

CS 320 Course Project - Software Design Document

for

JournalJay

Prepared by

Sizzle Snap

|  |  |  |
| --- | --- | --- |
| Laurel Anderson | 11638131 | laurel.o.anderson@wsu.edu |
| Irina Bejan | 11613022 | Irina.bejan@wsu.edu |

|  |  |
| --- | --- |
|  |  |
| Date: | 11/20/20 |
|  |  |
|  |  |
|  |  |

Content

Contents ii

1 Introduction 1

1.1 Project Overview 1

1.2 Definitions, Acronyms and Abbreviations 1

1.3 References and Acknowledgments 1

2 Activity Diagrams 2

2.1 Making a Journal Entry 2

2.2 Removing A User 3

3 Class Diagram 4

3.1 JournalJay Classes 4

4 Behavioral Diagram(s) 5

4.1 Making An Entry 5

4.2 Removing A User 7

Appendix A - Group Log 8

# Introduction

## Project Overview

For this project, JournalJay’s system is modeled using three different types of diagrams: activity, class, and state. There are two different activity diagrams, one for making a journal entry, and one for an admin removing a user. The third diagram is a class diagram. It shows all the predicted classes and methods with their relations. The fourth and fifth diagrams are state diagrams. The fourth diagrams models making a user entry and the fifth diagram models removing a user. Both state diagrams have state tables that show transitions.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Admin/  Administrator | Someone who is given specific permissions to manage and control the system. |
| State | Observable behavers of a system at a given time. |
| State Transition | Moving from one state in a system to another. |
| Event | An incident that makes the system do something predictable. |
| Action | A process that occurs immediately following a specific action as a result of that action. |
| Class Diagram | Shows the classes and models within a system and their relations with each other. |
| Activity Diagram | Shows the graphic representation of activities within a situation. |
| State Diagram | Shows the states of a systems and the actions that move the system state to state. |
| CS | Computer science |
| User | Someone who interacts with the web application. |

## References and Acknowledgments

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended

Practice for Software Requirements Specifications”, October 20, 1998.

# Activity Diagrams

## Making a Journal Entry

The user will be signed in at the start of this diagram. When they go to make a journal entry, the system generates a list of the color presets the user can choose from. The user can then choose what color the background of their journal entry can be. The computer will then generate mood presets and the user will choose one. From there, the user can review the entry and either accept or reject it. If the user rejects it, they will be sent back to the start of the process. If the user accepts the entry, they will be redirected to where they can review all previous entries.

