San Francisco State University

Department of Computer Science

Culminating Experience Report

Title: Email Server for Archiving Field Observation Data

Author(s): Alexander Birger

Date: 10/6/2013

Abstract:

In this project email based server for observing apps will be created. Email server to be used is James Server. James Server is using Java and runs on Windows, Linux, and MacOS. A Mailet will be used with James Server. The server should accept emails from observing apps. This could be a single observation with photo or multiple observations without photos. The project is different from other projects because it requires an email for sending data. The email is later parsed. The project also includes data extraction, storage in Postgres db, interface for viewing field observations, and bulk upload of field observations to a remote server.

Table of Contents

I. Introduction

1. Overview of Email Server for Archiving Field Observation Data

The Observer Apps are currently popular because of ease of use and declining prices of hand held devices. These devices also can be used in the field. The observer app is software for mobile devices that enables identifying, observing, and sharing observations of different species of animals or plants efficiently. Information specific to particular specie can be easily recorded. Usually a photo of specie can also be added. Another advantage is ease of species identification and sharing of the data. Also, the data can be edited and confirmed. Some apps also allow viewing the data on a map. This project focuses on the observer app by implementing the email server for archiving field observations. The email server for archiving field observation data is an important issue. Also, in a same way other data collection protocols can be automated. This project consists of three parts: main Mailet application and two accessory applications written jsp. This section also contains author contributions, related work, and outline of the report.

The email server for archiving field observation data is important because it can ease the task of field data collection. It is much easier to use hand held devices then the old way with pencil and paper. With every year more different hand held devices and technologies are emerging. Another factor is that the hardware is getting cheaper. In a few years one cannot see data collection without electronic devices. This project takes advantage of these factors and uses various technologies to ease the task of data collection. The main part of the project is the Mailet. It is written in Java and works with Apache James email server. Mailet is used in order to take email and save data in the database and picture from email on the hard drive. There are two additional components implemented as web applications. They are written in jsp and work with Apache Tomcat. There is a growing interest in hand held devices mainly because of lower cost of hardware. Different and well known hand held device technologies are emerging. The devices are easy to use and functional. This project utilizes hand held device technology for accurate data collection in a field. These devices can be used for transfer of observations to a laptop field archive without internet connectivity or power outlet. The next task would be to manually manage observations on field server and transfer observations from field server to internet based public observations archive. The task is accomplished by generating xml file with data and zip file with photos. The remote server then uploads both files and puts xml in the database and unzips the photos zip file into a remote server directory. In this project the email server will be implemented on a laptop with a WiFi HotSpot, thus it will not be necessary to have Internet or WiFi connectivity or a power outlet to use the server.

Also there is a way to automate other data collection protocols. One example is CalFlora website [3]. There are several functionalities that can be implemented as an application. iPod application would send email to Mailet, which will parse the email, put data into database, and save email photos into field server directory. Also, field and remote web servers need to be implemented. Entered data needs to be sent to the server as an email. The first functionality is login and register as a contributor. Next is adding a plant observation functionality, which includes photo upload. Adding plant observation includes scientific name, common name, observer, source, and other entries. Next is adding weed observation functionality, which includes photo upload. This also includes scientific name, common name, observer, source, and other entries. Next is adding a checklist functionality, which includes photo upload. This also includes observer, source, observation, date, collection, location name, natural status, and location description, and other entries. Next is adding a place to view California native plants functionality, which includes photo upload. This also includes observer, source organization, observer date, collection, location name, and other entries. Next functionality is photo observation file upload. After field server receives data, the data can be viewed, edited, and deleted through a field web server. Another functionality would be to send data from field server to remote web server. This will be accomplished by saving field server data and images as xml file and photos zip file. Next, remote server uploads xml file and zip file and puts data into a database and unzips the zip file into a directory.

The task of field data collection in order to identify plants often arises. Several things can be done in order to collect data efficiently. This project will involve creating an email server in order to archive field observation data without internet connectivity or power outlet. The project will include the Mailet and two web applications. Mailet is the main part of the application. Two web applications are Observation Edit and Observation Server web applications. Email server to be used is James Server (using Java, runs on Windows/Linux/MacOS). A Mailet will be used with James server. Other email software considered are Mozilla Thunderbird, Alpine, JavaMail, and SquirrelMail (a webmail package written in PHP). The server should accept emails from observing apps. This could be a single observation with photo or multiple observations without photos. Also, the project includes data extraction, storage in Postgres db, simple interface for viewing field observations, and bulk upload of field observations to a remote server. Observations Edit web application, written in jsp, is used to view, edit, and delete observations. It also saves data for transfer from field server to remote server in form of xml file with data and a zip file with photos. Observations Server web application, also written in jsp, is used for uploading generated xml file and zip file to a remote server. The data from xml is added to a database on the server and photos from zip file are added to photos directory on the server. The problem is important because it can help field scientists to link up with a server in order to identify different plants. The problem is interesting because it brings a new way of identifying plants.

2. Problem Statement

This project will involve creating an email server in order to archive field observation data without internet connectivity or power outlet. Two accessory web applications will also be developed. The email server used is James Server. It works with a Mailet. The Mailet will capture email sent to James Server and put the email data into database and the picture from email into a directory. Also developed are Observation Edit and Observation Server web applications. Observation Edit web application will reside on a field server. It will enable viewing, editing, and deleting records, as well as saving data as xml file and photos as photos zip file. Observation Server web application will reside on archive server. It will upload xml file and zip file and put xml into a database and unzip photos zip file into a directory on the archive server.

3. Contributions

The main author's contribution includes the Mailet. It contains the following features: main mailet, mailet data object, mailet db module, mailet file module, and mailet parse module. Error handling also took part of the project. Another part of the project was working on commenting the code. There was also some database work involved. The main mailet looks into the content of the message, which could be a string message or MimeMultipart message. After this, it calls one of the possible parsers (email observation with photo, email observation without photo, or bulk observations). After this the parser parses the emails. The parsed data is saved into database. Also, if the observation has file, it is saved on in the directory. Mailet data module holds the information about the data to be saved in the database, such as phenology, habitat, family, and species. Also, if the image file is provided, the module holds the file. Mailet DB module calls the postgre sql jdbc driver and saves the data to the database. The data is saved once the jdbc connection is established. It also chooses whether to save the file, depending on whether the file is provided. The name of the file is also saved into the database, also depending on whether the file is provided. The file is retrieved through MimeMultipart object. If the image file is not present, the string containing the message is parsed. Mailet File module is responsible for saving the file to the directory structure. Mailet parse module is responsible for parsing three different formats, such as observation with photo, observation without photo, and bulk observations. When email with photo is parsed, MimeMultipart object is retrieved as content. The first thing is to check for is whether the attachment is present. If it is present, it is retrieved. Next, the string part of the message is parsed. When parsing email without photos the string part of the message only is parsed. The same is done for bulk observations. Database work involved writing sql for the table, which would hold the data.

Observations Edit web application is written in jsp. index.jsp presents the screen for viewing, editing, and deleting the database records. editentry.jsp presents the screen for editing the database entry. Observations Edit application also allows for generating field server data xml file and pictures zip file for transfer to remote server. editsettings.jsp asks for xml file path, zip file path, and photos folder path. The next screen is generatexml_zipfiles.jsp. It generates xml file with data and zip file with photos.

Observations Server web application is also written in jsp. editsettings.jsp is used to set the folders for the remote server. It includes unzip folder, upload folder, and temp folder. index.jsp is used to upload xml file with data and zip file with photos. Next, xml parser puts xml file into the remote server database. The next step is to unzip the zip file with photos and put photos into remote server directory.

4. Related Work

In SFSU "Parks Observer Server and Web Application" [2] client devices are sending observational data to servers and social networks. Data storage and retrieval is done through indexed and configured Mongo DBMS. The server is written in NodeJS and Express framework. The client is written in JavaScript, AngularJS framework, and Twitter's Bootstrap CSS style library. Another project is SFSU “Server for Dichotomous Key & Field Data Collection Application”. [14] Field observations could be collected by scientists and amateurs through mobile devices that email these observations to a central server. The sever parses Observer applications emails and stores observations in databases and Google Fusion Tables. It also provides an administrative interface for data curation. Stored data can be viewed and edited by users through a web browser. Observations are displayed on a Google Map or tabular interface. The server software is written in JavaEE, Struts, Spring, Hibernate, JavaMail, and Java SOAP Web Services.

