Software Requirements Specification

Shroomid.net

CPSC 462

Professor: Dr. Lidia Morrison

Spring 2020

Created by
Paul Smith
Haowen Yong
Ahmed Alweheiby
Joseph Santiago

1. Introduction

1.1. Purpose

The following document describes the software requirements for our 462 group project. The document explains the general purpose and reasoning behind our web application and includes the software requirements and the overall goals of the project.

1.2. Scope

Our goal is to help people identify mushrooms that they encounter and in particular assist in identifying potentially dangerous mushrooms. This application can have a variety of uses, such as helping to identify mushrooms in your area that could be dangerous to pets. Another use would be assisting identifying mushrooms to eat for foraging. A user is able to upload a picture of a mushroom to help determine what the mushroom is and if it is known to be poisonous.

1.3. Intended Audience

The intended audience of our web application would be adults, or children with the supervision of adults. The application is intended for individuals in need of assistance to help determine mushroom species.

2. Overall Description

2.1. User Objectives

Shroomid.net is a web application that focuses on providing fairly accurate mushroom species identification based upon an image.

2.2. Production Functions

Shroom ID is a web application that provides classification of mushroom species as well as informing the user if a species is known to be poisonous.

Below are the basic functions of Shroom ID:

- The user access the web app by going to shroomid.net
- The user can upload pictures of mushrooms to receive identifications

2.3. Operating Environment

Shroomid.net is a webapp that runs HTML, Javascript and CSS through the clients browser. The backend deployment of the webapp is run with multiple AWS services in a serverless environment.

2.4. Similar System Information

There is a Poisonous-Mushroom-App for IOS (Source: https://github.com/Sonmone/Poisonous-Mushroom-App) that is very similar to how our project works but the code that is on github does not appear to be on the market at this time. There are other applications on IOS and Android markets that make a similar id by showing mushrooms that are similar and letting the user decide if that is their mushroom. Most websites that identify mushrooms do so by selecting similar images.

2.5. User Characteristics

This web app is for users who are interested in identifying mushrooms. It should not be used by children who could eat dangerous mushrooms. Inorder to use this application the user needs to be able to use a modern web browser.

2.6. Design and Implementation Constraints

The web app currently runs user side Javascript files and needs a modern browser inorder to execute these files. At the time of writing this, the latest version of Google Chrome was used for both mobile and PC testing, but other browsers should work just as well. The web app is limited to jpg image files and will not take uploaded motion jpg files from modern phones.

2.7. Assumptions

This SRS assumes that:

- Users understands english
- User must have access to the internet to get to web app
- User must be using a modern browser
- User must be able to upload files

3. Functional Requirements

3.1. The web app will be accessible at shroomid.net

Description:

The user enters shroomid.net ino their address bar and is taken to the home page

Precondition:

User has access to internet and modern browser

Postcondition:

HTML and CSS files are downloaded and displayed

3.2. The web app will allow the user to upload a single photo for identification

Description:

The user uploads a photo of a mushroom they would like to have identified

Precondition:

The user clicks on the “Choose File” button. Then the user then follows local devices instructions to upload the photo. After the file is selected the user clicks the “Upload Photo and Run Code”

Postcondition:

The chosen photo is uploaded to the web app

3.3. The web app displays identification information

Description:

The web app displays the identification results of the users photo

Precondition:

The user must have successfully upload a photo to the webapp

Postcondition:

The web app will display the following results:

- *The top three predictions with confidence scores*
- *The name of the top prediction*
- *Whether the top prediction is poisonous*
- *The image that was submitted*

3.4. The web app will not allow users to submit an empty choose file data field

Description:

The web app will not allow the user to attempt to upload a file before one is chosen

Precondition:

The user clicks on the “Upload and Run Code” button without a file chosen

Postcondition:

The browser will display the error message: “Please choose a file to upload first.”

3.5. The web app only allows one submission per transaction

Description:

The web app will not allow the user to upload the same file multiple times in a single transaction.