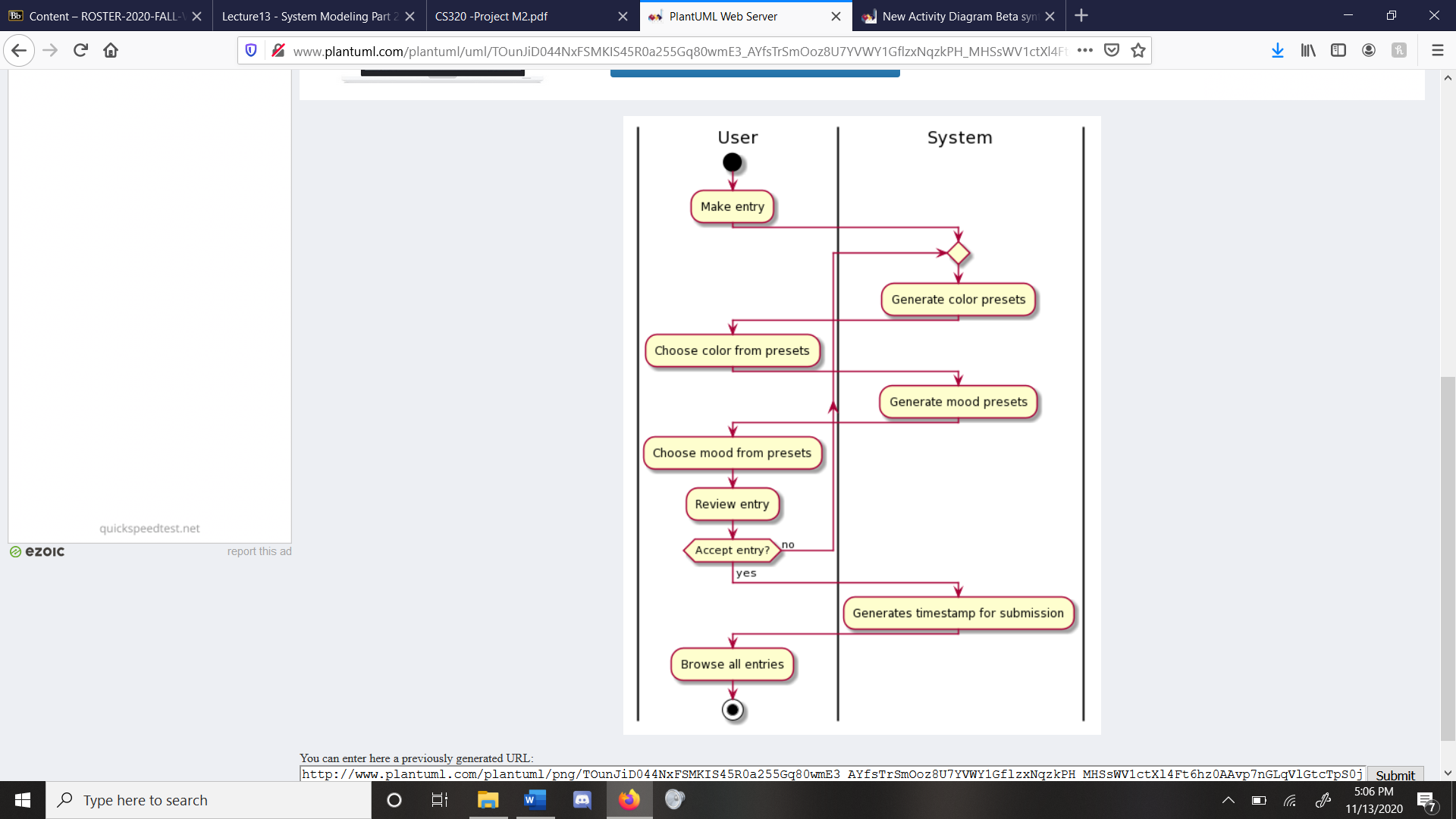


Figure 1: Making a Journal Entry

## Removing a User

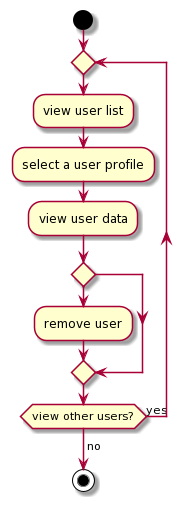
The administrator will be signed in at the start of this diagram. They have access to select and view any user’s data, as well as remove users if necessary. Users will be instructed to contact the dedicated Admin account if they choose to remove their account and all its data permanently.

Figure 2: Removing a User

# Class Diagram

## JournalJay Classes

Figure 3: JournalJay Classes

|  |  |
| --- | --- |
| **Class** | **Use** |
| UserProfile | Each user profile will be of this type; the class stores basic information including their username, password, an answer to a security question in “forgot,” true or false if they are an admin, and a list of their entries. |
| UserList | This is the grand list of all users in the system. We’ll need this to search at login to be able to report “wrong username or password,” as well as to retrieve a profile at correct login credentials. |
| EntryList | This is the list of all entries a user has. The list is checked before the user can make a new entry for the day and referenced when the user wishes to browse their previous entries. |
| Entry | Each user entry will be of this type; the class stores all the features of a journal entry. |
| Moods | This is essentially a list of all the moods a user can choose from. It will be in its own class for organization, especially in the case of adding increased functionality later. |
| Colors | This is essentially a list of all the background colors a user can choose from. It will be in its own class for organization, especially in the case of adding increased functionality later. |

