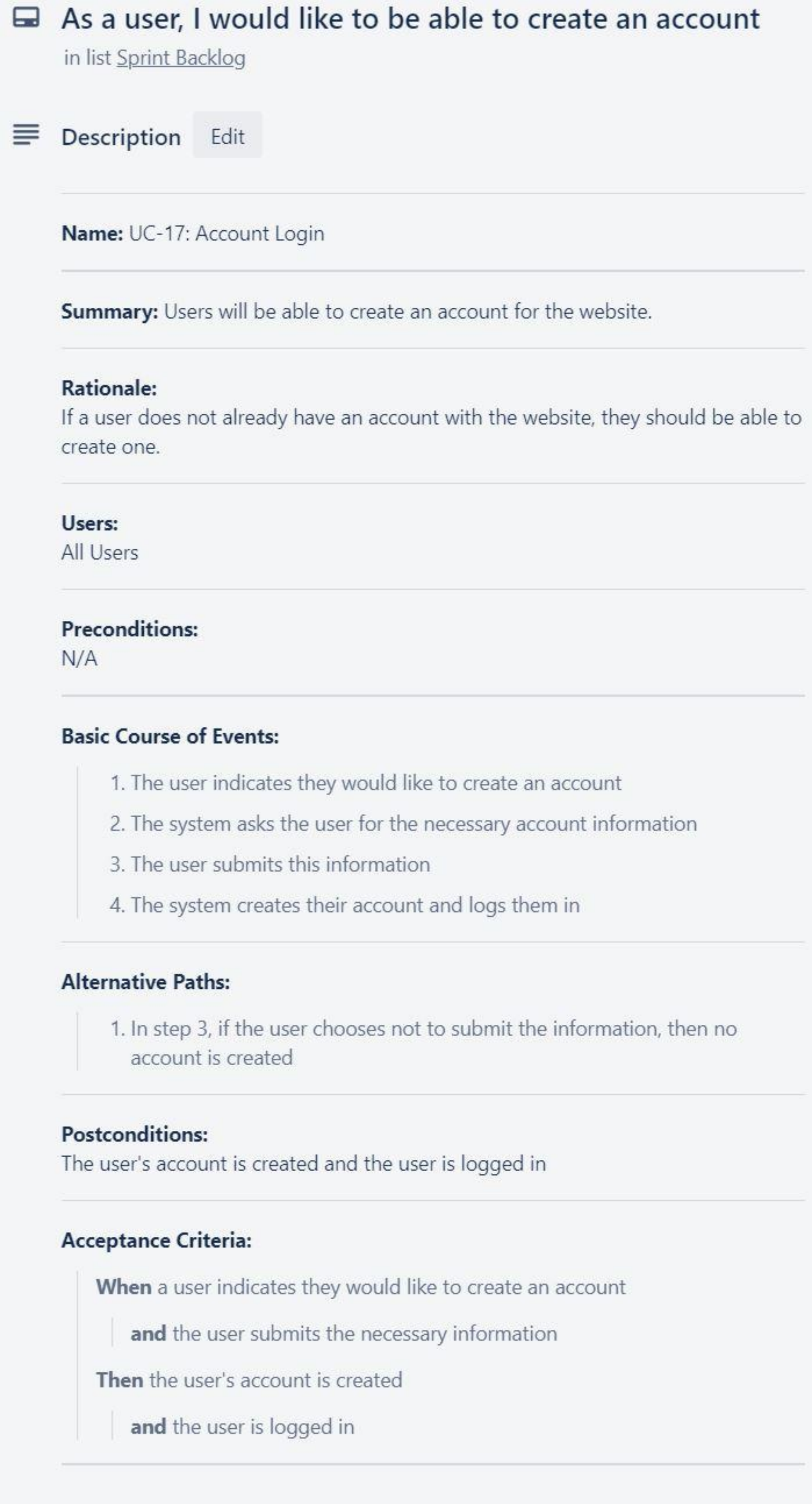
Basic Course of Events:

> 1. A user starts at a page of the application
> 2. The system loads that page
> 3. A user clicks to go to another aspect of the site.
> 4. The system loads that aspect or feature of the site.

Postconditions:
A user is able to navigate to the selected feature of the application.

Acceptance Criteria:

> **Given** a user has loaded up the application
>
> **When** a user clicks a different feature of the site
>
> **Then** the feature loads up for the user to use.



As a user, I would like to be able to create an account

in list Sprint Backlog

Description

Name: UC-17: Account Login

Summary: Users will be able to create an account for the website.

Rationale:
If a user does not already have an account with the website, they should be able to create one.

