MIAMI

CSC 431

Phantastic Fungi

Software Requirements Specification (SRS)

Team 06

JuanCarlos Jimenez	Scrum Master
Matthew Rossi	System Architect
Nolan McCarter	Requirements Engineer

Version History

Version	Date	Author(s)	Change Comments
1.0	2/20/22	JuanCarlos Jimenez Nolan McCarter Matthew Rossi	First draft

Table of Contents

Table of Tables

Table of Figures

1. System Requirements

1.1 Functional Requirements

1.1.1 Account Sign-up

Table 1: Account Sign-up

Title	Account Sign-up
Description	Form that allows users to create a new account.
Priority	Medium: 3
Precondition(s)	Open application after download and select "Create Account"
Basic Flow	User selects "Create Account" to sign-up for an account. Form opens with fields for name, username/email, and password. User submits information, the account is created and saved to a database with an encrypted password. User is directed to the login page.
Postconditions(s)	User account is created and stored. Upon creation the user is directed to the login page.
Use Case Diagram	3.1.1

1.1.2 Account Login

Table 2: Account Login

Title	Account Login
Description	Form that allows the user to login to an existing account
Priority	Medium: 3
Precondition(s)	Open application and select Login or automatically directed after creating account. User must have previously created an account.
Basic Flow	User enters username/email. User enters password. User presses "Sign in" and account validation occurs.
Postconditions(s)	If account validation is successful, brings user to home page. If the account doesn't exist, prompts the user to re-enter login credentials, create an account, or recover password.
Title	Account Login
Use Case Diagram	3.1.1

1.1.3 Continue as Guest

Table 3: Continue as Guest

Title	Continue as Guest
Description	Allows user to continue as guest with more limited functionality.

Priority	Medium: 3
Precondition(s)	Open application.
Basic Flow	User selects “Continue as Guest”.
Postconditions(s)	Brings users to a version of the home page without features that account owners would have. Has access to “Take Picture”, “Upload Picture”, and “Random Mushroom” features.
Use Case Diagram	3.1.1

1.1.4 Account Settings

Table 4: Account Settings

Title	Account Settings
Description	Allows user to modify settings
Priority	Medium: 3
Precondition(s)	User must be logged into their account
Basic Flow	A verified user clicks on “Account Settings” from the home page and is brought to a new page where they can change their account information or change settings, like whether to share location data, or to share the images they upload to help improve the classifier. Users can also logout of their account. Users can save or discard the changes they made to their settings.
Postconditions(s)	Returns to the home page after changes are saved or discarded. Returns to opening screen if user selects “logout”.
Use Case Diagram	3.1.1

1.1.5 Take Picture

Table 5: Take Picture

Title	Take Picture
Description	Allows user to take a picture from their camera
Priority	Highest: 1
Precondition(s)	User must have logged in or selected “Continue as Guest”. Must enable access to the camera.
Basic Flow	User clicks “Take Picture”, which opens camera. After taking picture, asks user to confirm that they want to submit this picture for classification.
Postconditions(s)	Runs image through “Verify Mushroom Presence”.
Use Case Diagram	3.1.1

1.1.6 Upload Picture

Table 6: Upload Picture

Title	Upload Picture
Description	User can upload a picture from camera roll to be classified.
Priority	Mandatory: 0
Precondition(s)	User must have logged in or selected “Continue as Guest”.

	User must allow access to camera roll.
Basic Flow	User clicks “Upload Picture”. User selects a picture from camera roll. User confirms that they want to submit this picture for classification.
Postconditions(s)	Runs image through “Verify Mushroom Presence”.
Use Case Diagram	3.1.1

1.1.7 Verify Mushroom Presence

Table 7: Verify Mushroom Presence

Title	Verify Mushroom Presence
Description	Uses binary neural network classifier trained on mushroom image data to determine whether uploaded picture contains a mushroom.
Priority	Low: 4
Precondition(s)	User must take or upload a picture.
Basic Flow	App tells user whether the picture is predicted to contain a mushroom or not. If it isn’t a mushroom it will suggest the user takes/uploads another picture, with an option to override and proceed with classification anyway.
Postconditions(s)	If picture is indeed a mushroom (or user overrides), will move to “Classify Mushroom”. Does not save images to the database for cleanliness and privacy reasons. Otherwise, returns to home page.
Use Case Diagram	3.1.1