# Behavioral Diagrams

## Making a Journal Entry State Diagram

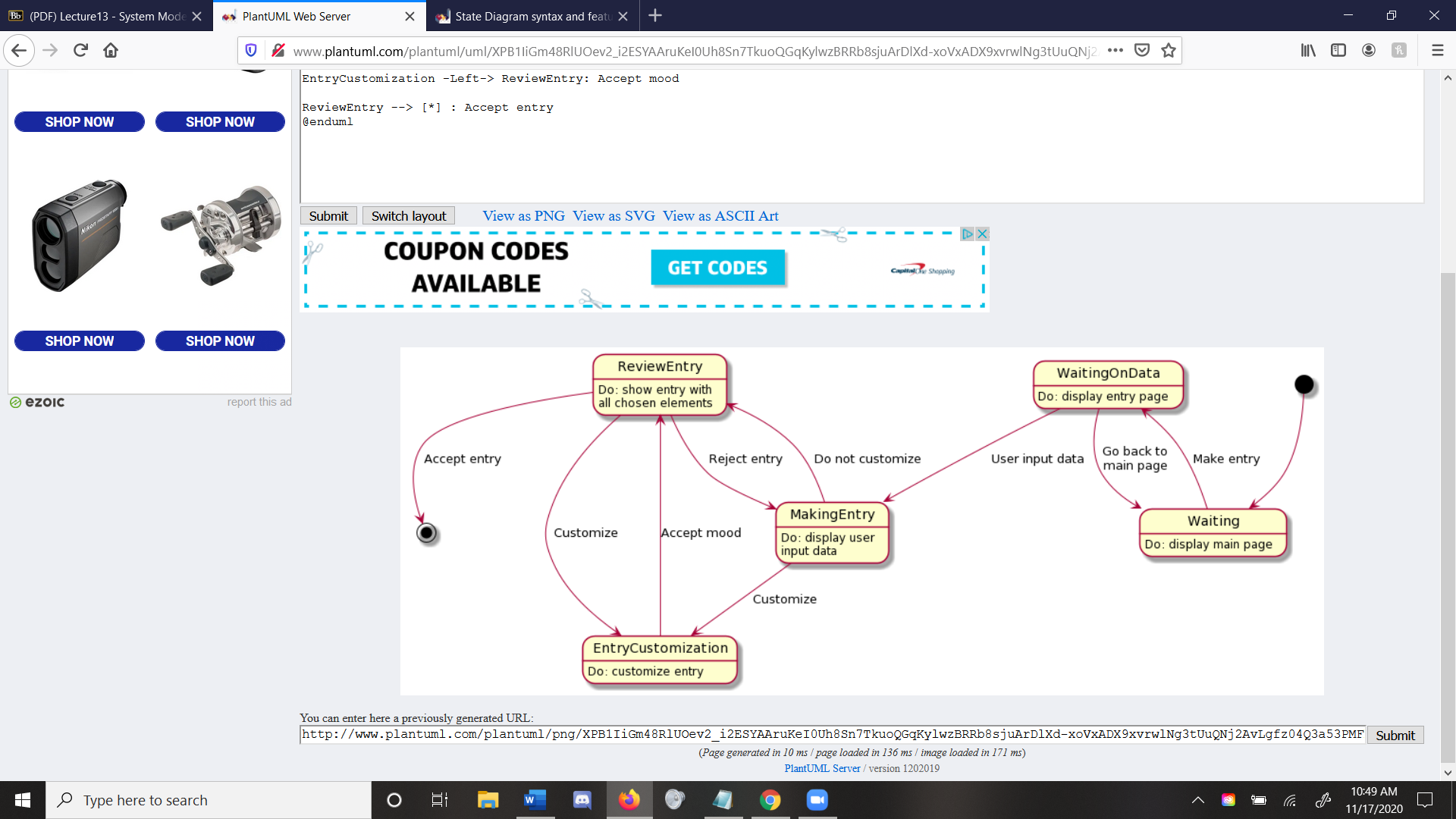
This is a state diagram for making a journal entry. The user will be signed into the site at the beginning of this process, and the system will check to make sure they are not admin before the first state.