Another example is WildObs observer app [18]. It allows identifying mammals, birds, snakes, bugs, and plants. Another feature of the app is lookup of thousands of species by name. One can also record their encounters with the wildlife and record the data in local database, as well as National Wildlife Federation Watch program. Next example is Shark Observer application [13]. It allows logging encounters with shark species on the Shark Observation Network. It also helps marine biologists understand sharks better. Calflora website enables to search California plants. [3] Both native plants and weeds can be searched. There are several categories to search, such as plant name, lifeform, native/non-native, elevation, community, rarity, category, and county. There is also a way to add observations. iNaturalist website is where one can search species such as kelp, diatoms, allies, mollusks, reptiles, mammals, birds, amphibians, fishes, arachnids, insects, plants, fungi, lichen, and protozoans. [8] eBird website concentrates on birds. [6] It allows to record the birds, keep track of the birds list, explore dynamic maps and graphs, share the sightings, and to contribute to science and conservation. eBird shares observations with educators, land managers, ornithologists, and conservation biologists. The website was launched in 2002 by the Cornell Lab of Ornithology and National Audubon Society. AmphibiaWeb website allows searching the database and browsing photos. [1] The site can be searched by scientific name, genus, family, order, and country. It also gives information on natural history, conservation, and taxonomy. Encyclopedia of Life website brings information about earth in text, images, video, sounds, and maps. [7] It allows learning about plants, animals, and microorganisms. It also gathers information from other databases. The information is later organized. Leafsnap is an electronic field guide developed by Columbia University, University of Maryland, and the Smithsonian University researchers. [10] The mobile application utilizes visual recognition software to recognize tree species from their leaves photographs. The images contain leaves, flowers, fruit, petiole, seeds, and bark. Right now trees of Northeast are used, but later entire continental United States will be included in the software. Images, species identifications, and geo-coded stamps of species locations are used. Pl@ntNet-Identify is software for automatic plant detection. It utilizes image-based identification software by comparing a given photo to an image database. [12] The Electronic Field Guide (EFG) Project has developed web-based applications for identification of species and ecological observations. [16] The project was created by Departments of Computer Science and Biology at the University of Massachusetts Boston. The funding was from the National Science Foundation. Database management system allows biologists and field naturalists to author electronic guides. The Kemper Garden Center at the Missouri Botanical Garden describes pictures, plant information, and sources of plants [11]. Scientific or common names can be searched. Mountain Watch is a project by Research and Education departments of Appalachian Mountain Club [17]. The mission of the club is to support conservation, education, and recreation for the northeastern mountain ranges of the Appalachian ridge. Mountain Watch evaluates air quality through visibility measurement and collects observations of flowering plants for climate change research. The primary focus of the project is alpine plant monitoring.

5. Outline of the Report

The following chapters in this document describe Email Server for Archiving Field Observation in detail. Chapter one is introduction. Chapter two is a tutorial guide. Chapter three is system architecture. Chapter four is software and hardware configuration. Chapter five contains diagrams. Chapter six describes James, Mailet, and Postgres. Chapter seven describes Observations Edit web application. Chapter eight describes Observations Server web application. Chapter nine is experimental results and analysis. Chapter ten is conclusion and future work. Chapter eleven is bibliography. Chapter twelve is appendix. Next discussed is tutorial guide.

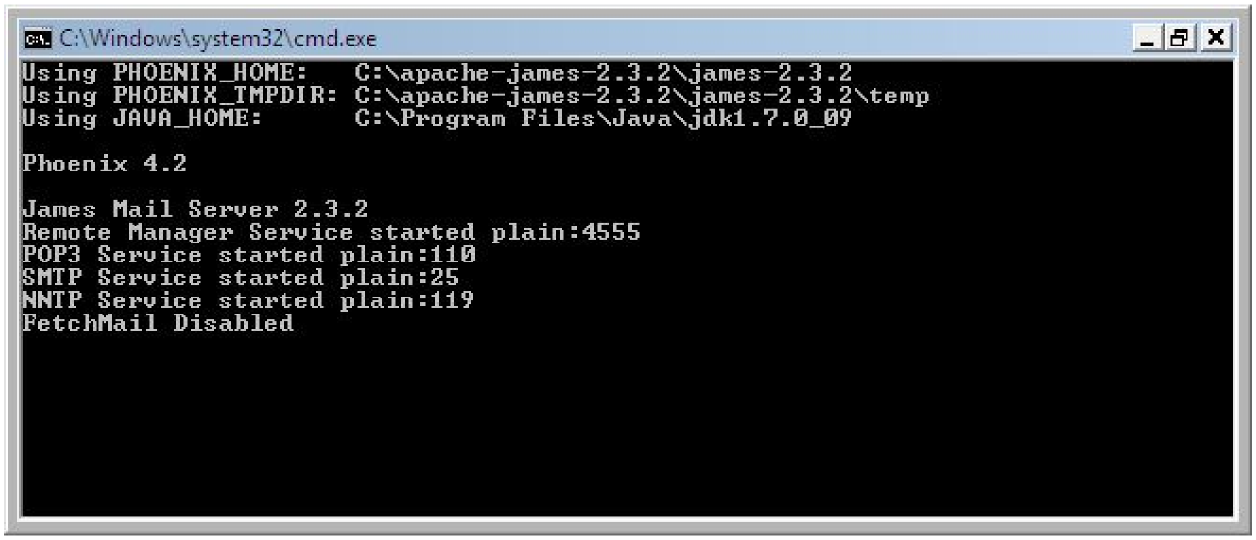
II. Tutorial Guide

The tutorial guide will explain how the project will be utilized by scientists for field data collection. The James Server will invoke Mailet application, which will put data into a database and emailed picture into a directory on a field server. Next, Observations Edit web application will be used to view, edit, and delete the data. The application can also save the data as xml file and put the image files into a zip file on a field server. Another application used is Observations Server web application. It will upload xml file and zip file onto an archive server and put xml into database and zip file into archive server directory.

1. Apache James

- **Start James Server**

  The James Server can be started by running run.bat, located within bin folder. This will also start the Mailet. On Linux systems run.sh is invoked.


C:\Windows\system32\cmd.exe

```
Using PHOENIX_HOME:   C:\apache-james-2.3.2\james-2.3.2
Using PHOENIX_TMPDIR: C:\apache-james-2.3.2\james-2.3.2\temp
Using JAVA_HOME:      C:\Program Files\Java\jdk1.7.0_09

Phoenix 4.2

James Mail Server 2.3.2
Remote Manager Service started plain:4555
POP3 Service started plain:110
SMTP Service started plain:25
NNTP Service started plain:119
FetchMail Disabled
```

