Software Requirements Specification

for

Eye in the Sky

Version 0.1

Prepared by

Group Name: String cheese;

Jason Worden	5340385	worden.jason.p@gmail.com
Alexander Huitric	4982351	ahuitric@gmail.com
Andrew Hascall	4915021	drewaugust@gmail.com
EJ Fernandes	5114061	ejectbutton@gmail.com

Instructor:	Chandra Krintz
Course:	CS 189A-B
Lab Section:	Thursday 6:00 PM
Teaching Assistant:	Geoffrey Douglas
Date:	2/11/2014

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Jason Worden Alex Huitric EJ Frenandes Andrew Hascall	Creation of file and first documentation of requirements for project Eye in the Sky	2/11/2014
1.1	Alex Huitric	Updated sections the described used hardware, updated and added user stories, updated use case diagram	3/3/2014

1 Introduction

1.1 Document Purpose

This software requirements specification document for project Eye in the Sky is intended to provide a detailed explanation of the features and design specifications for this application. Additionally, it will serve as a tool to communicate the design plans of Team String Cheese with PowWow Energy so that both parties may fully understand the specifications, requirements, and design of the Eye in the Sky project.

1.2 Product Scope

Water usage and conservation is important to farmers' profits and to the general well-being of our planet. Eye in the Sky will help farmers take better care of their crops and help better manage their water supply. Our project will provide farmers with the tools needed to find water leaks, and to detect stress levels of crops to prevent them from dying - all in a manner much more efficient than can be done today.

A small unmanned aircraft -- attached with an infrared camera, optical camera, and a GPS system -- with will be flown over a farm. This collected imagery will be used to collect anomalies of cool spots that may signify water leaks as well as anomalies that may signify water stress in crops. Farmers will be able to access this data via a secure web application.

1.3 Intended Audience and Document Overview

This document is intended for:

- the faculty and teaching assistants for CS189A/B;
- our mentors Jim Klingshern and Olivier Jerphagnon;
- our team;
- and anyone else interested in learning more about our project.

The remainder of this document provides the general product description and a technical outline for the requirements of this system. Section 2 will give a high level description of the implementation. Section 3 will give a more detailed description of the specific requirements and the low-level implementation for the different components of the application, which include various hardware and software interfaces and functional requirements.

1.4 Definitions, Acronyms and Abbreviations

Item	Definition	Version used (if applicable)
API	Application Programming Interface; a protocol used as an interface so that software components can communicate with each other	
Python	a high-level programming language	2.7.5
Django	a high-level web application framework on top of the Python language	1.6.1
OpenCV	an open source, cross-platform computer vision and image processing library	2.4.8
UAV	unmanned aerial vehicle (a.k.a. drone)	
APM	type of autopilot UAV provided by 3D Robotics	
IR	infrared; part of the electromagnetic spectrum with wavelengths between 0.8 micrometers and 1 millimeter that is used for thermal imaging	
GPS	Global Positioning System; used to acquire geolocation data	
CWSI	crop water stress index; measure of the relative transpiration rate occurring from a plant at the time of measurement - using a measure of plant temperature and the vapor pressure deficit, which is a measurement of the dryness of the air	
GUI	graphical user interface	
AWS	Amazon Web Services	
HTML	HyperText Markup Language; the standard markup language for creating web pages and other information that can be displayed in a web browser (latest standard is HTML5)	
CSS	style sheet language used for describing the look and formatting of HTML documents (latest standard is CSS3)	
jQuery	cross-platform JavaScript library that simplifies the client-side scripting	2.1.0
Twitter Bootstrap	front-end framework with responsive CSS and Javascript components	3.1.0
NIR	Near-InfraRed (electromagnetic radiation with a wavelength of 700-1400nm)	

1.5 Document Conventions

This document is singe-spaced and has 1” margins. The body is 12-point Arial font. Major headers are green, 24-point Trebuchet MS font and center-aligned. Subheaders use bolded 24-point Arial font. Smaller subheaders are bolded 12-point Arial font.