Figure 5: Entry Customization

Figure 4: State Diagram for Making a Journal Entry

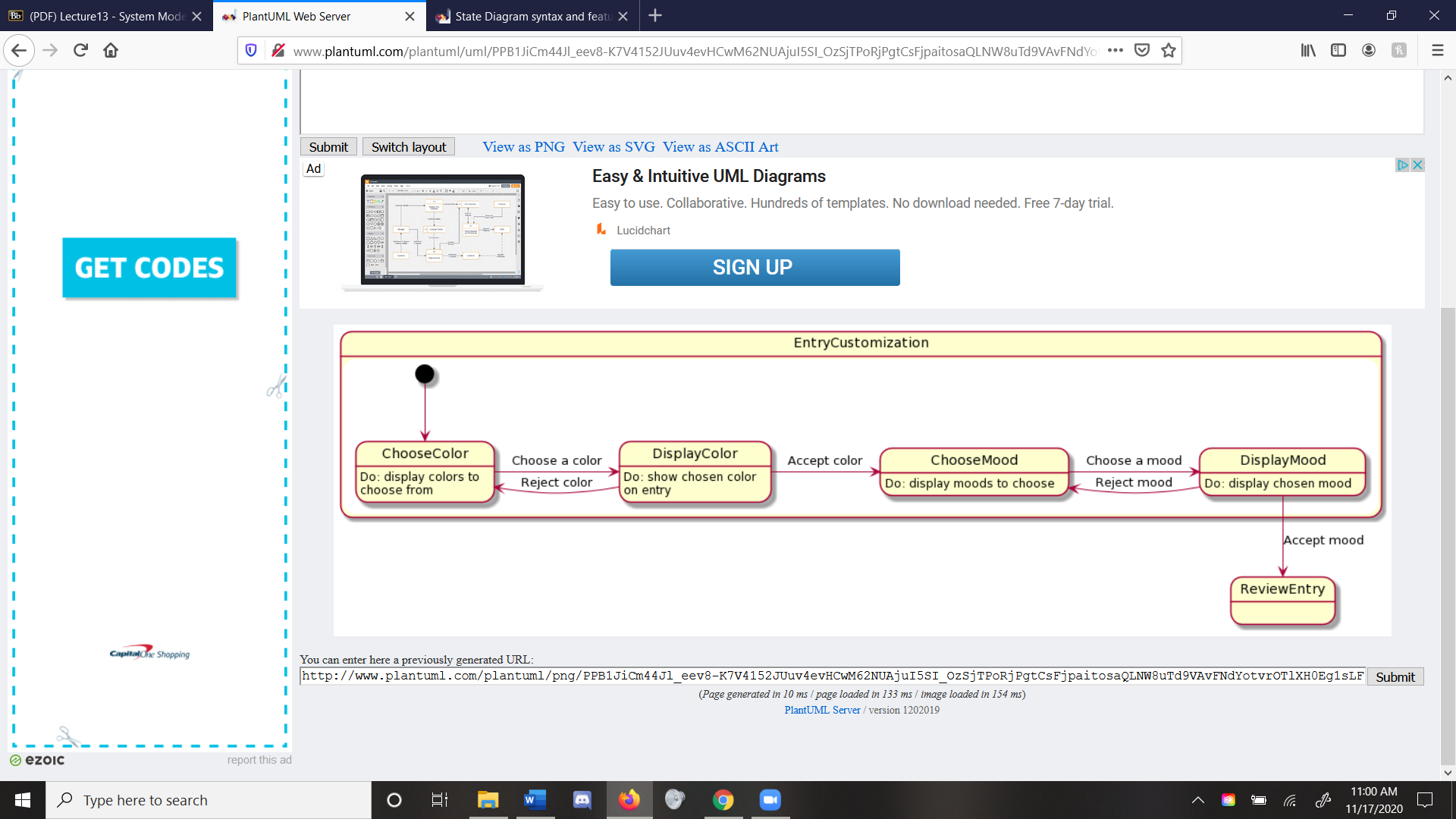


Table 1: Make a Journal Entry

|  |  |  |  |
| --- | --- | --- | --- |
| **Start** | **Transition** | **End** | **Description** |
| Waiting | Make entry | WaitingOnData | The system displays the main page until the user chooses to make an entry. |
| WaitingOnData | Go back to main page | Waiting | The user can choose to return to the main menu without creating a journal entry. |
| WaitingOnData | User inputs data | MakingEntry | Here, the user is actively inputting data into the journal entry. |
| MakingEntry | Customize | EntryCustomization | Once the user is done inputting data into the entry, they can then choose to customize the entry. The EntryCusomization table has the transitions for the class. |
| MakingEntry | Do not customize | ReviewEntry | The user can also choose not to customize the newly created entry and skip to reviewing it. |
| ReviewEntry | Reject entry | MakingEntry | The user can decide to change the user input data or customize from this stage. |
| ReviewEntry | Accept entry | FinalState | The user can accept the journal entry. |

