

CS 320 Course Project - Software Design Document

for

JournalJay

Prepared by

Group Name: Sizzle Snap

|  |  |  |
| --- | --- | --- |
| Laurel Anderson | 11638131 | laurel.o.anderson@wsu.edu |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |

|  |  |
| --- | --- |
|  |  |
| 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 Diagram(s) 2

2.1 Making a Journal Entry 2

2.2 D-2 2

2.2 D-3 2

3 Class Diagram(s) 3

3.1 D-1 3

3.2 D-2 3

3.2 D-3 3

4 Behavioral Diagram(s) 4

4.1 D-1 4

4.2 D-2 4

4.3 D-3 4

Appendix A - Group Log 5

< A table of contents

Edit this table as needed to suit the page numbers and titles of the diagrams you included. >

# Introduction

*<TO DO: Please provide a brief introduction to your project.>*

## 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

<Define all the terms necessary to properly interpret the report, including acronyms and abbreviations.

TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>

|  |  |
| --- | --- |
| **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

<List any other documents or Web addresses to which this document refers

TO DO: Use the standard IEEE citation guide for this section.>

# Activity Diagram(s)

## 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.

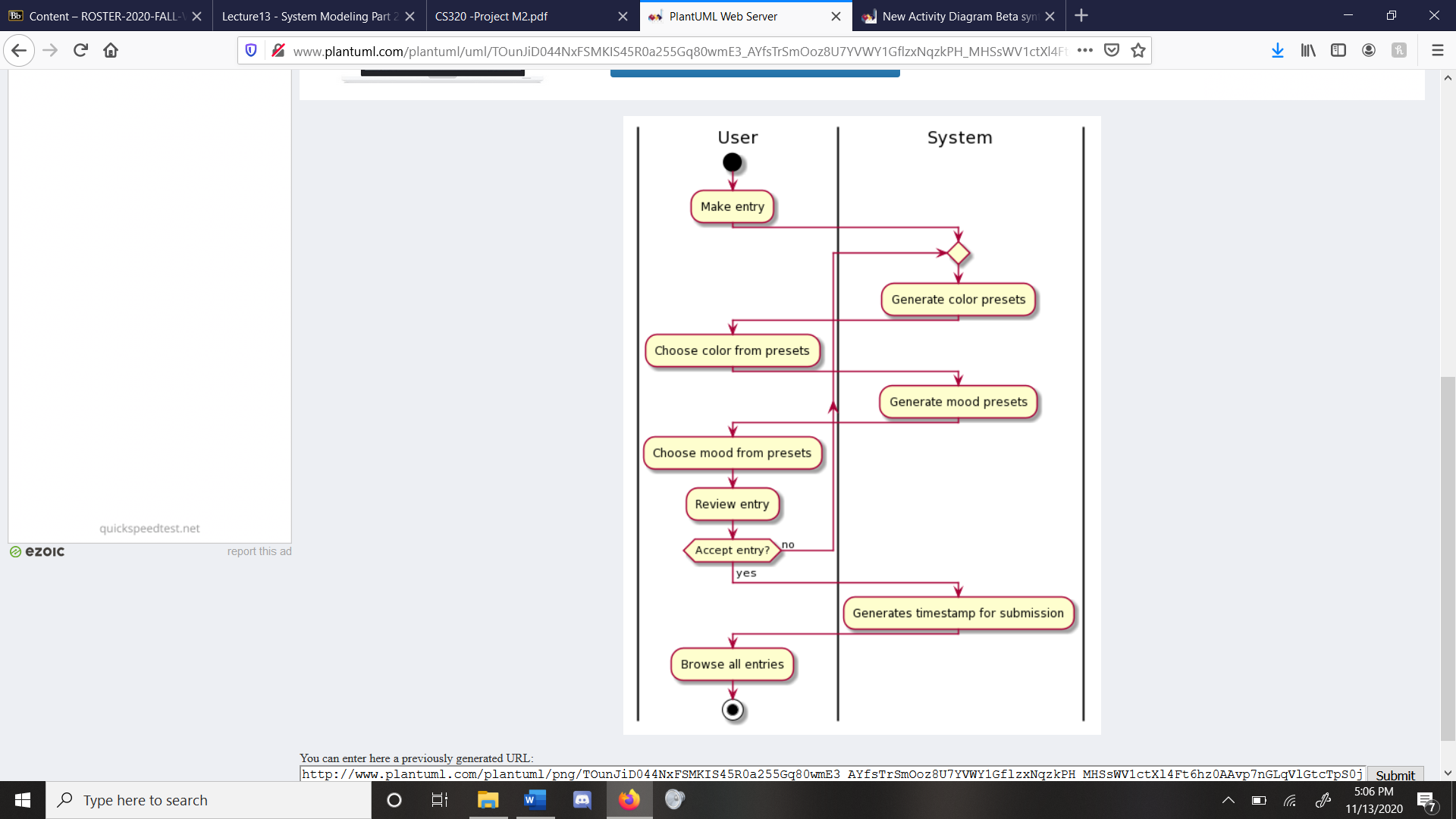


Figure 1: Making a Journal Entry

## D-2

<Same as for D-1, repeat for as many diagrams as you have for this section. Remove if not needed.