1.6 References and Acknowledgments

- Django
  - https://www.djangoproject.com/
- OpenCV
  - http://opencv.org/
- jQuery
  - http://jquery.com/
- 3D Robotics, Inc.
  - http://3drobotics.com/
- FLIR Systems
  - http://www.flir.com/US/
- MaxMax
  - https://www.maxmax.com/

2 Overall Description

2.1 Product Perspective

Eye in the Sky incorporates both existing products and new components.

It includes a UAV with on-board thermal and optical cameras as well as a GPS device. The UAV is an X8 series multicopter made by 3D Robotics, Inc. that comes with an existing API and GUI to automate flight routes. The cameras are Canon ELPH 110 Powershots, one of which is in a stock configuration, and the other which has had an 830nm near-infrared filter installed by MaxMax. After the UAV collects imagery during a flight, images are uploaded to the AWS server. On this server, image processing occurs in C++ with the OpenCV library.

The relevant images and results of the image processing are then accessible via a web application. We will build on top of Django using Python, and run this application from the AWS server using a MySQL database. Farmers will be the users of the web application in order to review the data about their own farm's conditions.

2.2 Product Functionality

- Users must be able to sign in securely to the web application.
- Users will be able to select a flight route to cover their farm.
- The product must be able to cover a farm in one flight and collect geolocated thermal and optical imagery.
- The product must be able to process this imagery and geolocate possible water leaks or areas of water-stressed crops.
- Users must be able to see where anomalies have been found that represent water leaks or water-stressed crops.

2.3 Users and Characteristics

The users of our web application are owners or managers of small farms (20 acres or less) who are familiar with using a modern web browser on a desktop or laptop.

2.4 Operating Environment

Eye in the Sky's web application will be running on an AWS server with Django, Python 2.7, Apache, and MySQL.

2.5 Design and Implementation Constraints

Design:

- Back-end languages
  - C++ with OpenCV (2.4.8)
  - Python (2.7.5) with Django (1.6.1)
- Front-end languages
  - Javascript
    - with jQuery (2.1.0)
  - HTML5
  - CSS3
    - using Twitter Bootstrap (3.1.0) front-end framework
- Database: MySQL
- Server/infrastructure: AWS

Constraints:

- Must run on Google Chrome 32 and Mozilla Firefox 26
- User must authenticate him or herself at login

2.6 User Documentation

To set up the flight plan above a farm, the farmer must familiarize themselves with the APM Flight Planner 2.0 software for their own UAV.

- Introduction to the software
  - http://copter.ardupilot.com/wiki/introduction/
- First-time setup
  - http://copter.ardupilot.com/wiki/initial-setup/

The web interface will be very user-friendly and not require an in-depth manual.

2.7 Assumptions and Dependencies

Since UAVs are not meant to carry heavy objects, we as assuming the camera size will be very small (negligible weight). The camera will take video of the ground as the UAV flies over the designated path, hence we are assuming there to be an interface between the camera and memory (direct memory access).

