UNIVERSITY OF NEW MEXICO

System Requirements Definition

For The

Collaborative Solar Image Annotation

Prepared For:
Professor Marios S. Pattichis
pattichis@gmail.com
Director, Image and Video Processing and Communications Lab
Chair, Computer Engineering Program
Dept of ECE, UNM
www.ivpcl.org

and

AFRL Space Weather Center of Excellence
Dr. Nick Arge
Dr. Carl Henney
Dr. Rachel Hock

Prepared by:
ECE 435 Fall Semester Project Team

02-OCT-2014

Table of Contents

Revision History

Version	Reason For Changes	Date
1	Initial Release	02-OCT-2104

1. Introduction

1.1 Purpose

The objective of this project is to provide a collaborative internet application (web app) in which users will be able to annotate solar observational images. Users will use a mouse to click along the perimeter of coronal holes in the images. The annotated data will be joined with its respective image and made available for further analysis by the image provider.

The motivation driving the project purpose cited by the sponsor are concerns of space weather generated by solar activity and its impacts which can cause significant damage to spacecraft operations, critical satellite operates like GPS, astronauts, radio communications and the power grid. The data from this project will also improve space weather prediction capabilities.

& local files

1.2 Product Scope

This solar image annotation software will be capable of running on an open-source web server. The web app will retrieve solar images from the Solar Dynamics Observatory website and then allow users to annotate coronal holes on images. Multiple users will be managed and supported, and collaboration among the users will be allowed. Software will consist of PHP, MySQL, JavaScript, HTML and CSS only.

You do need a library for manipulating FITS files. You may have to use Python or C++ through CGI. C++ is best!

1.3 References

A presentation by the sponsor, Professor Marios S. Pattichis, was given on the 18th of September 2014 in order to promote this project for development. The presentation slides describe general project requirements, specify software components and recommend the book "Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites", June 2014 by Robin Nixon. In addition, the presentation covered the motivation for the project, names of AFRL Space Weather Center of Excellence co-sponsors and why the sponsors are interested in promoting this project to UNM students.

The FITS part can wait. You can assume PNG files for the first part.

2. Overall Description

2.1 Product Perspective

The solar annotation web app is a follow-up to a similar existing Linux application. This web application will have similar general functionality; users will be able to view images, mouse-click on coordinates and store their position in the database. However, this is a completely new standalone program, written from the ground-up with web technologies in mind. Our web app has more features than the original Linux application, users can collaborate on images and the solar image download process is completely automated. ✓

2.2 Product Functions

The first major function is that users must be able to make an account and log into the site where they can view their solar annotation data. Administrative users need to have the ability to see all user annotation data, with the ability to easily download and export this data. ✓

The web app has to be able to pull the latest solar image, convert it from FITS to the necessary compressed image format, likely PNG. ✓

For annotating images users will need to have the ability to click on screen, make waypoints that are stored in a coordinate form with the top left of the image being (0,0) and the bottom right of the image being (1,1). ✓

On the backend the web app needs a MySQL database. In this database user information such as first name, last names, passwords, image information, and coordinates.

info is stored in FITS format.

2.3 User Definition

Types of users will consist of administrators and regular users. Regular users will have the ability collaborate with other users and annotate solar images. Administrators will also have the ability to annotate solar images and collaborate, but they will also have the ability to view all other coronal solar annotations from other users and save/download annotation output to a CSV file (annotation data can be pulled from the MySQL database).

2.4 Operating Environment

ok.

The operating environment will consist of an x86_64 Linux web server running either Red Hat Enterprise, CentOS, Ubuntu Server, or Debian. Options for the operating system depend on where the website is hosted. Software will be a full LAMP Stack including Linux, Apache, MySQL, PHP, JavaScript, and CSS. Various web browsers with different versions will be needed for testing on the client side including: Internet Explorer, Firefox, Google Chrome, and Safari.

2.5 Design and Implementation Constraints

Web data can be exchanged in XML or JSON.