2. Mailet

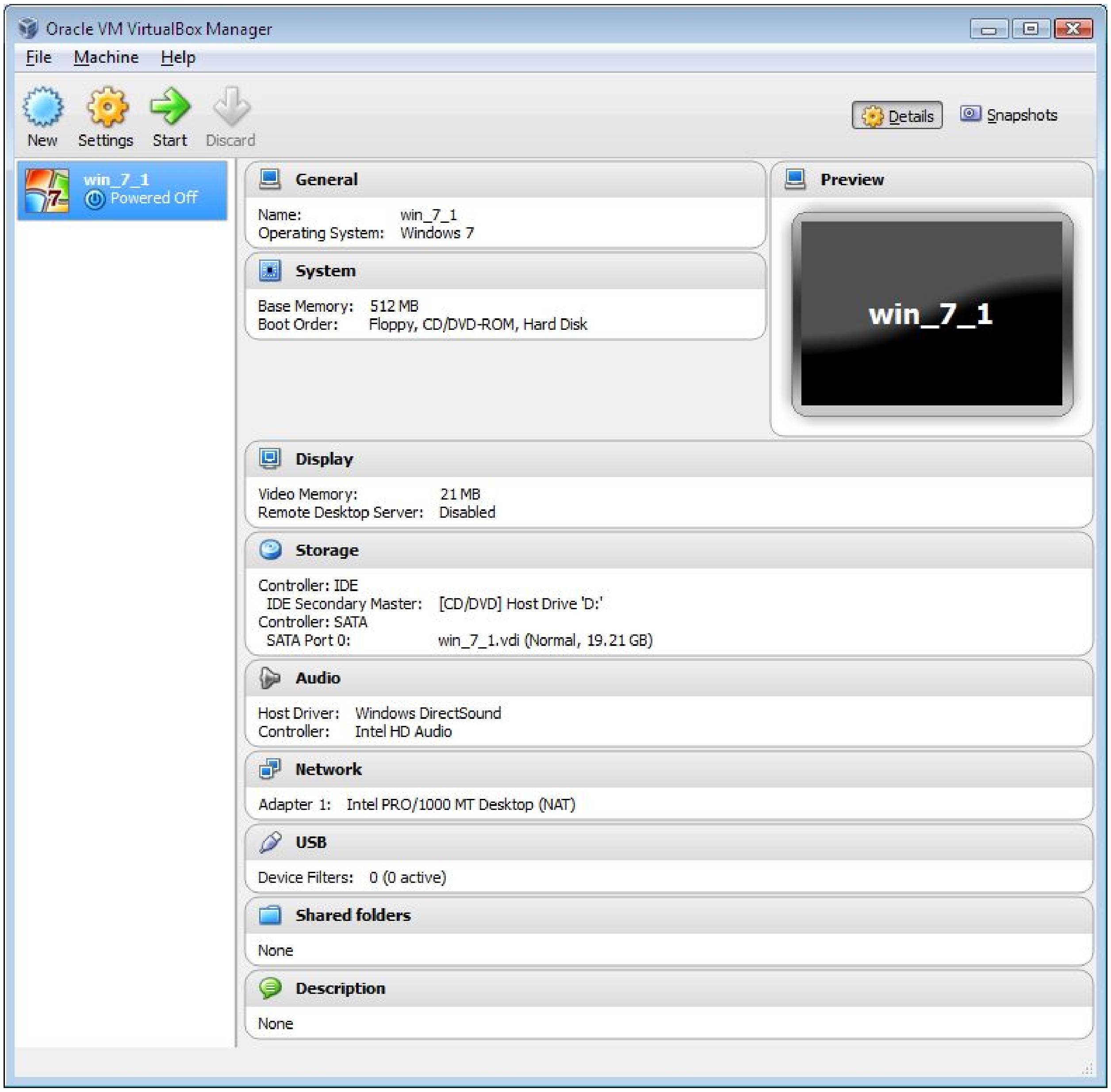
- **Run Mailet application**

  The first step is to run SNFCFlora application. Next, select observations tab and email observations. This can include observation without photo, observation with photo, and all observations. Next, the James Server will invoke the Mailet. The Mailet will put the observation data into database and place the photo file (if it is included in the email) into a directory set in the observations.ini file.

3. Virtual Box

- **Start Virtual Box**

  The first step is to start the Virtual Box.



General
Name: win_7_1
Operating System: Windows 7

System
Base Memory: 512 MB
Boot Order: Floppy, CD/DVD-ROM, Hard Disk

Display
Video Memory: 21 MB
Remote Desktop Server: Disabled

Storage
Controller: IDE
IDE Secondary Master: [CD/DVD] Host Drive 'D:'
Controller: SATA
SATA Port 0: win_7_1.vdi (Normal, 19.21 GB)

Audio
Host Driver: Windows DirectSound
Controller: Intel HD Audio

Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

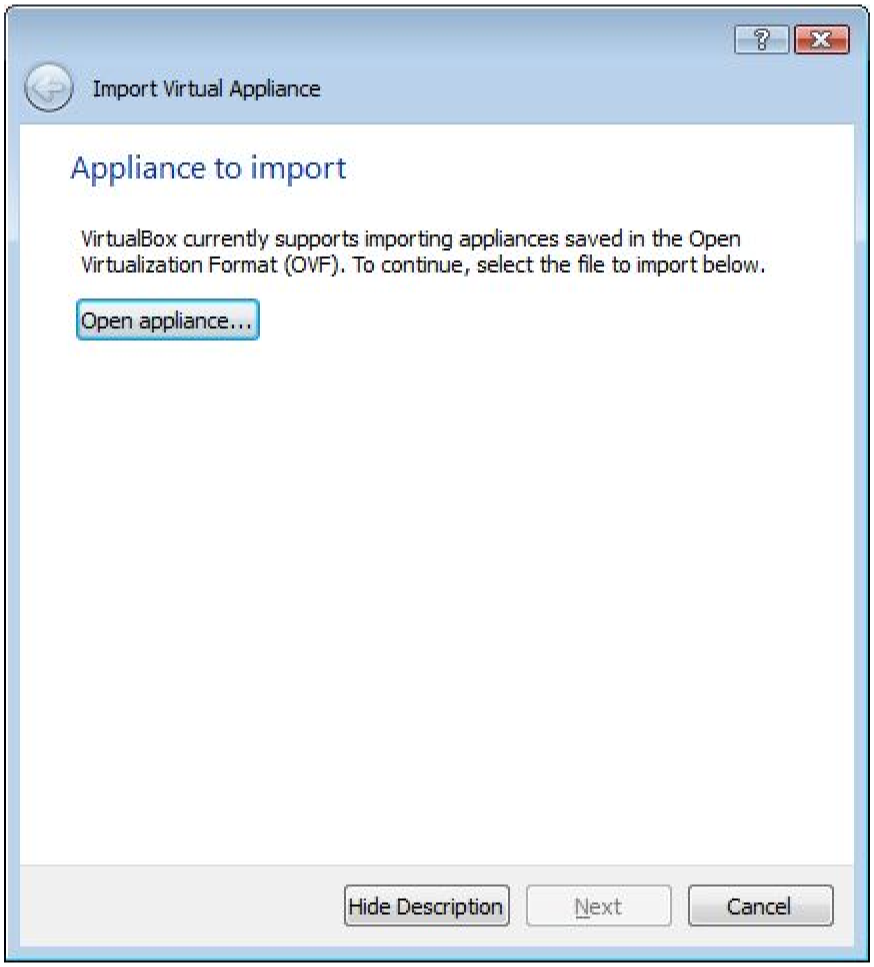
USB
Device Filters: 0 (0 active)

Shared folders
None

Description
None

- **Import appliance**

The next step is to import appliance.


Import Virtual Appliance

Appliance to import

VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

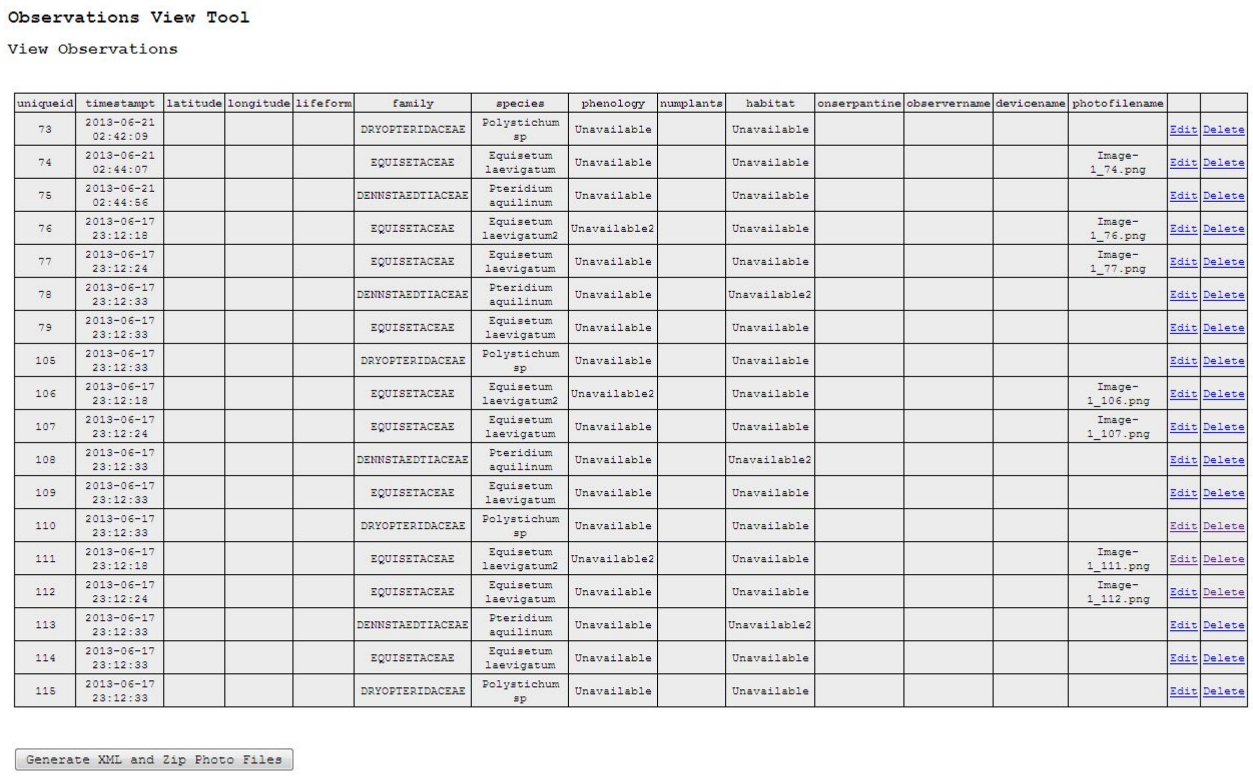
- **Start imported appliance**

Next, start the imported appliance.