3 Specific Requirements

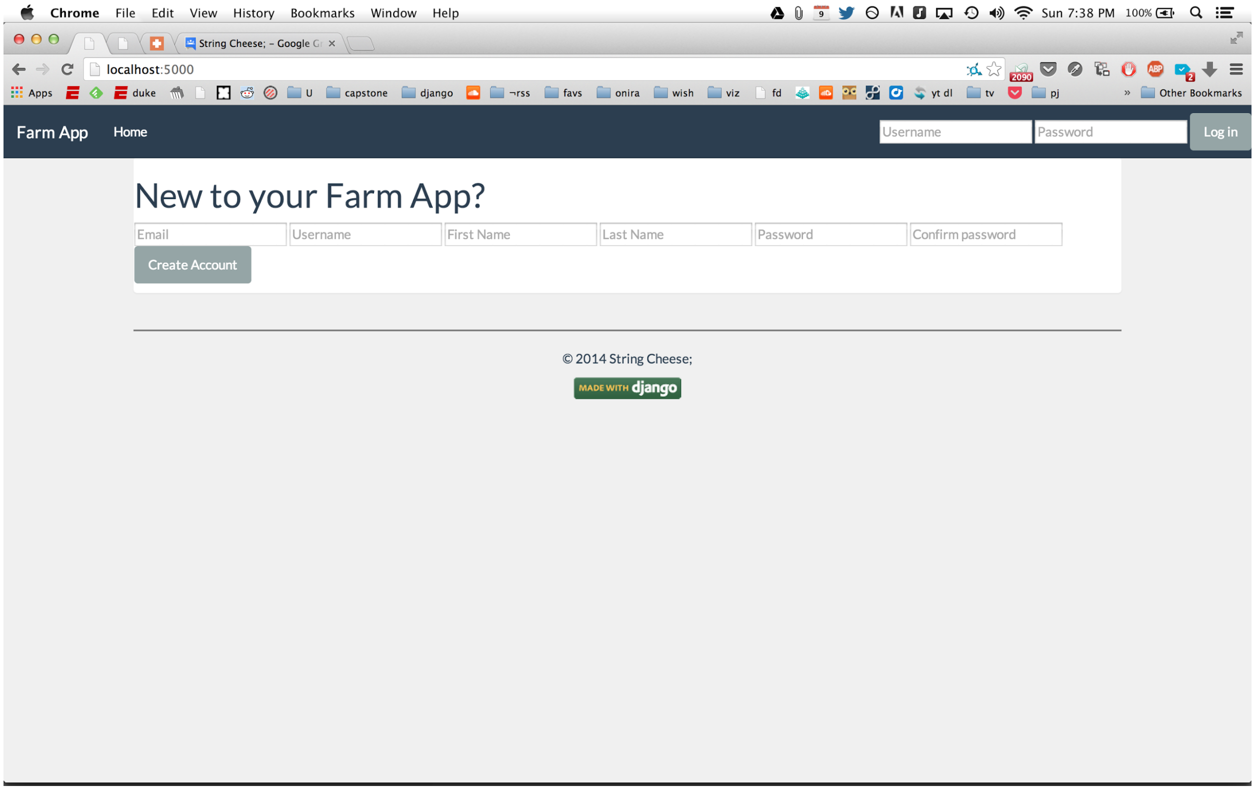
3.1 External Interface Requirements

3.1.1 User Interfaces

The GUI will consist of browser interfaces which will allow users to register for an account, log in, set flight data, data about