As mentioned briefly above the design of the web application is limited to the use of HTML5 or previous versions, CSS, PHP, MySQL, and JavaScript. Web page formatting can be in ~~either~~ HTML ~~or JSON~~. All code produced including client and server side scripting will be considered the "deliverables" of the project and be open-source. Application layer protocols used between the web application and server will be the standard HTTP and HTTPS protocols for distributed information systems.

The images retrieved from the Solar Dynamics Observatory (SDO) website are not under copyright and are available for non-commercial use and educational purposes. The images can be reformatted into different image formats of various resolutions as necessary for use with the web application. A format of PNG was suggested but not required for use with the web application. The only built-in security will be user account access. A visitor to the site may view but not modify or collaborate with other analysts or participate in forums unless they have established a user account and successfully logged into the application.

Also allow users to upload their own images.

2.6 User Documentation

Basic instructions on how to use the web app will be provided in a help section on the site. Additional help can be found within a thread on the forum.

2.7 Assumptions and Dependencies

Our solar image web app will be easily deployed and ran on open-source platforms, i.e. the web app will not need proprietary software in order to function properly. The application will be compatible with most modern, relatively up-to-date web browsers (Chrome, Firefox, Safari, and Internet Explorer). Accurate retrieval of solar images needed for annotation will depend on NASA's SDO website.

Appendix A: Glossary

CentOS: Linux distribution that attempts to provide a free, enterprise-class, community-supported computing platform which aims to be 100% binary compatible with its upstream source, Red Hat Enterprise Linux (RHEL)

CSS: Style sheet software language used for describing the look and formatting of a document written in a markup language.

CSV: A format used in data export files in which data is separated by the character ",".

FITS: An open standard defining a digital file format useful for storage, transmission and processing of scientific and other images.

HTML: Standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.

HTML5: The latest standard of HTML.

HTTP: The underlying protocol used by the World Wide Web to define how messages are formatted and transmitted.

HTTTPS: Secure version of HTTP in which data is not transmitted in plain-text format.

JSON: A lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999.

JavaScript: An object-oriented computer programming language commonly used to create interactive effects within web browsers.

LAMP: is an acronym for an archetypal model of web service solution stacks, originally consisting of largely interchangeable components: Linux, the Apache HTTP Server, the MySQL relational database management system, and the PHP programming language. As a solution stack, LAMP is suitable for building dynamic web sites and web applications.

MySQL: Open-source relational database management system.

PHP: A server-side scripting language designed for web development and also used as a general-purpose programming language.