4. Observation Edit Web Application

- **Run Observations Edit web application**

Observations Edit web application is invoked through Apache Tomcat.

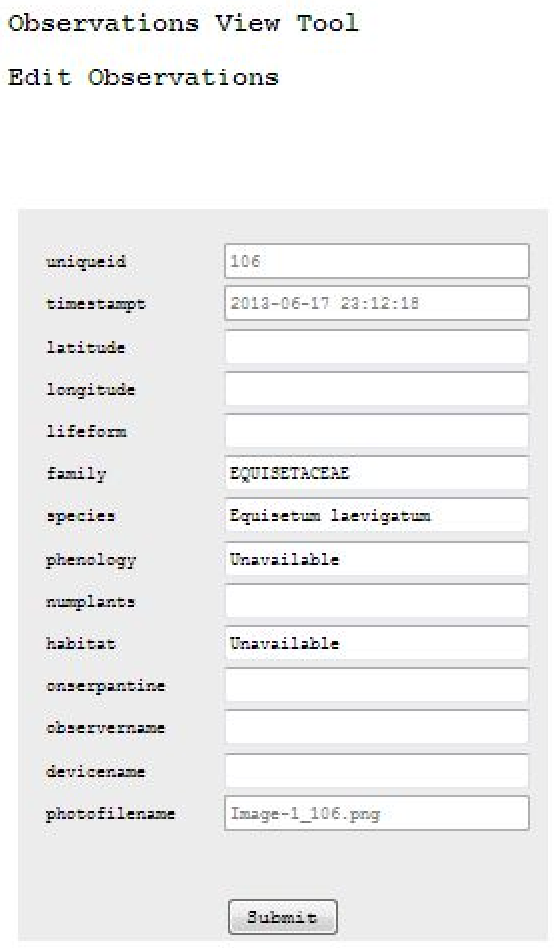


Observations View Tool

View Observations

uniqueid	timestampt	latitude	longitude	lifeform	family	species	phenology	numplants	habitat	onserpantine	observername	devicename	photofilename		
73	2013-06-21 02:42:09				DRYOPTERIDACEAE	Polystichum sp	Unavailable		Unavailable					Edit	Delete
74	2013-06-21 02:44:07				EQUISETACEAE	Equisetum laevigatum	Unavailable		Unavailable				Image-1_74.png	Edit	Delete
75	2013-06-21 02:44:56				DENNSTAEDTIACEAE	Pteridium aquilinum	Unavailable		Unavailable					Edit	Delete
76	2013-06-17 23:12:18				EQUISETACEAE	Equisetum laevigatum2	Unavailable2		Unavailable				Image-1_76.png	Edit	Delete
77	2013-06-17 23:12:24				EQUISETACEAE	Equisetum laevigatum	Unavailable		Unavailable				Image-1_77.png	Edit	Delete
78	2013-06-17 23:12:33				DENNSTAEDTIACEAE	Pteridium aquilinum	Unavailable		Unavailable2					Edit	Delete
79	2013-06-17 23:12:33				EQUISETACEAE	Equisetum laevigatum	Unavailable		Unavailable					Edit	Delete
105	2013-06-17 23:12:33				DRYOPTERIDACEAE	Polystichum sp	Unavailable		Unavailable					Edit	Delete
106	2013-06-17 23:12:18				EQUISETACEAE	Equisetum laevigatum2	Unavailable2		Unavailable				Image-1_106.png	Edit	Delete
107	2013-06-17 23:12:24				EQUISETACEAE	Equisetum laevigatum	Unavailable		Unavailable				Image-1_107.png	Edit	Delete
108	2013-06-17 23:12:33				DENNSTAEDTIACEAE	Pteridium aquilinum	Unavailable		Unavailable2					Edit	Delete
109	2013-06-17 23:12:33				EQUISETACEAE	Equisetum laevigatum	Unavailable		Unavailable					Edit	Delete
110	2013-06-17 23:12:33				DRYOPTERIDACEAE	Polystichum sp	Unavailable		Unavailable					Edit	Delete
111	2013-06-17 23:12:18				EQUISETACEAE	Equisetum laevigatum2	Unavailable2		Unavailable				Image-1_111.png	Edit	Delete
112	2013-06-17 23:12:24				EQUISETACEAE	Equisetum laevigatum	Unavailable		Unavailable				Image-1_112.png	Edit	Delete
113	2013-06-17 23:12:33				DENNSTAEDTIACEAE	Pteridium aquilinum	Unavailable		Unavailable2					Edit	Delete
114	2013-06-17 23:12:33				EQUISETACEAE	Equisetum laevigatum	Unavailable		Unavailable					Edit	Delete
115	2013-06-17 23:12:33				DRYOPTERIDACEAE	Polystichum sp	Unavailable		Unavailable					Edit	Delete

This will allow viewing, editing, and deleting entries. Also, it will generate data xml file and zipped photos file for upload to remote server. In order to edit an entry, edit link needs to be selected. In order to delete an entry, delete links should be selected.

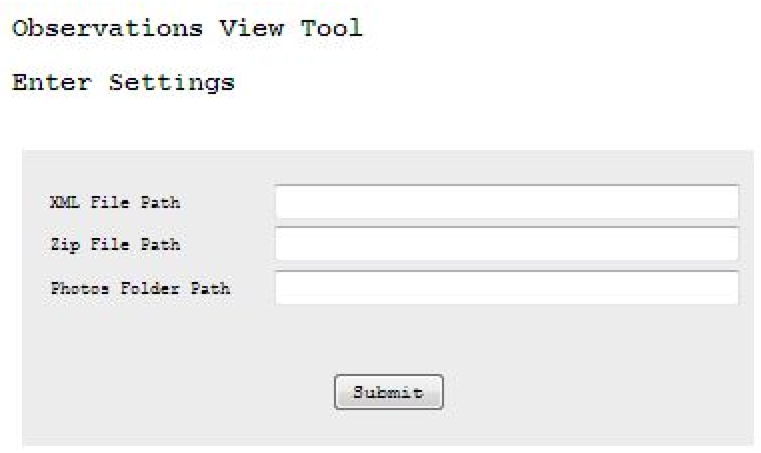


Observations View Tool

Edit Observations

uniqueid	106
timestampt	2013-06-17 23:12:18
latitude	
longitude	
lifeform	
family	EQUISETACEAE
species	Equisetum laevigatum
phenology	Unavailable
numplants	
habitat	Unavailable
onserpantine	
observername	
devicename	
photofilename	Image-1_106.png

In order to generate xml file and zip file button need to be selected. Next, settings need to be entered. This will include xml file path, zip file path, and photos folder path.