a given field, upload images captured from the UAV, and view the results of processing those images for each session.

Interfaces for each of the functions will appear as follows:

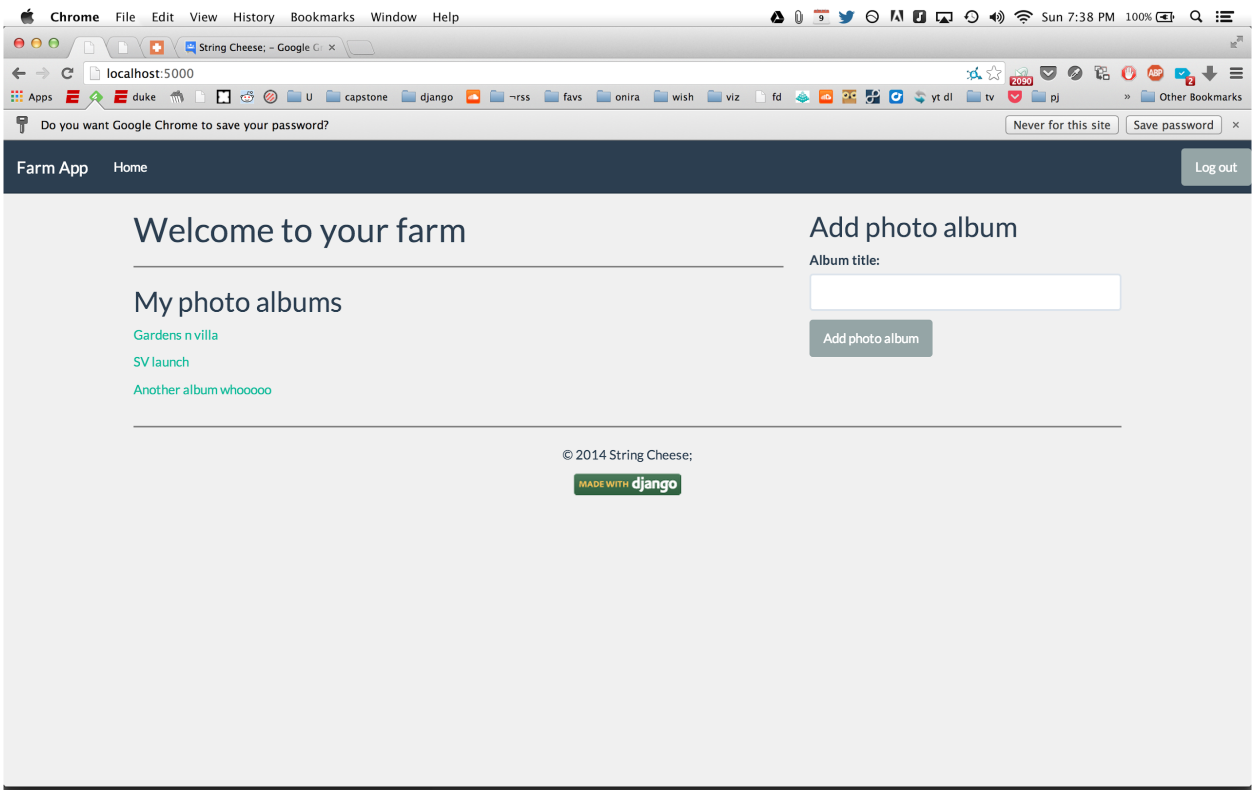
User log-in/sign-up page:



New to your Farm App?

© 2014 String Cheese;

User dashboard:



Chrome File Edit View History Bookmarks Window Help
Sun 7:38 PM 100%
String Cheese; - Google G
localhost:5000
Apps duke U capstone django ¬rss favs onira wish viz fd yt dl tv pj Other Bookmarks
Never for this site
Save password
Farm App
Home
Log out
Add photo album
MADE WITH django

Do you want Google Chrome to save your password?

Welcome to your farm

My photo albums

Gardens n villa

SV launch

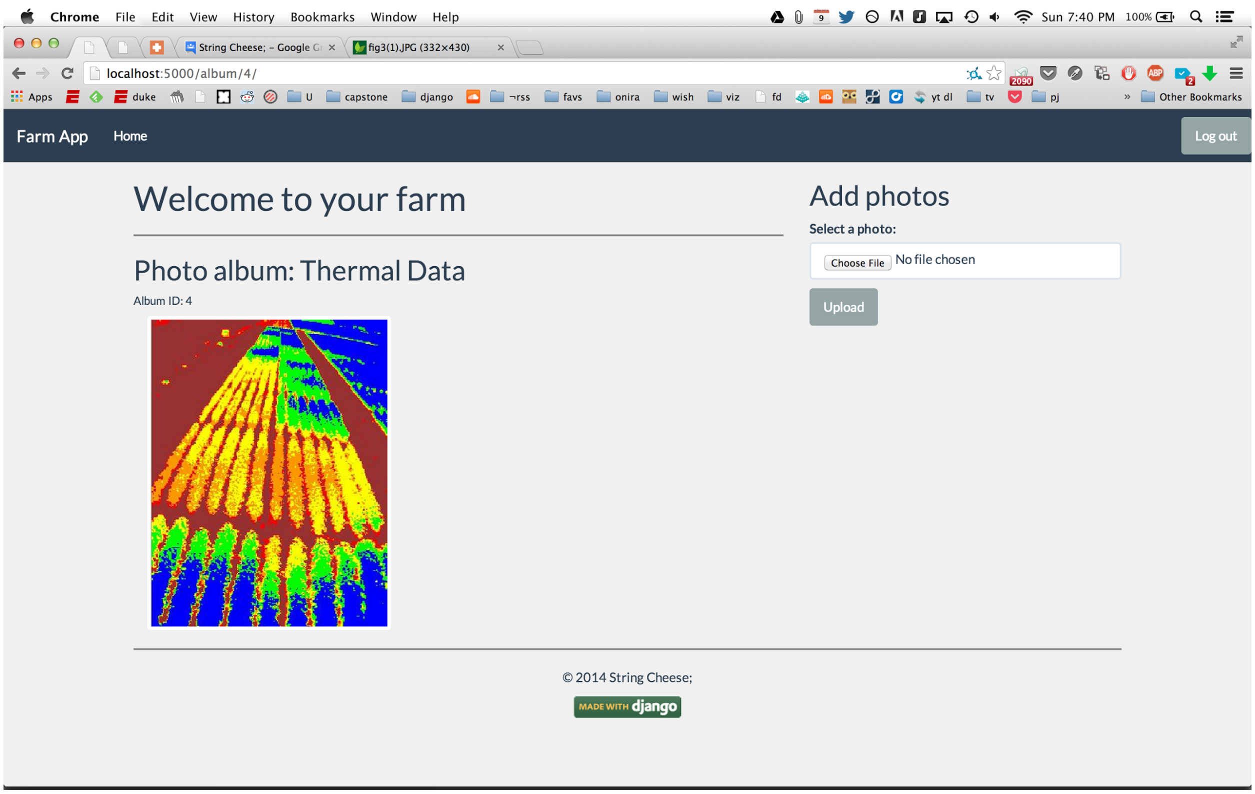
Another album whooooo

Add photo album

Album title:

© 2014 String Cheese;

Uploading session images & viewing results:



Welcome to your farm

Photo album: Thermal Data

Album ID: 4

Add photos

Select a photo:

No file chosen

© 2014 String Cheese;

MADE WITH django

3.1.2 Hardware Interfaces

Interface between remote users and the web server

The remote users will be connected to the web server through the internet, using whatever local infrastructure (such as Ethernet or IEEE 182.11 wireless or a direct connection) is used to interface their device to the wide internet, and will use whatever infrastructure is utilized by their respective ISPs and backbone providers to transmit data.

Interface between the image processing server and the web server

The web server and image processing server will both run on the same physical server, utilizing the local loop-back connection on ports 12345 and 12344 to send command and status signals to each other. They will share a hard-drive array with a common file system.

Interface between UAV and computer systems

The UAV will have flight data uploaded via USB 2.0, and flight logs will be retrieved via USB 2.0. Captured images will be written to a Secure Digital card, which will then be plugged into a computer and uploaded to the Web Server.

3.1.3 Software Interfaces

User & web server interface

The web server will interface with the browsers Firefox 26 and Chrome 32 running on the host operating system of the user's device of choice. The web server itself shall run on a Linux server with an installation of Python 2.7 and Django.

CV server interface

The Open CV server will run on a Linux server with an installation of Open CV and Python 2.7.

3.1.4 Communications Interfaces

Web services

The web services accessed by the user will transmit data using HTTP for non-encrypted pages, and HTTPS for any encrypted data (such as user login and the display of image processing results). The web services themselves will be presented in HTML web pages.

Captured images will be uploaded to the web server via the use of FTP or SFTP. JPEG formatted images will be used as the accepted image type as well as the output image type. The output JPEG images will be of 90-100% quality to avoid levels of image degradation which would interfere with the display of the results.

Image Processing Server to Web Server interface

The two servers will communicate with a simple signalling protocol over local ethernet using TCP on ports 12344 and 12345. The two servers will both have access to either a shared harddrive bank or NAS device, allowing them to share a file system and files, rather than to directly transmit images back and forth, which would use up memory on both systems.

3.2 Functional Requirements

User Registration

Feature	Description	Priority
Display registration page	Displays necessary page to allow new user registration	Required
Submit registration	Allow registration form to post data to the server to complete registration	Required
Process registration	Allow user to be entered into user tables and set the	Required

	account to active	

User Login

Feature	Description	Priority
Submit login form	Allow user to send their encrypted login credentials	Required
Authenticate login data	Allow for authentication of the login credentials against the userdata stored in the database.	Required
Ignore invalid login	Display an invalid username or password error if a user attempts to submit incorrect credentials. System should not authenticate this user.	Required

User Homepage

Feature	Description	Priority
Allow logging out	Link to allow the user to log out of their current login-session.	Required
Create new session	Allow user to create a new image analysis session. Link to the session creation system.	Required
Manage & view sessions	Allow user to go to the session management page to upload images and view processed results	Required