Users:
All Users

Preconditions:
N/A

Basic Course of Events:

> 1. The user indicates they would like to create an account
> 2. The system asks the user for the necessary account information
> 3. The user submits this information
> 4. The system creates their account and logs them in

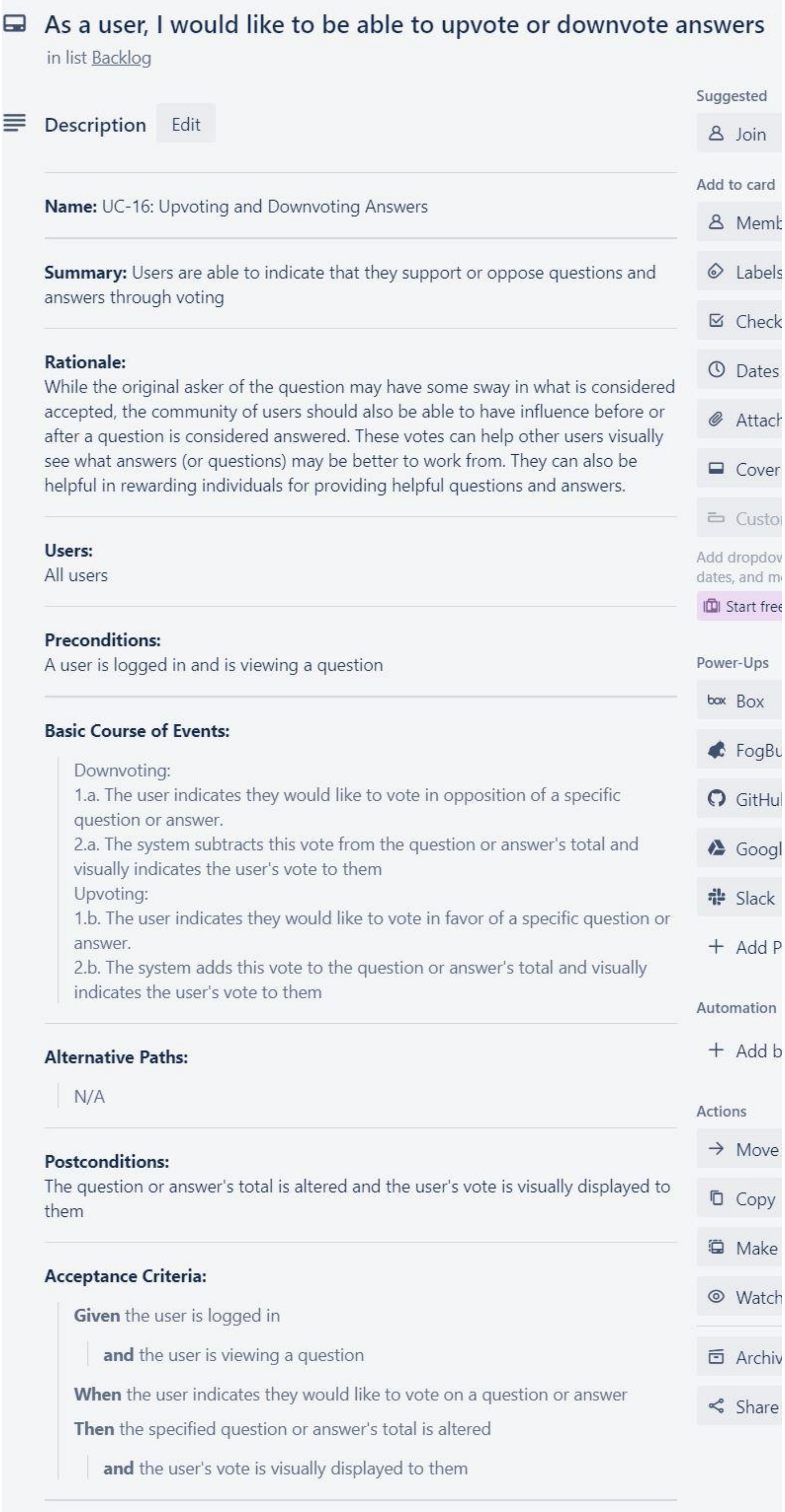
Alternative Paths:

> 1. In step 3, if the user chooses not to submit the information, then no account is created

Postconditions:
The user's account is created and the user is logged in

Acceptance Criteria:

> **When** a user indicates they would like to create an account
>
> > **and** the user submits the necessary information
>
> **Then** the user's account is created
>
> > **and** the user is logged in



As a user, I would like to be able to upvote or downvote answers

in list Backlog

Description

Edit

Name: UC-16: Upvoting and Downvoting Answers

Summary: Users are able to indicate that they support or oppose questions and answers through voting

Rationale:
While the original asker of the question may have some sway in what is considered accepted, the community of users should also be able to have influence before or after a question is considered answered. These votes can help other users visually see what answers (or questions) may be better to work from. They can also be helpful in rewarding individuals for providing helpful questions and answers.

Users:
All users

Preconditions:
A user is logged in and is viewing a question

Basic Course of Events:

> Downvoting:
> 1.a. The user indicates they would like to vote in opposition of a specific question or answer.
> 2.a. The system subtracts this vote from the question or answer's total and visually indicates the user's vote to them
> Upvoting:
> 1.b. The user indicates they would like to vote in favor of a specific question or answer.
> 2.b. The system adds this vote to the question or answer's total and visually indicates the user's vote to them

Alternative Paths:

> N/A

Postconditions:
The question or answer's total is altered and the user's vote is visually displayed to them

Acceptance Criteria:

> **Given** the user is logged in
>
> > **and** the user is viewing a question
>
> **When** the user indicates they would like to vote on a question or answer
>
> **Then** the specified question or answer's total is altered
>
> > **and** the user's vote is visually displayed to them