Observations View Tool

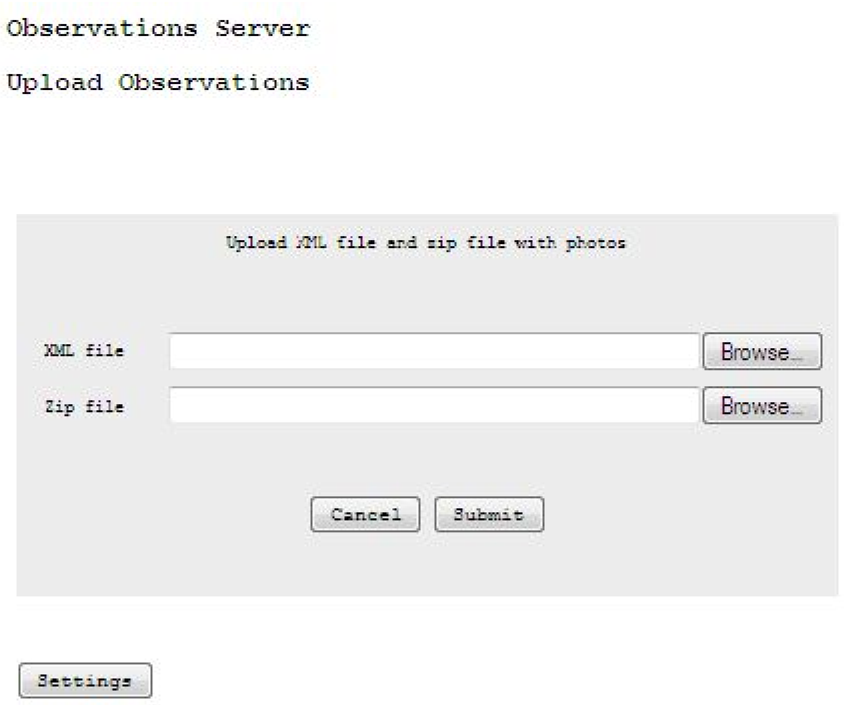
Enter Settings

XML File Path	
Zip File Path	
Photos Folder Path	

5. Observation Server Web Application

- **Run Observations Server web application**

Observations Server web application will upload xml file with data and zip file with photos to remote server, where data is placed into database and zip file is unzipped into a directory. XML file and zip file need to be selected. Next, submit button is pressed.


Observations Server

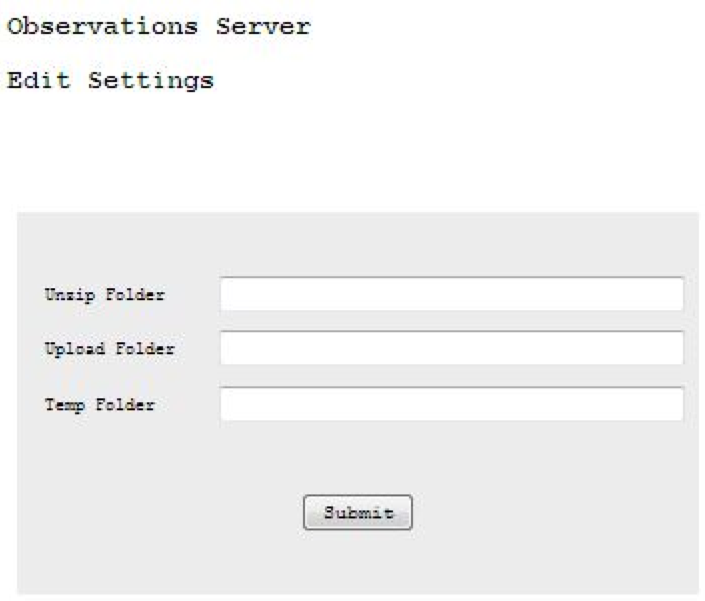
Upload Observations

Upload XML file and zip file with photos

XML file

Zip file

In order to upload observations, settings need to be saved on the server. This will include unzip folder, upload folder, and temp folder. In order to save the settings, submit button is pressed.


Observations Server

Edit Settings

Unzip Folder

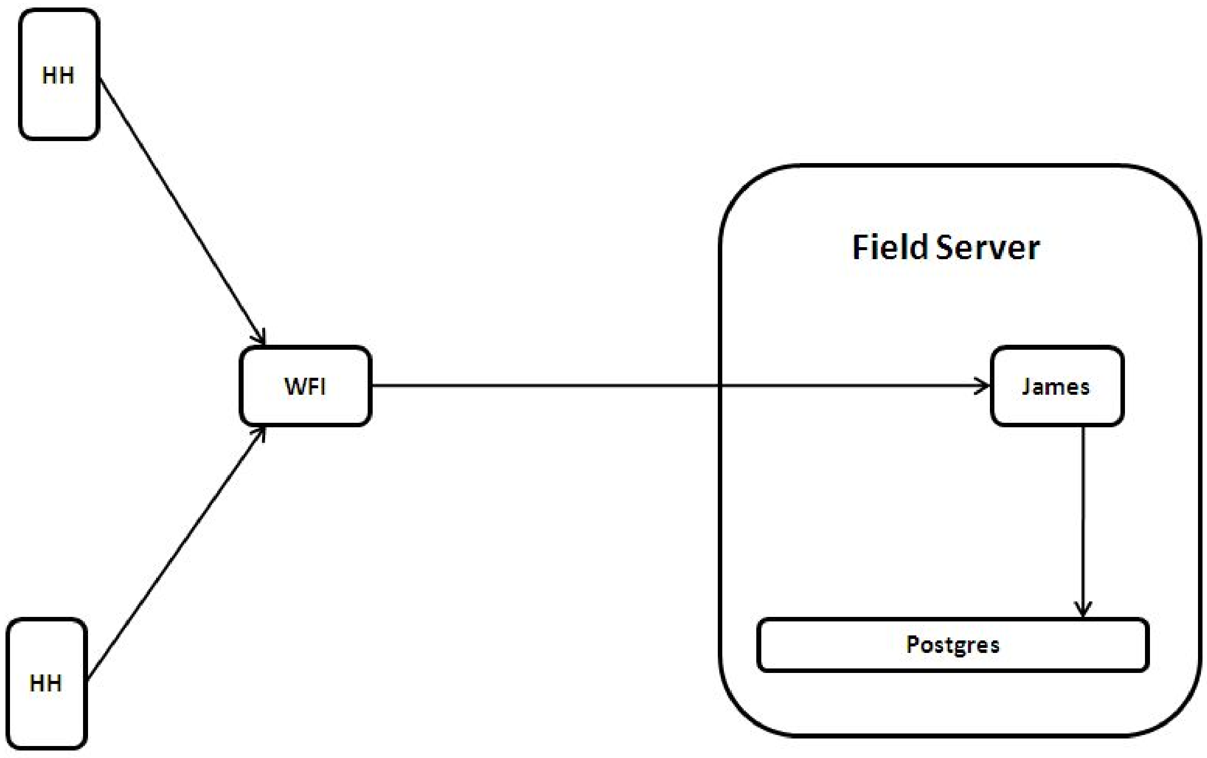
Upload Folder

Temp Folder

This section included the tutorial on how the system can be utilized. The next section will present system architecture and will include diagrams for the system.

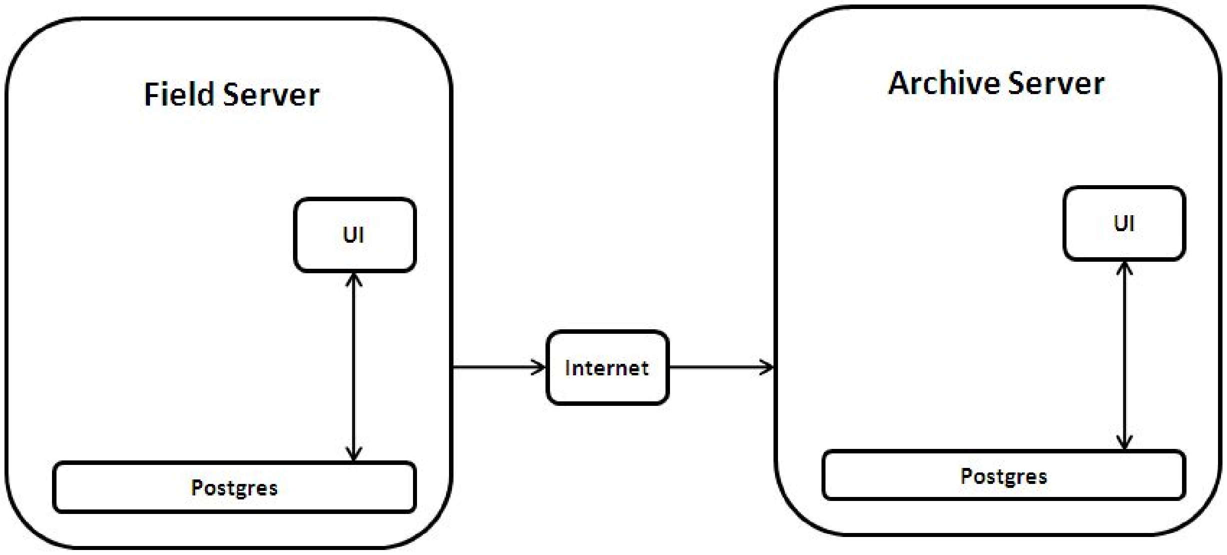
III. System Architecture

This section will present the architecture for the system and will include diagrams. The diagrams will show communication between handhelds and server over WiFi hotspot, communication between the server and archive destination over the internet, and system diagram. The firsts step is when the communication between handheld devices and the field server takes place over WiFi Hotspot. Next, the James Server is invoked on a field server and puts the data into Postgres database and the file into the directory structure.