User Create Session

Feature	Description	Priority
Set flight area	Allow user to specify a field region by inputting the geographical data	Required
Report area oversized	Report area is being too large if it is larger than the flight range of the UAV. Suggest creating sub-sessions	
Set field type	Allow user to specify field type (by plant type)	Non-critical
Set processing criteria	Allow user to select which processing features they wish to use	Non-critical
Create sub-sessions	Provide option to allow user to create additional sub-sessions which share the same properties except flight area. This is meant for use with large fields.	Non-critical
Check Parameters	Require that the user has provided the necessary parameters to create the session, otherwise reject and display an error	Required
Save session data	Save the data from the session so it may be managed at a later time	Required

User Manage Session

Feature	Description	Priority
Upload images	Allow user to upload captured images	Required
Start processing	Allow user to initiate the image processing	Required

View images	Allow user to view the uploaded images	Required

User Upload Images

Feature	Description	Priority
Transmit images to server	Provide mechanism to transmit the images from the user's computer to the server.	Required
Save images	Save the images to an appropriate directory on the shared drives/NAS	Required

Server-Server communication

Feature	Description	Priority
New Job request	Send a request to create a new image processing job, transmitted in header	Required
Job Status request	Send a request to get the status of a pending job, transmitted in header	Required
Job Status reply	Send a reply to report on the status of a pending job, transmitted in header	Required
Job Complete reply	Send a reply to report that a job is complete, transmitted in header	Required
Job ID	Job ID must be sent for each transaction, as a transaction ID. Consists of fixed length Username hash and Session ID, which designate the appropriate directory on NAS	Required

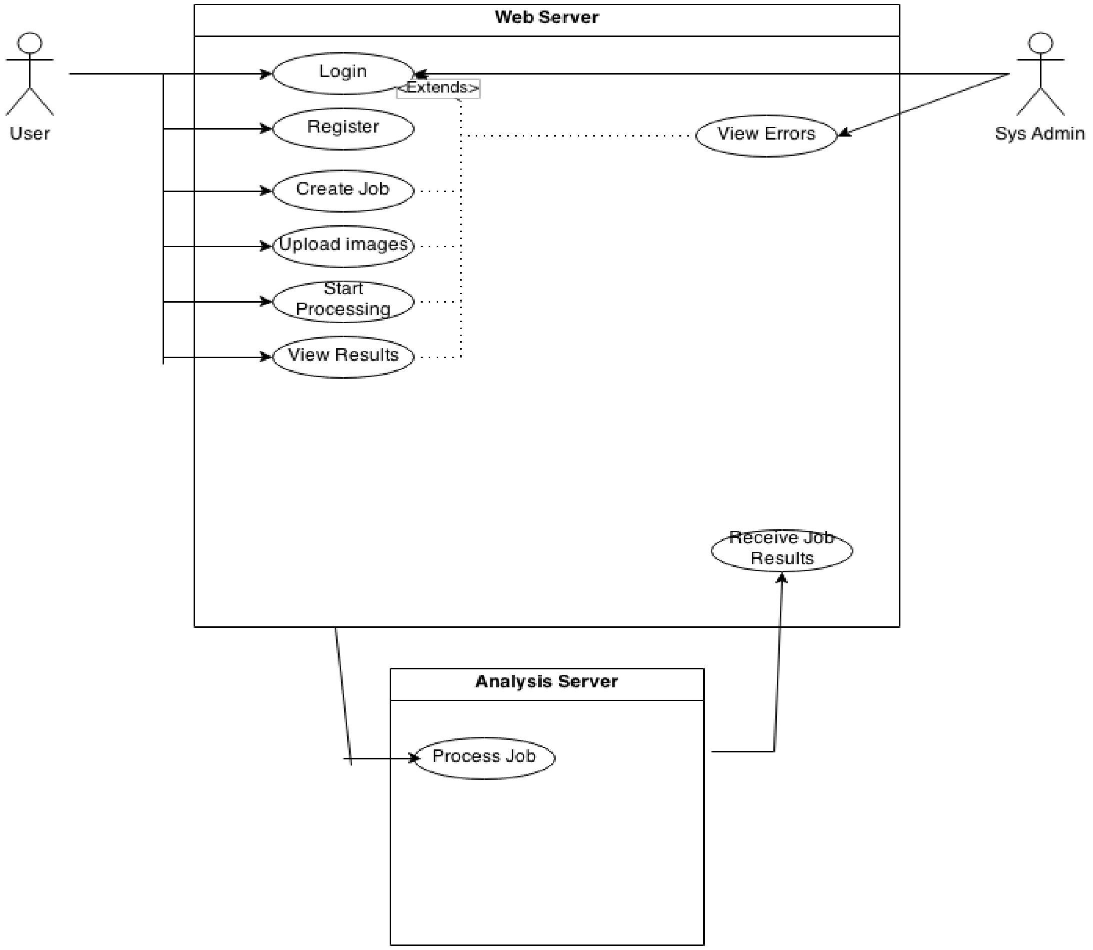
	or drives to look for.	
Job Options	Specify data regarding each of the possible processing options, either to request they be completed or to report on their status.	Required