1.1.8 Classify Mushroom

Table 8: Classify Mushroom

Title	Classify Mushroom
Description	Uses multiclass neural network classifier trained on mushroom image data to determine the genus and species of the mushroom in the picture.
Priority	Mandatory: 0
Precondition(s)	User must take/upload an image that passes the mushroom verification (or user overrode verification).
Basic Flow	App tells user the predicted classification for the mushroom in the image, including genus and species.
Postconditions(s)	Brings user to page detailing the genus and species of the mushroom. Has options to return to home page, view their uploads if they are logged into an account, or view additional information.
Use Case Diagram	3.1.1

1.1.9 View My Uploads

Table 9: View My Uploads

Title	View My Uploads
Description	Allows user to view all previous uploads
Priority	Medium: 3
Precondition(s)	User must be logged into an existing account and have previously submitted at least one mushroom
Basic Flow	User clicks on my account. Selects My Uploads. Displays all previous images they uploaded along with the genus and species classification.
Postconditions(s)	Allows users to delete any uploads, or go back to the classification of a previous upload. Has option to return to home page.
Use Case Diagram	3.1.1

1.1.10 Show Additional Information

Table 10: Show Additional Information

Title	Show Additional Information
Description	Allows user to visit Wikipedia page on the genus and species of mushroom they submitted.
Priority	Medium: 3
Precondition(s)	User must have taken/uploaded a picture of a mushroom that passes verification and receives a classification or must be viewing a submission from "My Uploads".
Basic Flow	User selects "Extra Information". Brings user to a Wikipedia page on the genus and species of that mushroom. Shows users other pictures of that mushroom from mushroomobserver.org.
Postconditions(s)	User can select to return to home page.
Use Case Diagram	3.1.1

1.1.11 Update Model (Developer-only)

Table 11: Update Model (Developer-only)

Title	Update Model (Developer-only)
Description	Allows developer/maintainer to easily upload new parameters for the neural networks and deploy without significant downtime.
Priority	Medium: 3
Precondition(s)	Developer must be logged in and have trained a neural network that has more desirable performance than those currently deployed.
Basic Flow	Developer selects "My Account" then selects "Developer View", at which they can upload a file containing the new model weights (or revert to a previous model). The app quickly tests a small suite of curated images to confirm that the new weights work. If they pass the test, the dev can confirm replacement.
Postconditions(s)	The app will now use the new model for all users.

	The old weights are automatically backed up to the database. The last 3 models are saved by default. Developer is then returned to homepage.
Use Case Diagram	3.1.1

1.2 Non-Functional Requirements

1.2.1 Password Encryption

Table 12: Password Encryption

Title	Password Encryption
Description	When a user creates an account it will save the corresponding login information as an encrypted password to be stored in a database.
Priority	High: 2
Applicable FR(s)	1.1.1

1.2.2 Automated Password Recovery Emails

Table 13: Automated Password Recovery Emails

Title	Automated Password Recovery Emails
Description	Automatically send emails for password resets.
Priority	High: 2
Applicable FR(s)	1.1.2

1.2.3 Uptime

Table 14: Uptime

Title	Uptime
Description	The application must be down for no longer than 15 minutes each month.
Priority	Medium: 3
Applicable FR(s)	N/A

1.2.4 Inference Time

Table 15: Inference Time

Title	Inference Time
Description	Model inference for verification and classification must be performed in less than 5 seconds.
Priority	High: 2
Applicable FR(s)	1.1.7, 1.1.8

1.2.5 Model Replacement Downtime

Table 16: Model Replacement Downtime

Title	Model Replacement Downtime
Description	When a developer updates the model parameters, the app should experience no more than 5 minutes of downtime.
Priority	Medium: 3
Applicable FR(s)	1.1.11

2. System Constraints

2.1 Tool Constraints

2.1.1 Source Control Constraint

Table 17: Source Control Constraint

Title	Source Control Constraint
Description	For class, we are required to use GitHub for source control.
Priority	Mandatory: 0

2.1.2 Mobile Application Framework

Table 18: Mobile Application Framework

Title	Mobile Application Framework
Description	We will be using Flutter from Google to design our application. Flutter allows for cross-platform deployment from a single codebase. Reference: docs.flutter.dev/
Priority	Mandatory: 0

2.1.3 Web Framework Constraint

Table 19: Web Framework Constraint

Title	Web Framework Constraint
Description	We will be using Django, because it is Python-based, to develop a web framework to host the neural networks and perform inference with them. With Django, we can build RESTful APIs for the frontend to call. Reference: docs.djangoproject.com/en/4.0/
Priority	Mandatory: 0

