**Augur Semester Project**

**Phase 1: Use Cases**

**Group 16**

**Hamza Thabit, Paul Austin Hales, Yizhen Wang**

**Table of Contents**

**Collecting Open Source Project Info …............................. 3**

**Editing Open Source Project Info ………………………….. 6**

**Viewing Open Source Project Info ………………………….9**

**Credits …………………………………………………………. 12**

## **Use Case 1:**

## **Collecting Open Source Project Info**

## **Description**

Community managers fill out information form about their project. This can include a short description, goals, needed technical skills, and link to the project’s repository. The information provided will be stored so that it can later edited by the manager or viewed by other people interested in the project. Information provided by the community manager will allow people interested in joining open source projects to get short overview about different projects.

## **Triggers**

Community manager wants to improve the newcomer experience or grow their project community. Verified community manager clicks Project Info button and is taken to form to describe their project

## **Actors**

Community managers of open source projects

## **Preconditions**

User has some sort of authentication that they are a community manager. Community Manager will have a list of projects that they oversee

## **Main Success Scenario (Goals)**

Community Manager fills out any pertinent information about their project. The data is then stored in a database so that it can be viewed later by any newcomers to the open source project or can be edited by the community manager.

## **Alternate Success Scenarios**

Community manager could provide some text file that could be saved and either downloaded or viewed in the browser by newcomers.

Information form provided by community manager is saved, but no statistics about the project could be retrieved

## **Failed End Condition**

Community manager does not think this functionality will increase newcomer experience or grow their community numbers. As a result of this, they will not use the service.

Information that manager entered was not able to be inserted into the database. The information will now not be able to be retrieved at a later time.

## **Extensions**

Project Information could be extended to cover repositories as well

Community managers able to later edit their project description

## **Steps of Execution (Requirements)**

1. Community Manager verifies their identity
2. Community manager fills out information form
3. Manager clicks submit
4. Information is saved into database

## 

## 

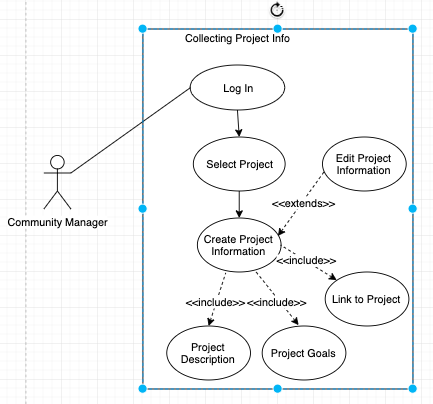
## 

## 

## 

## 

## **Collecting Project Information**



After logging in, the community manager will select from their list of projects the project that they wish to create an information page about. The community manager will then fill out a form about different aspects of their project. After they have completed the form, the manager will submit the form, and it will be entered into the database.

## **Dependent Use Cases**

Editing Open Source Project Information

Viewing Open Source Project Information

## 

## **Use Case 2:**

## **Editing Open Source Project Information**

## **Description**

Community managers wish to change the project description, goals, or other information about their project. The previous data that they inputted will be loaded from the database. The manager will then be able to change any information or make any corrections. The new changes will then be applied to the entry in the database.

## **Triggers**

Community Manager wishes to update information about their project over time. Verified community manager clicks Edit Info button and is taken to info form.

## **Actors**

Community managers of open source projects

## **Preconditions**

User has some sort of authentication that they are a community manager. Information about the selected project has already been submitted and inserted into the database.

## **Main Success Scenario (Goals)**

Community Manager fills out any pertinent information about their project. The data is then stored in a database so that it can be viewed later by any newcomers to the open source project or can be edited by the community manager.

## **Alternate Success Scenarios**

Community Manager has not previously filled out an information form about the project they are looking to update. In this situation, the user will be prompted to the collecting project info page and will fill out the information form.

## **Failed End Condition**

Previous information about the project could not be retrieved from the database. New changes made by the user could not be made and applied to database.