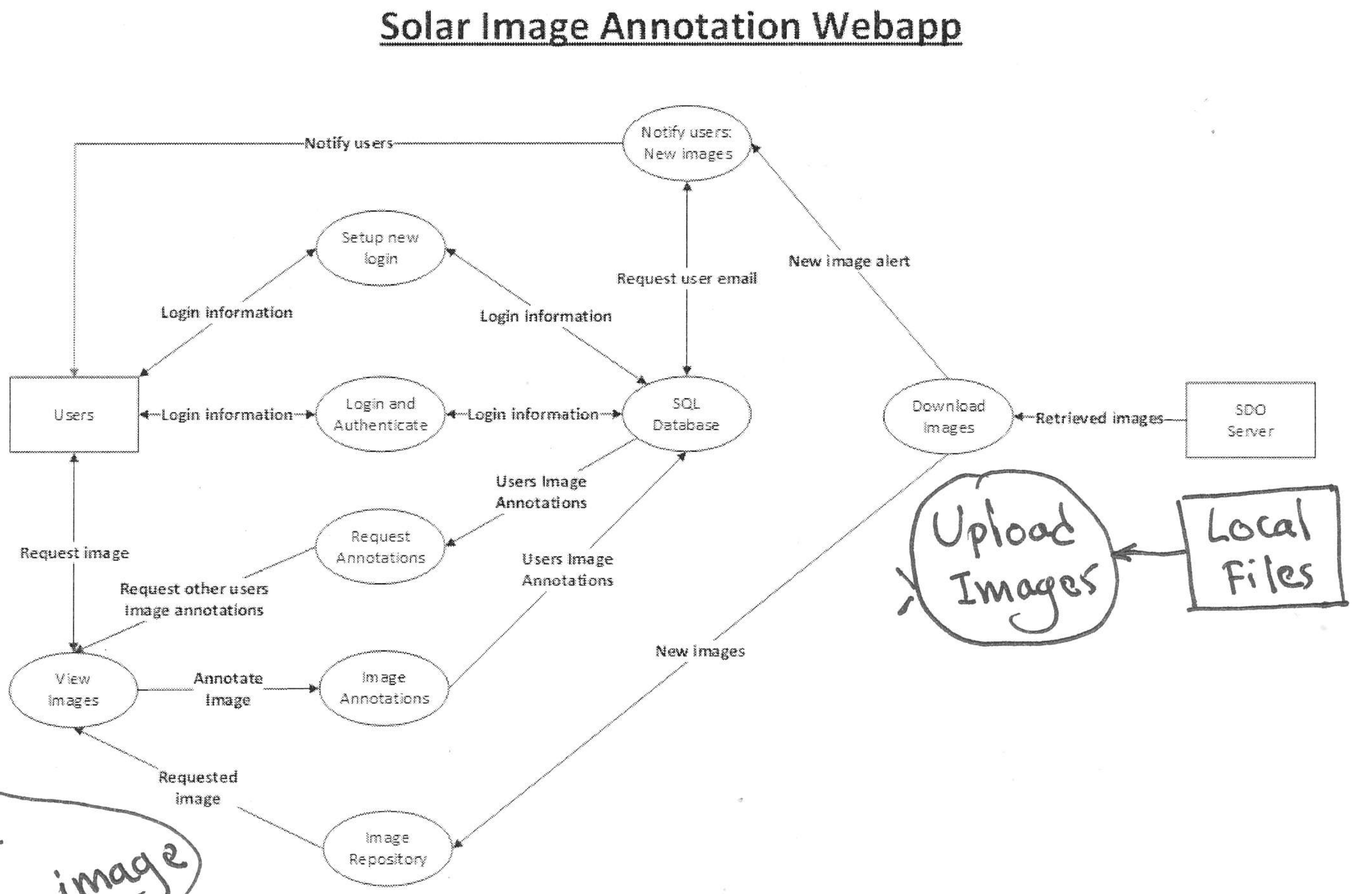
PNG: A computer file format for image compression.

Web app: Any application software that runs in a web browser. It is created in a browser-supported programming language (such as the combination of JavaScript, HTML and CSS) and relies on a web browser to render the application.

Appendix B: Acronyms

AFRL	Air Force Research Laboratory
CentOS	Community Enterprise Operating System
CSS	Cascading Style Sheets
CSV	Comma Separated Values
FITS	Flexible Image Transport System
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTML5	Hypertext Markup Language Version 5
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
LAMP	Linux, the Apache HTTP Server, the MySQL
MySQL	My Structured Query Language
NASA	National Aeronautics and Space Administration
PHP	PHP: Hypertext Processor with Personal Home Page as recursive "backronym"
PNG	Portable Network Graphics
SDO	Solar Dynamics Observatory
UNM	University of New Mexico