Table 2: Entry Customization

|  |  |  |  |
| --- | --- | --- | --- |
| **Start** | **Transition** | **End** | **Description** |
| ChooseColor | Choose a color | DisplayColor | The user can choose a color for the journal entry and the system then goes to display it. |
| DisplayColor | Accept color | ChooseMood | The user then can choose a mood and the system displays it on the journal entry. |
| DisplayColor | Reject color | ChooseColor | The user can choose to reject the color and go to choose a new color. |
| ChooseMood | Choose a mood | DisplayMood | The user can choose a mood and it will display on the journal entry. |
| DisplayMood | Reject mood | ChooseMood | The user can return to choose a different mood by rejecting the current mood. |
| DisplayMood | Accept mood | FinalState | The user can accept all customization options and go to the next state outside of customizing. |

## Removing a User

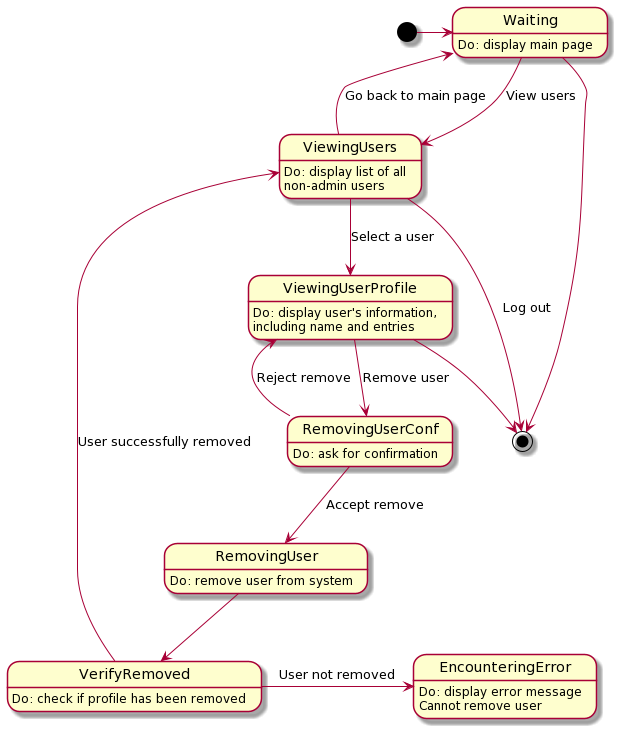
This is a state diagram for making a journal entry. The administrator will be signed into the site at the beginning of this process, and the system will check their administrative status before loading the main page.

Figure 6: User Removal

Table 3: Removing a User

|  |  |  |  |
| --- | --- | --- | --- |
| **Start** | **Transition** | **End** | **Description** |
| Waiting | View users | ViewingUsers | Admin can select to view users. |
| Waiting | Log out | FinalState | Admin can log out from here. |
| ViewingUsers | Select a user | ViewingUserProfile | Admin selects a user, and the system retrieves their profile. |
| ViewingUsers | Log out | FinalState | Admin can log out from here. |
| ViewingUserProfile | Remove user | RemovingUserConf | Admin can select to remove user. |
| ViewingUserProfile | Log out | FinalState | Admin can log out from here. |
| RemovingUserConf | Accept remove | RemovingUser | System will verify they want to remove; if accept, proceeds. |
| RemovingUserConf | Reject remove | ViewingUserProfile | If reject, goes back to their profile. |
| RemovingUser |  | VerifyRemoved | After removing a user, the system will automatically perform a check. |
| VerifyRemoved | User not removed | EncounteringError | If the user was not removed, it will throw an error. |
| VerifyRemoved | User successfully removed | ViewingUsers | If removed, admin will be sent back to page of users. |

Appendix A - Group Log

10-2-20 10:00am–10:20am Filled out the team agreement form.

10-9-20 9:00am-9:25am Drafted a schedule and rough timeline to complete the SRS document. Discussed concerns and initial ideas in our project requirements.

10-16-20 9:00am-9:45am Sectioned out the SRS document. Set initial deadline of first draft for 10-25-20).

10-30-20 9:45am-10:30am Talked about project specifications and worked on the SRS document.

11-04-20 4:00pm-5:30pm Discussed outstanding changes and set up GitHub repo for teamwork.

11-06-20 12:00pm-1:00pm Made final necessary adjustments.

11-13-20 12:00pm – 1:00pm Met via FaceTime and discussed the class diagram. Brainstormed the classes and methods that we may use.

11-20-20 Discussed any missing and improvable elements for the diagrams over text message.