2.1.4 Web Server Constraint

Table 20: Web Server Constraint

Title	Web Server Constraint
Description	Because we cannot store neural networks in app on mobile devices, we need to host them on a server. We will run the Django web framework on an ASGI Daphne web server, which we'll deploy with AWS Lightsail. Reference: docs.djangoproject.com/en/3.2/howto/deployment/asgi/daphne/
Priority	Mandatory: 0

2.2 Language Constraints

2.2.1 Frontend User Interface

Table 21: Frontend User Interface

Title	Frontend User Interface
Description	As we are using Flutter, the app itself will be written in Dart.
Priority	Mandatory: 0

2.2.2 Backend Framework

Table 22: Backend Framework

Title	Backend Framework
Description	As we are using Django, the backend will be written in Python.
Priority	Mandatory: 0

2.2.3 Neural Networks

Table 23: Neural Networks

Title	Neural Networks
Description	The two neural networks are written in and trained in Python. Reference: github.com/google-research/big_transfer They will be accessed by the app for inference via API calls.
Priority	Mandatory: 0

2.3 Platform Constraints

2.3.1 Application Platform

Table 24: Application Platform

Title	Application Platform
Description	Flutter is platform independent, although the application is intended for use as a mobile app.
Priority	Lowest: 5

2.4 Storage Constraints

2.4.1 Neural Network Parameters Storage

Table 25: Neural Network Parameters Storage

Title	Storage Constraints
Description	Set by AWS Lightsail. We will need to choose a sufficiently lightweight neural network to be able to store its parameters on the web server.
Priority	Mandatory: 0

2.4.2 User Account Information Storage

Table 26: User Account Information Storage

Title	User Account Information Storage
Description	Set by AWS Lightsail. User account data includes: Username, Password (encrypted), Email, Uploaded images with classification, (With location data, if provided), Settings (Share location on/off, Share images to public database). Such storage is necessary to have a verified user system, but not essential to the core functionality of the app.
Priority	Medium: 3

2.5 Computation Constraints

2.5.1 Inference Computation Constraint

Table 27: Inference Computation Constraint

Title	Inference Computation Constraint
Description	Set by AWS Lightsail. Will need enough computational power to run inference with the neural networks.
Priority	Mandatory: 0

2.6 Network Constraints

2.6.1 Internet Access

Table 28: Internet Access

Title	Internet Access
Description	Users will need internet access to use the app’s features.
Priority	Mandatory: 0

2.6.2 Access Database

Table 29: Access Database

Title	Access Database
Description	Client application needs to access the database to retrieve user account information.
Priority	High: 2

2.6.3 Send Image for Inference

Table 30: Send Image for Inference

Title	Send Image for Inference
Description	Uploaded image needs to be sent to server for inference and the result needs to be returned to client.
Priority	Mandatory: 0

2.7 Deployment Constraints

2.7.1 AWS Lightsail Deployment

Table 31: AWS Lightsail Deployment

Title	AWS Lightsail Deployment
Description	Our mobile application will be deployed on a web server provided by AWS Lightsail, which has a 3 month free trial.
Priority	Medium: 3

2.8 Transition & Support Constraints

2.8.1 Neural Network Maintenance Constraint

Table 32: Neural Network Maintenance Constraint

Title	Neural Network Maintenance Constraint
Description	If either of the neural networks exhibit undesirable classification behavior, the developers need to be able to

	easily upload a new set of parameters without significant app downtime.
Priority	High: 2

2.8.2 End of Life

Table 33: End of Life

Title	End of Life
Description	The developers of this project (unless they fail this class) will cease working on the project at the end of the semester, at which point the project must be retired unless a new team can take over development.
Priority	Lowest: 5

2.9 Budget & Schedule Constraints

2.9.1 Budget Constraint

Table 34: Budget Constraint

Title	Budget Constraint
Description	As there is no funding for this project, there is no budget. We cannot use anything that costs money.
Priority	Lowest: 5

2.9.2 Semester End Constraint

Table 35: Semester End Constraint

Title	Semester End Constraint
Description	Because this project is for class, it must be completed by the end of the Spring 2022 semester.
Priority	Mandatory: 0

2.9.3 Free Trial Constraint

Table 36: Free Trial Constraint

Title	Free Trial Constraint
Description	AWS Lightsail provides only a 3-month trial. After that time, the application will have to be retired or a source of funding to maintain a web server must be secured.
Priority	Lowest: 5