Appendix C: Data Flow Diagram



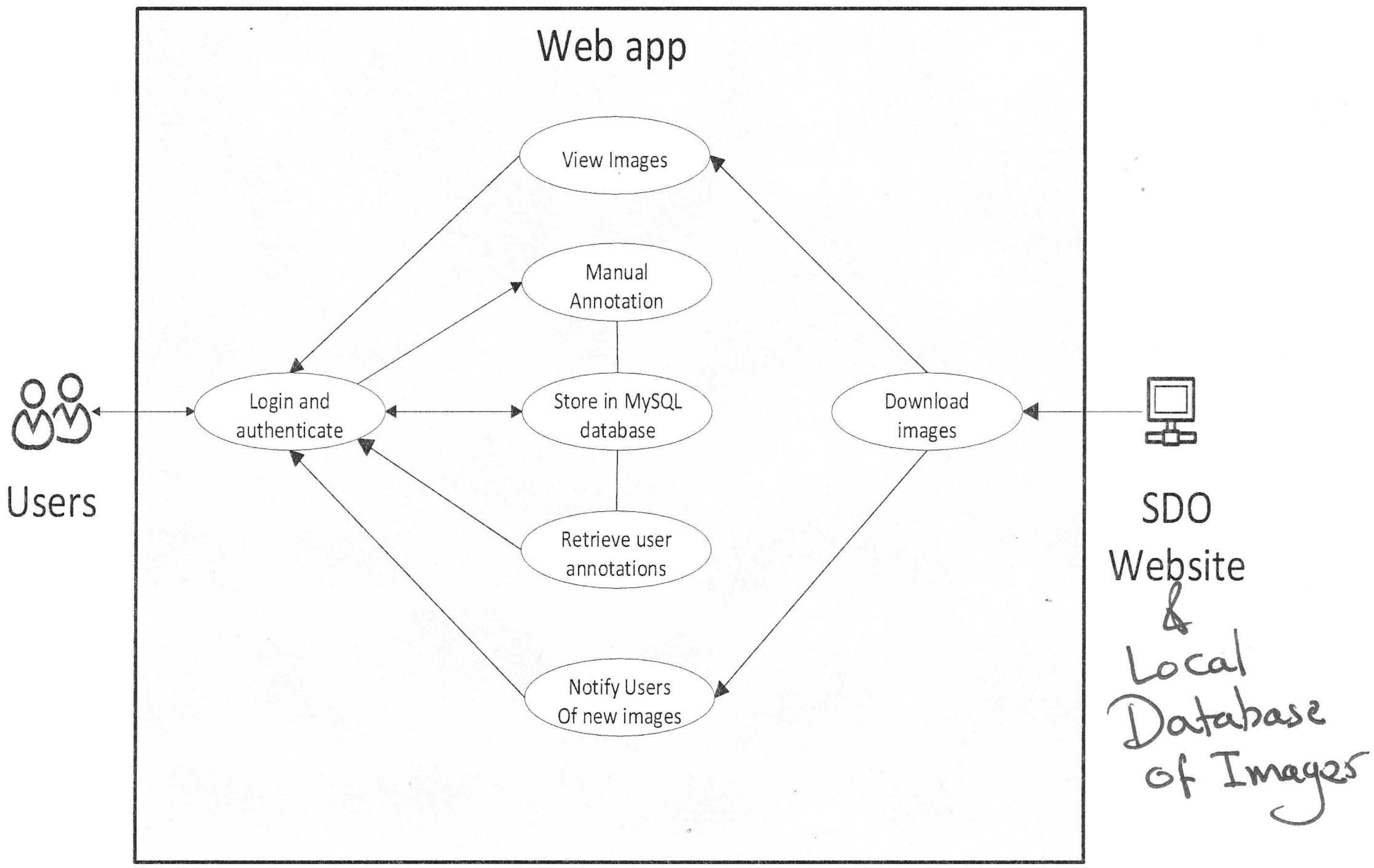
Data Flow Diagram

Start with one image

Your SQL will have to have many ~~at least two~~ databases:

- One for user information (login, etc)
- One database per user that stores the images that they did.
- Each image will have many coronal holes
- Each coronal hole will save all of the mouse clicks.

Appendix C: Use Case Diagram



Use Case Diagram

In addition to "Use Case", you need to have "Administrator Use" who will be able to access the images using a query.

- Organize images by date.

- Please start with one image to see what it takes to build the databases.