Updating the project information does not help the community manager convey what the new goals and information about the project are

## **Extensions**

This could be extended to cover repositories as well

## **Steps of Execution (Requirements)**

1. Community Manager verifies their identity
2. Community manager fills out information form
3. Manager clicks submit
4. Information is updated in the database

## 

## 

## 

## 

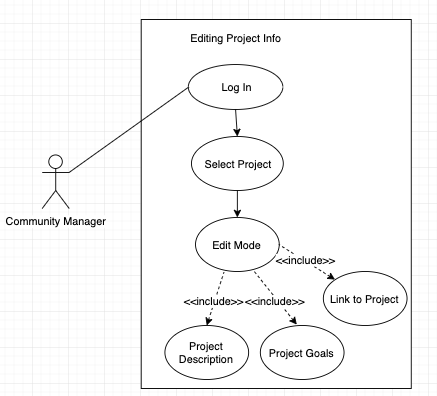
## 

## 

## 

## 

## **Editing Project Information**



After logging in, the community manager will be presented a list of their projects. The manager can then chose the project that they wish to edit it’s information. The completed form about the project will be loaded. Then, the community manager can then edit any part of the information form. After completion, the new form will then be updated in the database.

## 

## 

## 

## 

## 

## 

## **Use Case 3:**

## **Viewing Open Source Project Information**

## **Description**

People interested in contributing to open source projects can view introductory information about a variety of open source projects. This information could include a short description, goals, needed technical skills, and link to a repository. It allows people that are unfamiliar with open source projects to view basic information about projects so that they may determine if that project interests them and can then learn more about it.

## **Triggers**

Newcomer is interested in joining an open source project. User finds one that interests them and wants to learn about the project.

## **Actors**

Newcomers to open source projects

## **Preconditions**

Community Managers have provided introductory information about their projects. If no managers have not filled out the form, there will be no information to display to the newcomer.

## **Main Success Scenario (Goals)**

User selects the project they want to know more about. Information previously filled out by the community manager will be displayed. Metrics about the project will also be displayed. Likely metrics that will be included are: number of contributors, lines added, and number of commits.

## 

## **Alternate Success Scenarios**

Metrics about the project could not be retrieved, but information about the project provided by the community manager is retrieved and displayed.

A description about the project could not be retrieved or has not been filled out by the community manager. Metrics, however, are retrieved and displayed.

## **Failed End Condition**

No information (description or metrics) about the searched or selected project could be retrieved.

User does not find information provided about the project helpful. They decide not to join the project because they do not know enough about it

## **Extensions**

User could be allowed to search/ select repositories and projects.

## **Steps of Execution (Requirements)**

1. User searches for or selects project from list
2. Information from community manager is loaded
3. Metrics about project are retrieved and then displayed

## 

## 

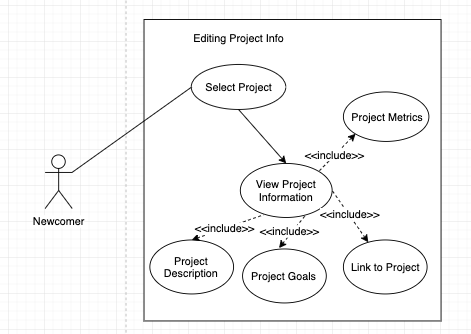
## 

## 

## 

## 

## **Viewing Project Information**



A user interested in joining an open source project can search or select from a list an open source project to learn more about. If the community manager of the project selected has filled out an information about their project, that information will then be displayed for the newcomer to view.

**Credits**

**Use Case 1**

* **Use Case Information: Austin Hales**
* **Use Case Diagram: Austin Hales**
* **Peer Review: Hamza Thabit & Yizhen Wang**

**Use Case 2**

* **Use Case Information: Austin Hales**
* **Use Case Diagram: Austin Hales**
* **Peer Review: Hamza Thabit & Yizhen Wang**

**Use Case 3**

* **Use Case Information: Austin Hales**
* **Use Case Diagram: Austin Hales**
* **Peer Review: Hamza Thabit & Yizhen Wang**