Precondition:

The user clicks on the "Upload and Run Code" button after the a photo has been selected

Postcondition:

The web app turns off the functionality of the "Upload and Run Code" button until results are presented

3.6. The web app will not return results that are below minimum thresholds

Description:

The web app will not display results that are below the minimum confidence level. Default level is .30

Precondition:

The user clicks on the "Upload and Run Code" button after the a photo has been selected

Postcondition:

The web app displays a default failure image and displays: “Scores were too low”

3.7. The web app resets the page after each use

Description:

The web app will reset the page back to default, clearing the previous results, when the user uploads a new file

Precondition:

The user clicks on the “Choose File” button. Then the user then follows local devices instructions to upload the photo. After the file is selected the user clicks the “Upload Photo and Run Code”

Postcondition:

The default mushroom image is displayed and the previous results are cleared

4. Non-Functional requirements

4.1. Security

- No personal information is saved
- Photos uploaded are infrequently deleted but not used for any other purposes

4.2. Accuracy and Precision

- The default minimum to display results is .30

4.3. Availability

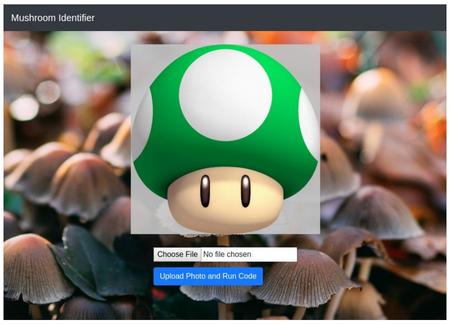
- The web app should meet or exceed an uptime of 99.99%

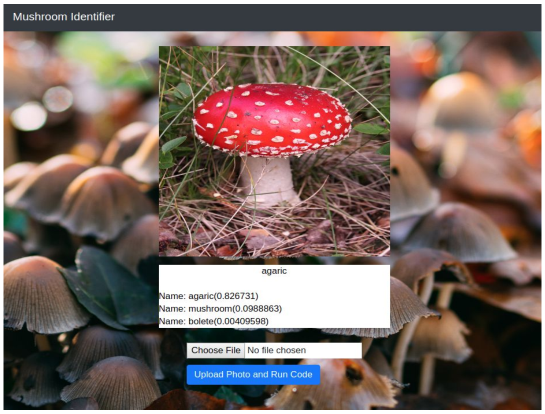
4.4. Version Control

- Any changes or upgrades to the functionality, aesthetics or performance to the webapp should be available to users in all areas by a maximum of 48 hours

5. Interface Requirement

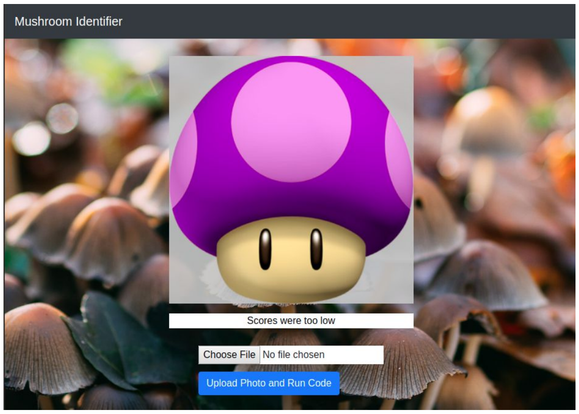
When the user enters the address shroomid.net they will be taken to the homepage of the webapp. After the photo has been submitted either an error img and results will display or the picture that was uploaded along with the results.