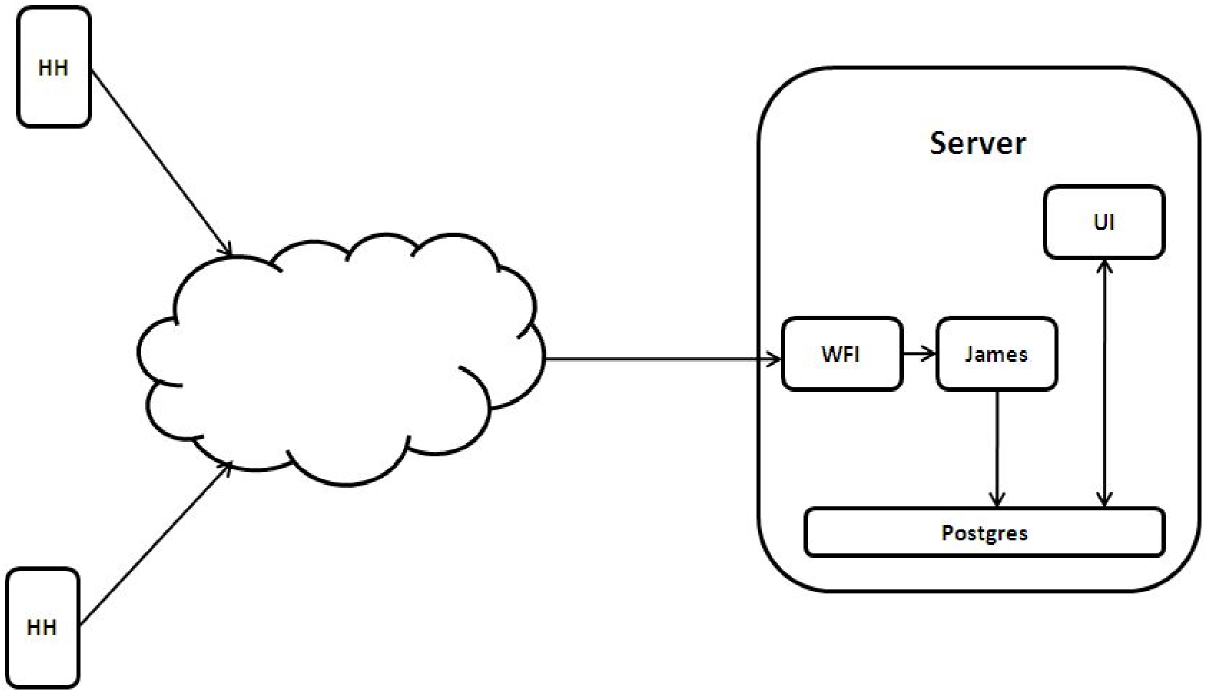
1. Communication between handhelds and server over WiFi HotSpot

Next, the communication between the field server and archive server takes place over the Internet when the field server is back on campus. The data is uploaded from a field server to remote server in the form of data xml file and photos zip file.



2. Communication between the server and archive destination over the Internet

In the next step, the diagram for the whole system is presented.



3. System Diagram

Next, software and hardware configuration is discussed.

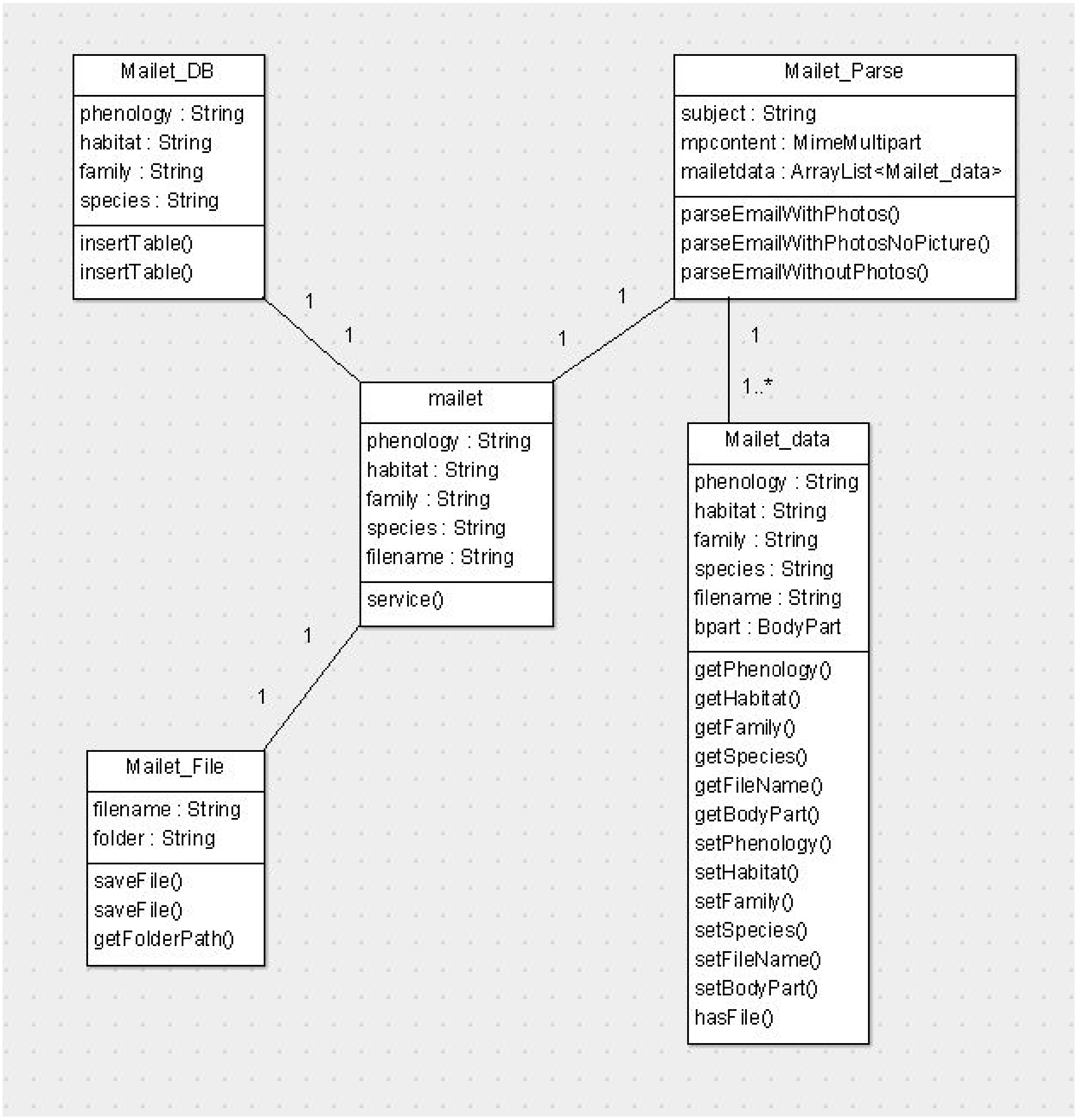
IV. Software and Hardware Configuration

The software configuration consists of James email server, Mailet, and Postgres database. Mozilla Thunderbird is used as a client. Observations Edit and Observations Server web applications also need to be installed through Tomcat. The hardware configuration consists of laptop, wireless router, and iPod devices. The iPod devices send emails to the James Server on a laptop through wireless router, which is connected to the laptop. After this, James server invokes the Mailet, which parses the email and saves the data to the postgres database and file to a filed server directory. Next, Observations Edit web application can be used to view, edit, and delete observations on a field server. The application can also save field server data and photos in xml file and zip file. After this, Observations Server web application can upload both files to remote server in order to add xml file to server database and add photos from a zip file to server directory. Next, James, Mailet, and Postgres are discussed, which are parts of software and hardware configuration discussed in this section.