UAV operation and Image capturing

Feature	Description	Priority
Capture Images	Capture images at a regular interval while in flight and save to an SD card.	Deferred (No hardware)

3.3 Behaviour Requirements

3.3.1 Use Case View



3.3.2 User Stories

As a user...

1. ...I need to be able to log into the website securely
Given: A user name and a password (2 sets: valid, invalid)
Tests: System can process login, and if it is valid, allow login and set page to the user homepage. If it is not valid, prevent login and offer user registration

2. ...I need to be able to register with the website
Given: Completed fields for the user registration page
Tests: a new user entry can be added to the database

3. ...I need to be able to create a new analysis session
Given: Create new session clicked
Tests: a new analysis session will be created in the database, the directories for the images will be created, and the user is redirected to the upload page

4. ...I need to be able to manage sessions
Given: A user login
Tests: The user homepage will display the current sessions (if any)

5. ...I need to be able to upload images to analyze
Given: A set of images to upload
Tests: the images are saved to the appropriate analysis session directory

6. ...I need to be able to have the images analyzed
Given: An uploaded set of images and a click on the analyze button
Tests: the images are analyzed and output images are generated in the appropriate user directory

7. ...I need to be able to view analyzed images
Given: Analyzed images in the appropriate user directory
Tests: the images (if any) can be displayed to the user

8. ...I need to be able to access the captured images in order to upload them
Given: A UAV (which has just flown)
Tests: the images will be present on an SD card which may be inserted into a computer for transfer

9. ...I need to be able to specify the region in which the UAV will fly and capture images
Given: A UAV and coordinates
Tests: the UAV can successfully fly through the region and store images to its onboard memory

As a web server:

10. ...I need to be able to access pending, queued, and completed analysis jobs
Given: A database with entries of the job and a job to query
Test: Return status of the job

11. ...I need to be able to detect errors (if any) for a particular job from the analysis server
Given: A return value from the analysis server program
Test: Display error to the system administrator

12. ...I need to be able to start new jobs for the analysis server
Given: a job to process
Test: The Analysis server can be successfully started for that particular job

13. ...I need to be able to restart jobs in the event of a server outage
Given: pending and queued jobs in the database right after a restart
Test: The Analysis server can be successfully started for each, and all jobs may be completed

14. ...I need to be able to manage currently pending job count with a queue
Given: a full pending jobs list and a new job
Test: The new job gets added to the queue instead of starting up the analysis server again

15. ...I need to be able to convert a GPS coordinate into an embedded google maps element with a marker over that location
Given: a GPS coordinate pair
Test: Generate the expected html string for the embedded google map element

As an image Analysis Server

16. ...I need to be able to examine images and find anomalies
Given: a test set of paired visible & NIR images with and without anomalies
Test: The anomalies are correctly identified (if any are present in the test set)

17. ...I need to be able to determine plant stress levels
Given: a test set of visual & thermal images of plants
Test: the anomalies are correctly identified (if any are present)