agaric

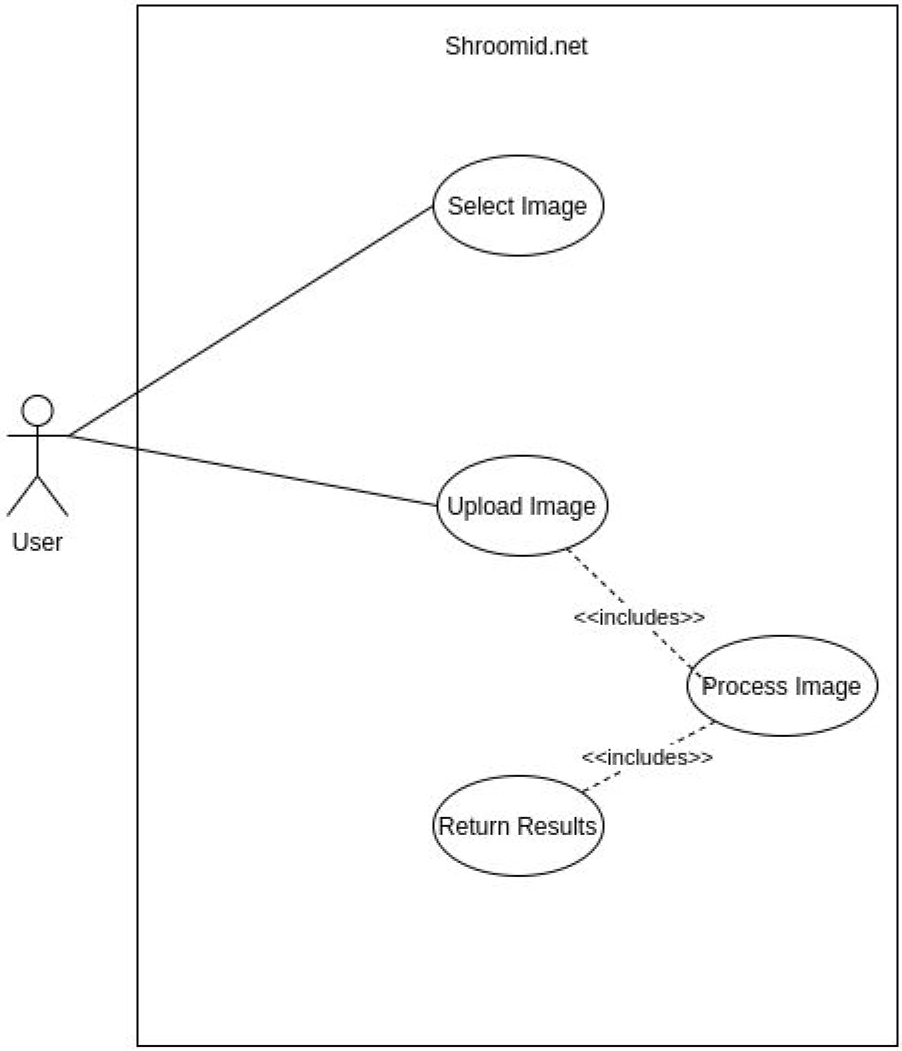
Name: agaric(0.826731)
Name: mushroom(0.0988863)
Name: bolete(0.00409598)