V. Diagrams

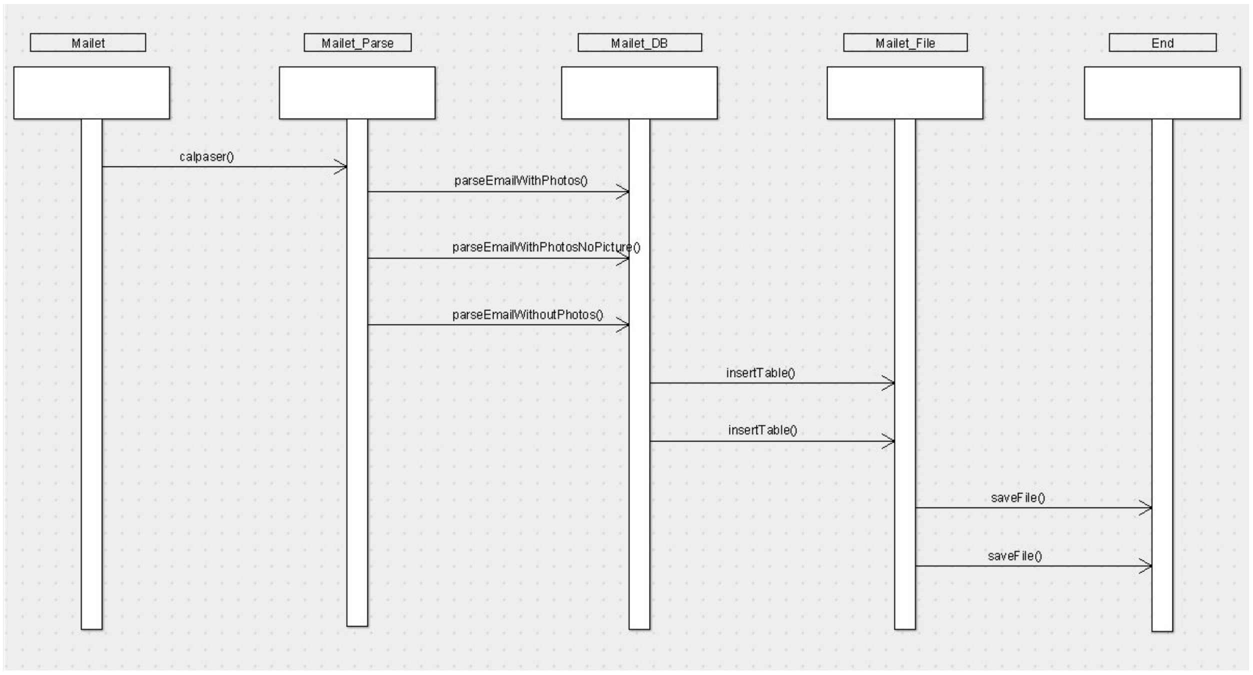
1. Mailet Class Diagram

Mailet Class diagram describes the classes in the application, such as mailet, Mailet_Parse, Mailet_DB, Mailet_File, and Mailet_data. Mailet is the main part of the application. Mailet_Parse is responsible for parsing the email. Mailet_DB works with database. Mailet_File saves the file. Mailet_data holds the data.



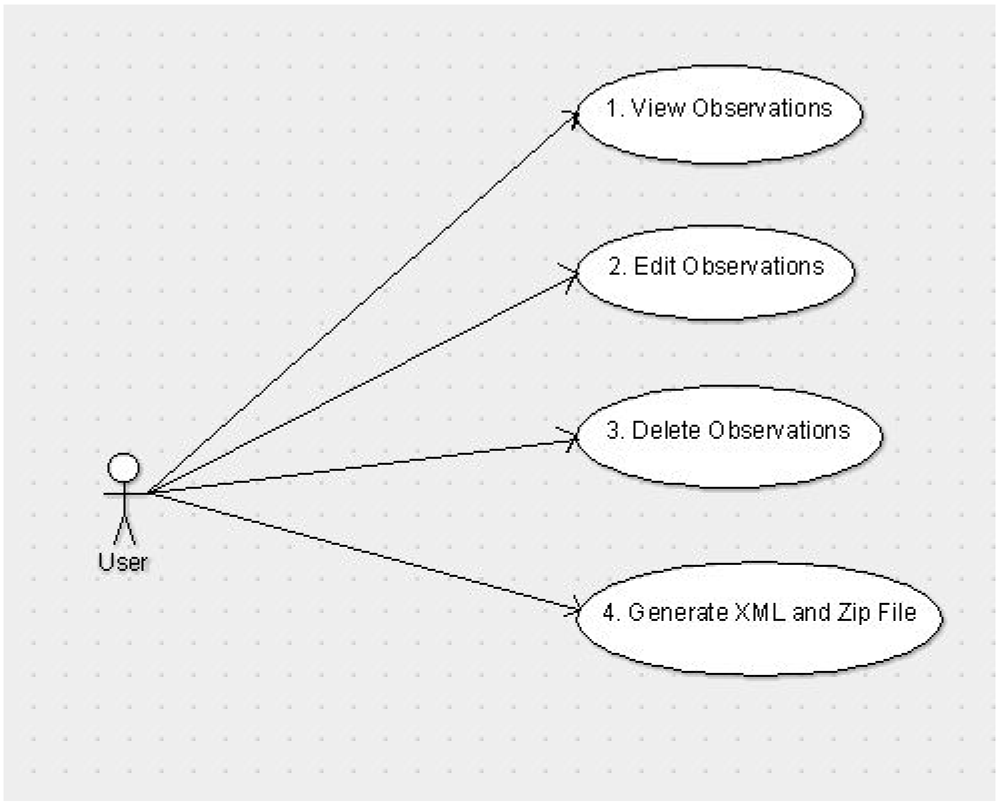
2. Mailet Sequence Diagram

Mailet sequence diagram describes the parts of the mailet sequentially.



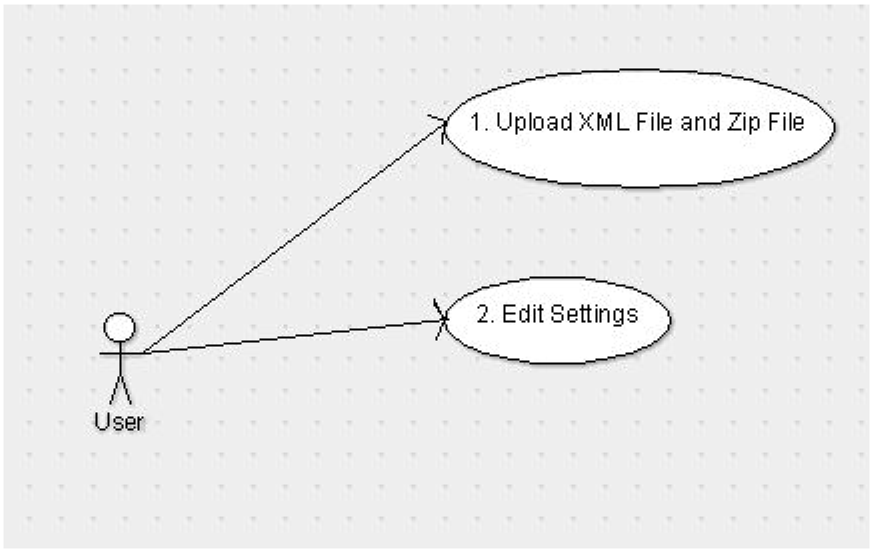
3. Observations Edit Use Case Diagram

Observation Edit use case diagram describes Observation Edit application. The user can view observations, edit observations, delete observations, and generate xml file and zip file.