3. Requirements Modeling

3.1.1 System Use-Case Diagram

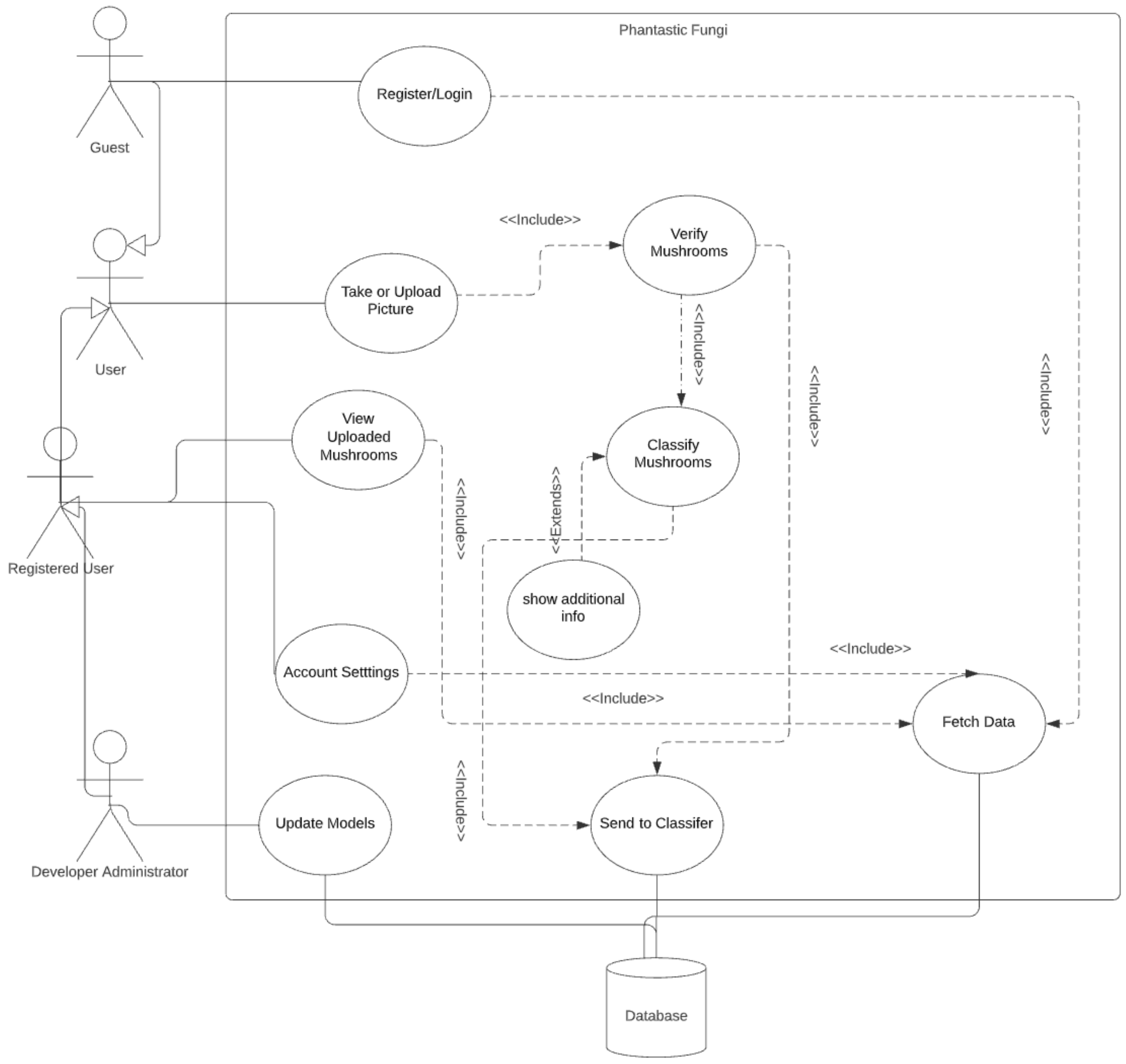

Figure 1: System Use-Case Diagram

4. Evolutionary Requirements

4.1 Functional Requirements

4.1.1 Random Mushroom

Table 37: Random Mushroom

Title	Random Mushroom
Description	Selects a random mushroom from database and shows it to user.
Priority	Low: 4
Precondition(s)	User must have already selected to continue as guest or login to account.
Basic Flow	User selects to view a random mushroom. Directs the user to a random page on Mushroom Observer.
Use Case Diagram	N/A

4.2 Non-Functional Requirements

4.2.1 Location Data

Table 38: Location Data

Title	Location Data
Description	Users can opt into sharing location data to add to classifications, either automatically or manually. Such data can improve classification accuracy and contribute to the database.
Priority	Medium: 3
Applicable FR(s)	1.1.4, 1.1.8