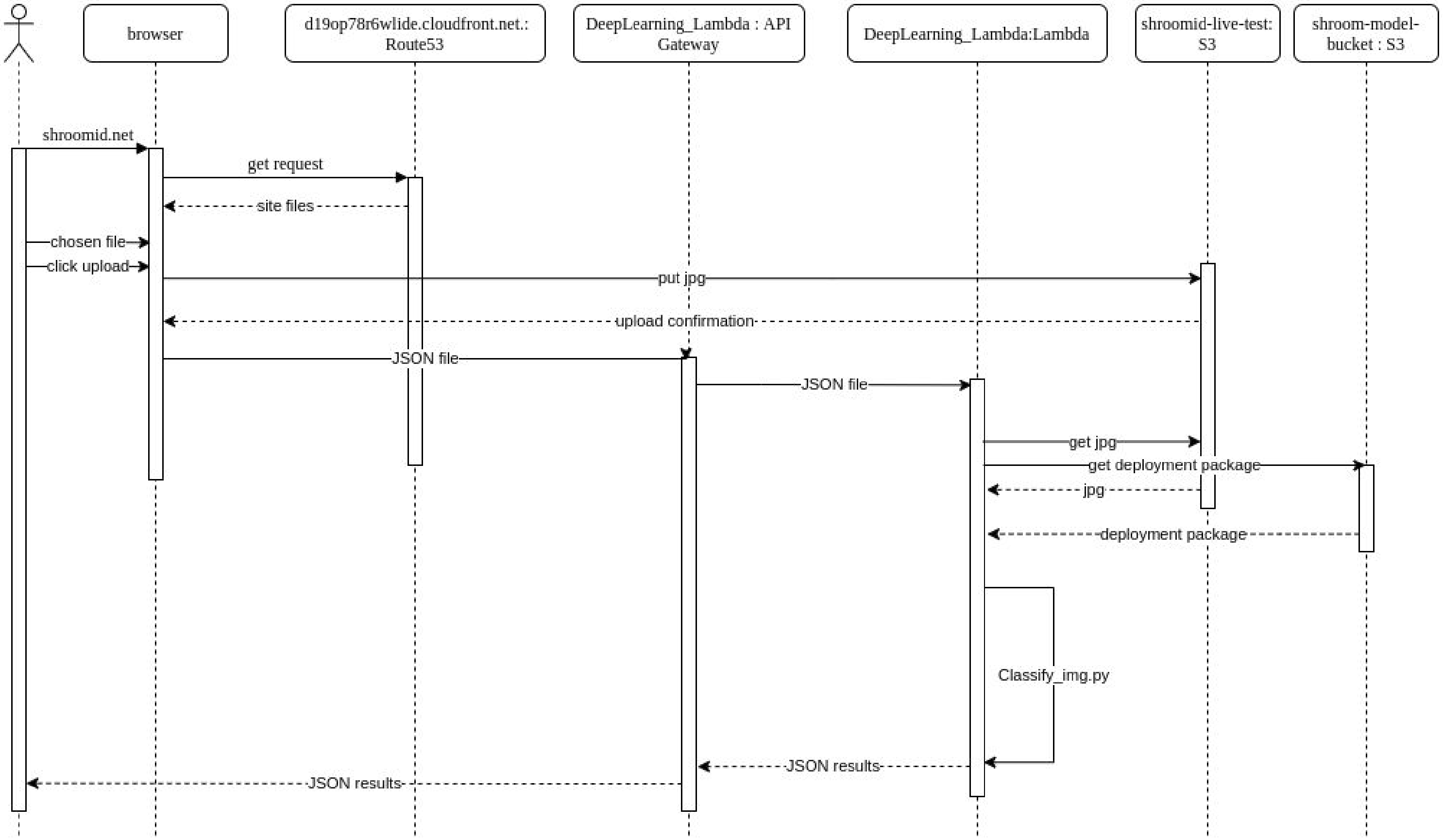
Scores were too low

6. UML Diagrams

6.1. Use Case Diagram



6.2. Sequence Diagram



7. Future Works

7.1. Finish Training Model

At the end of this project a completed model has not been finished

7.1.1. Model Integration

When a working model has been completed it needs to be integrated into our web app. Integration would involve creating a new deployment package, making any changes to the Classify_img.py file that the new .pb file needs and linking all of the new identifications to the lambda function that loads the tensorflow .pb file.

7.2. Poison Identification

At the end of this project there is no poison identification made by the web app. Currently the best identification that can be made is for the genus of a mushroom and not the species. When the web app can make species level identifications then the poison identification dictionary will be implemented to the Classify_img.py and return results page.

APPENDIX A: References and Tools

References

https://www.tensorflow.org/

https://github.com/google/inception

https://aws.amazon.com/blogs/machine-learning/how-to-deploy-deep-learning-models-with-aws-lambda-and-tensorflow/ (Deployment Package)

Tools

Google Colab

Amazon Web Services

Draw.io