TO DO: Repeat same process as you did for D-1: Title, Diagram, Description, Traceability.>

## D-3

<Same as for D-1, repeat for as many diagrams as you have for this section. Remove if not needed.

TO DO: Repeat same process as you did for D-1: Title, Diagram, Description, Traceability.>

# Class Diagram(s)

## D-1

<Provide a title, the class diagram, and a table with the classes along with their descriptions.

TO DO: Provide a class Diagram for your program that shows the classes that exist for the system and their association, composition, and generalization. Provide a table underneath with a brief 1-2 sentence description of each class and their use within the system. Replace the title, D-1, with a more meaningful one.>

## D-2

*<Same as for D-1, repeat for as many diagrams as you have for this section. Remove if not needed.*

*TO DO: Repeat same process as you did for D-1: Title, Diagram, Classes and their Descriptions. >*

## D-3

*<Same as for D-1, repeat for as many diagrams as you have for this section. Remove if not needed.*

*TO DO: Repeat same process as you did for D-1: Title, Diagram, Classes and their Descriptions. >*

# Behavioral Diagram(s)

## 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.

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: EntryCusomization

|  |  |  |  |
| --- | --- | --- | --- |
| 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. |

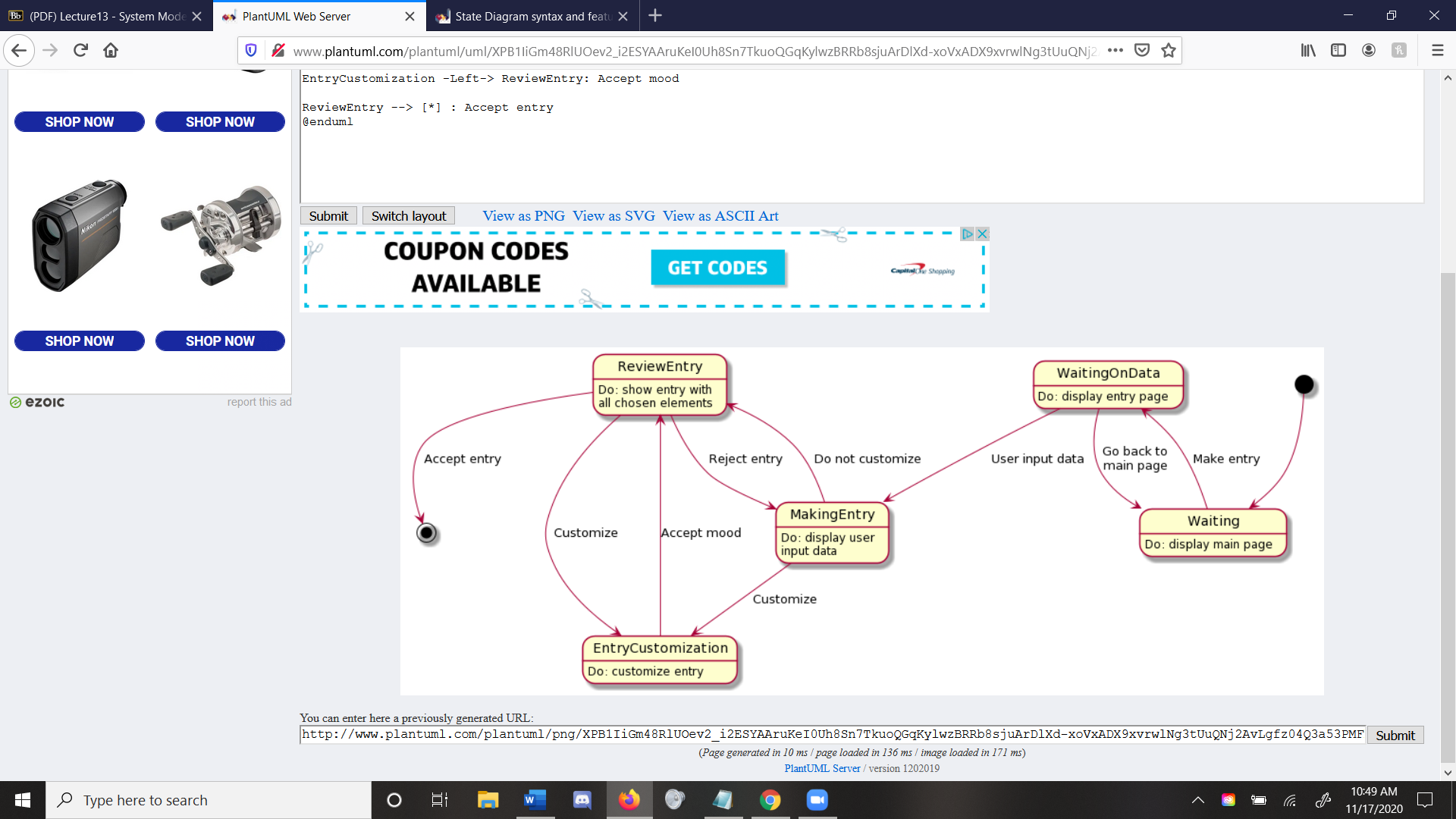


Figure 2: State Diagram for Making a Journal Entry

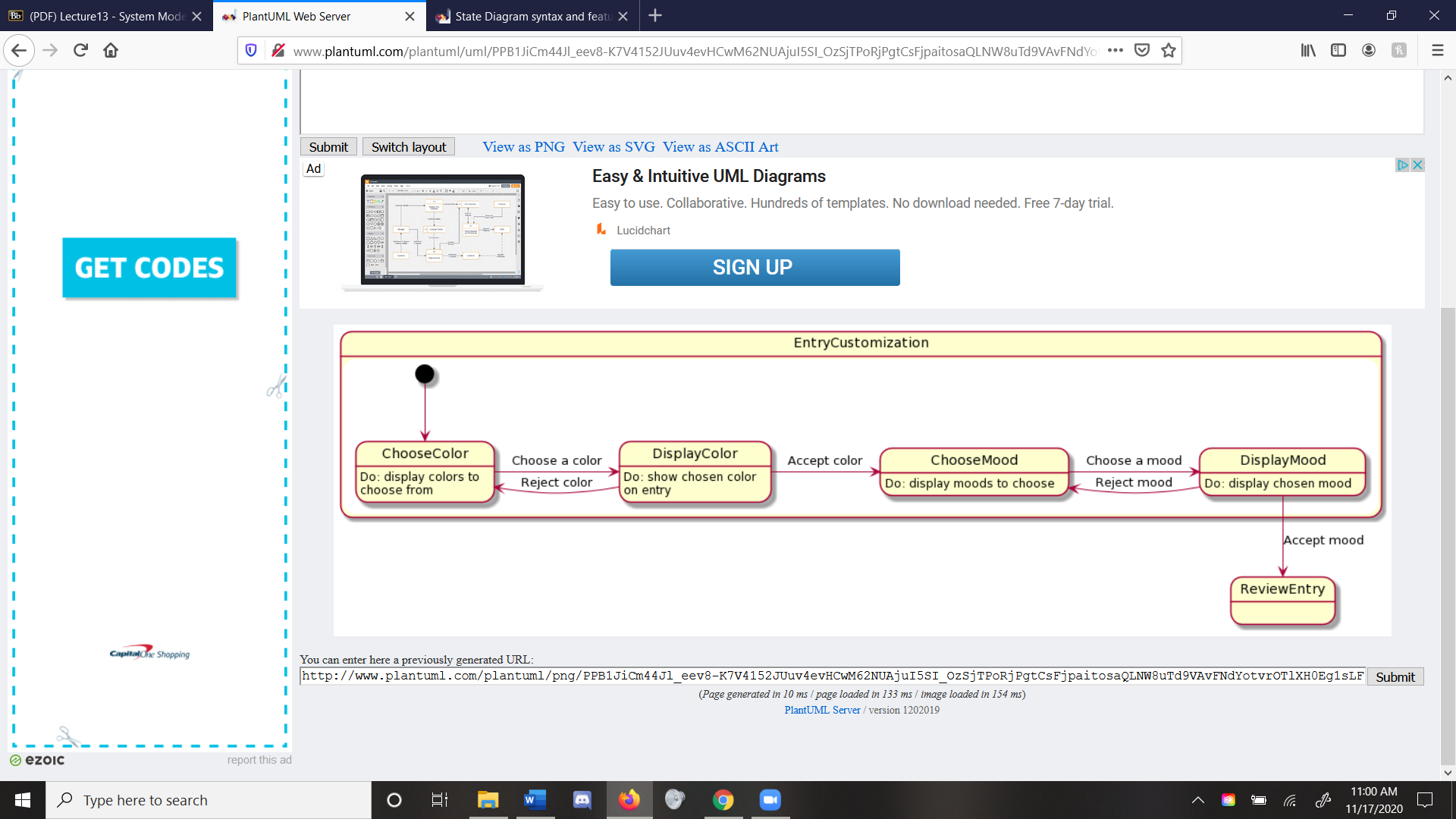


Figure 3:Entry Customization

## D-2

<Provide a title, the behavioral diagram, and a brief description about it.

TODO: Provide a Behavioral Diagram (sequence or state) and then give a description of what activity it is describing visually. For sequence diagrams, this would be each actor involved and the function calls between them. For state diagrams, this would be a table of each state with each state transition also labeled and described. A series of related complex states should be simplified as a “superstate” with a more in-depth view of it shown in a separate diagram. See the “operation” state in the microwave state diagram from Lecture 13 slides 27 and 28 for an example of this.>

## D-3

<Same as for D-1, repeat for as many diagrams as you have for this section. Remove if not needed.

TODO: Repeat same process as you did for D-1: Title, Diagram, Description.>

Appendix A - Group Log

< Please include here all the minutes from your group meetings, your group activities, and any other relevant information. This should contain and continue the contents from the SRS. This is optional for one-person projects.>

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