18. ...I need to be able to differentiate plants from non-plants
Given: a test pair of images (visible & NIR) containing plants and non-plant objects
Test: the plants are correctly differentiated from the non-plants

19. ...I need to be able to compute Normalized Difference Vegetation Index values for each pixel

Given: a test pair of images (visible & NIR) containing plants in various stress levels and non-plant objects
Test: the plants are correctly colored red through green while non-plants are colored magenta or blue

20. ...I need to be able to load a specific plant profile to tune the Normalized Difference Vegetation Index values for a specific plant species
Given: a plant parameters file and image testpair for non-standard plant
Test: the plant's stress levels can be correctly identified

21. ...I need to be able to match up a pair of slightly mis-aligned images to process them
Given: two images (one visible, one NIR) within +/- 64 pixels of eachother
Test: the correct overlap between the images can be found

22. ...I need to be able to calculate the location of an image
Given: a set of visible images and a telemetry log from the X8 UAV
Test: the location of the images may be derived

As an Administrator

23. ...I need to be able to view any critical errors that are occuring
Given: a list of errors in the database
Test: The list of errors is displayed on a web page (if logged in)

24. ...I need to be able to securely log in
Given: a login name and password, one valid and one invalid pair
Test: The valid pair allows a login, the invalid pair does not allow a login

4 Other Non-functional Requirements

4.1 Performance Requirements

1. The Infrared imaging system shall capture images at a resolution of no more than 1 square centimeter per pixel.
2. The Visible spectrum imaging system shall capture images at a resolution greater than or equal to the Infrared imaging system.
3. The Image Processing system shall recognize anomalies in plant stress levels as well as temperature levels versus the ambient temperature.
4. The Image Processing system shall recognize anomalies of at least 1 square meter in area
5. The Web services system shall process and report all data from a single session within an hour. This requirement excludes any bandwidth limitations which may hinder the acquisition and reporting of images as they are transmitted to and from the user.

4.2 Safety and Security Requirements

Website and information security is being managed by our mentors at PowWow Energy.

Safe operation of the UAV requires that:
1. The UAV’s flight path will allow it to avoid any obstructions upon which it may crash
2. The UAV will not be instructed to fly farther than it can safely return to the launch site on one full battery with a 15% safety margin.

3. The UAV will be able to return to its launch site and attempt landing if unfavorable conditions arise (such as the sudden arrival of unfavorable weather conditions).
4. The UAV will fly below 400 feet above the ground to prevent collisions with FAA regulated aircraft in public airspace.

4.3 Software Quality Attributes

4.3.1 Robustness

The servers shall not lose vital customer data or customer history in the event of a harddrive failure. They shall maintain a backup or backups of the images uploaded, as well as of the user database on separate harddrives, optionally on backed up daily or more frequently to remote sites as well.

4.3.2 Availability

The servers shall be able to continue providing services to users in the event of a harddrive failure. By using a RAID-equipped NAS device or server, the user data, sessions, and results should all be available to the servers even if a harddrive fails.

4.3.3 Maintainability & Adaptibility of the Image Processing Server

The image processing software shall be maintainable and expandable to allow future crop profiles to be added in without having to extensively alter or rewrite any image processing modules. This will be accomplished by allowing parameters from a crop profile file to be passed to the image processing server to allow it to adjust its detection parameters from the default values in order to correctly detect anomalies for fields with different species of plants. By adding a new file that may be used, a new species or set of species may be supported without having to rewrite any of the image processing code.

4.3.4 Maintainability & Adaptability of the Web Server

The web server software shall be maintainable so as to allow compatibility with future web browsers of the Chrome and Firefox series, as well as to allow expansion to support compatibility with other web browsers. This will be accomplished through using non-proprietary HTML conventions in the formatting of the web pages, as well as using expandable and easily changed plaintext HTML templates or generator scripts which a maintainer may alter without having to rewrite any of the core code.

4.3.5 Portability of the Service

The Service shall be useable on any modern PC running any current, supported version of Windows, Mac OS, or Linux and has an internet connection. This shall be accomplished by restricting user interaction with the services through only a website, which will work on any current implementation of Chrome or Firefox.

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>