4. Observations Server Use Case Diagram

Observation Server use case diagram describes the Observations Server application. The user can upload xml file and zip file and edit settings for the archive server.



VI. James, Mailet, and Postgres

In this section James, Mailet, and Postres are discussed. Implementation is done through Mailet, which contains the following features: main mailet, mailet data object, mailet db module, mailet file module, and mailet parse module. When email is sent, James server invokes the Mailet, which puts parsed email data into field server database and email photo into field server directory. Java Apache Mail Enterprise Server is a java enterprise mail server built by the Apache group. [4] It has pluggable architecture and a Mailet infrastructure that works in a similar way for email as servlets do for e-mail and for web servers. It offers new possibilities for the internet. The server enables to construct a message with one or more recipient addresses using a mail user agent (MUA). There are many forms for text-based, Web-based, and GUI applications. The e-mail client is used to send mail to a mail transfer agent (MTA). Also, it can poll an MTA to fetch e-mail messages sent to the users address. E-mail account on a mail server (MTA) using standard Internet protocols is required. It works with e-mail offline (using POP3) or leaves the email on the server (using IMAP). SMTP (Simple Mail Transfer Protocol) protocol is used to send mail from client to the MTA and between MTAs. E-mail servers rely on DNS and e-mail-specific records called mail transfer (MX) records. MX records are slightly different from DNS records, which are used to resolve URLs and contain additional priority information used to route mail. DNS is the key to routing email. James is an MTA, while JavaMail API provides MUA framework. JavaMail can be used to set up a test for James installation.

There are several James design objectives. It is written in Java language in order to maximize portability. [4] It is secure and provides the number of features that protect the server environment itself and provide secure services. James is a multithreaded application, which takes advantage of many of the benefits available for Avalon framework. Avalon is an Apache Jakarta project and features the Phoenix high-performance server infrastructure. James provides a comprehensive set of services, many of which are usually available only in high end or well established e-mail servers. The services are mainly implemented using the Matcher and Mailet API. Standard email protocols are supported (SMPT, POP3, IMAP) and others.

There are several requirements for designing an application. [5] A user can send email to set account to unavailable mode. The message is stored. A user can also cancel unavailable messages. The message is discarded and stored message is removed. Then a user receives notification. When a user receives an email the sender receives a copy of the stored message. Recognizing recipients that have stored unavailable messages is more complicated. Matcher is developed. For efficiency it is important to check whether user is local before testing for the presence of available directory. Other than this the process is straightforward. Unavailable directory is used. Several operations are required, including detecting the presence of a message, saving a message, reading, and deleting. These functions need to know the location of a directory in which a message is stored. James, Mailet, and Postgres parts of project are used to parse email. Next, Observations Edit part of the project is discussed.

VII. Observations Edit

Observations Edit is web application responsible for viewing, editing, and deleting observations on a field server. It also saves field server observations and photos as xml data file and zipped photos file. Observations Edit web application is implemented through jsp, Postgresql, and Apache Tomcat. Next, Observations Server web application is discussed.

VIII. Observations Server

Observations Server is web application responsible for uploading generated xml and zipped photos from field server to remote server. Xml parser is implemented for uploading xml. Xml is converted into database records on the remote server and zipped photos are unzipped into a directory on the remote server. Observation Server web application is implemented through jsp, Postgresql, and Apache Tomcat. In the next section experimental results and analysis are discussed.

IX. Experimental Results and Analysis

In this experiment software running time was measured on several operating systems including Windows Vista, Windows 7, and Gentoo Linux installed on VirtualBox. The additional hardware used was Linksys router, and iPod. Client software used was Mozilla Thunderbird. The software responded appropriately and was consistent. The measurements proved that the software could be a useful tool for collecting filed observation data. In the next section conclusion and future work are discussed.

X. Conclusion and Future Work

In this paper email based server for observing apps is discussed. The main components of the server are James email server and a Mailet. The client email software considered is mainly Mozilla Thunderbird. The server should accept emails in form of a single observation or multiple observations. The project also includes data extraction, storage in Postgres db, simple interface for viewing field observations, and bulk upload of field observations. Among the topics discussed are tutorial guide, system architecture, software and hardware configuration, and experimental results and analysis.

XI. Bibliography

[1] amphibiaweb.org, "AmphibiaWeb" N.p. Web. 11 May 2013. <http://amphibiaweb.org/>.

[2] Bukhin, Vladimir, Parks Observer Server and Web Application. San Francisco State University. Department of Computer Science. Culminating Experience Report. SFSU-CS-CE-13.01. 2/27/2013

[3] calflora.org, "Calflora". N.p. Web. 11 May 2013. <http://www.calflora.org/>.

[4] Duguay, Claude. "Working with James, Part 1: An introduction to Apache's James enterprise e-mail server." Developer Works. N.p., 10 Jun 2003. Web. 30 Jan 2013. <http://www.ibm.com/developerworks/java/library/j-james1/index.html>.

[5] Duguay, Claude. "Working with James, Part 2: Build e-mail based applications with matchers and mailets." Developer Works. N.p., 10 Jun 2003 . Web. 23 Feb 2013. <http://www.ibm.com/developerworks/java/library/j-james2/index.html>.

[6] ebird.org, "eBird" Audubon and Cornell Lab of Ornithology. Web. 11 May 2013. <http://ebird.org>.

[7] eol.org, "Encyclopedia of Life" N.p. Web. 11 May 2013. <http://eol.org/>.

[8] inaturalist.org, "iNaturalist" N.p. Web. 11 May 2013. <http://www.inaturalist.org/>.

[9] Kim, Sunyoung, Jennifer Mankoff, and Eric Paulos. "Sensr: Evaluating a Flexible Framework for Authoring Mobile Data-Collection Tools for Citizen Science." . N.p., Web. 7 Oct 2013.

[10] leafsnap.com, "leafsnap.com" N.p., n.d. Web. 9 Mar 2013. <http://leafsnap.com/>.

[11] Missouri Botanical Garden, N.p. Web. 9 Mar 2013.
< http://www.missouribotanicalgarden.org/>

[12] PlantNet, "PlantNet" . N.p. Web. 9 Mar 2013. <http://www.plantnet-project.org
/page:tools?langue=en>.

[13] Shark Observer. “Shark Observer.” N.p., n.d. Web. 14 Sep 2013.
<https://play.google.com/store/apps/details?id=com.wSharkObserver&hl=en>.

[14] Si, Xuyuan, Server for Dichotomous Key & Field Data Collection Application. San Francisco State University. Department of Computer Science. Culminating Experience Report. SFSU-CS-CE-12.07. 4/25/2012

[15] SquirrelMail Administrator's Manual. Squirrel Mail. SquirrelMail, 19 04 2012. Web. 23 Feb 2013. <http://squirrelmail.org/docs/admin/admin.html>.

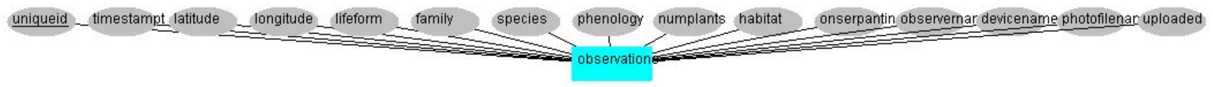
[16] The Electronic Field Guide (EFG), efg.cs.umb.edu. N.p., n.d. Web. 9 Mar 2013.
<http://efg.cs.umb.edu/efg/>.

[17] Wiggins, Andrea. "Free As In Puppies: Compensating for ICT Constraints in Citizen Science." . N.p., Web. 7 Oct 2013.

[18] WildObs Observer. “WildObs Observer.” iTunes, n.d. Web. 14 Sep 2013.
<https://itunes.apple.com/us/app/wildobs-observer/id309451803?mt=8>.

XII. Appendix

1. ER Diagram



2. Table Structure

```
CREATE TABLE observations (
        uniqueid integer PRIMARY KEY,
        timestampt timestamp DEFAULT current_timestamp,
        latitude decimal,
        longitude decimal,
        lifeform varchar(100),
        family varchar(100),
        species varchar(100),
        phenology varchar(100),
        numplants varchar(100),
        habitat varchar(100),
        onserpantine varchar(100),
        observername varchar(100),
        devicename varchar(100),
        photofilename varchar(100),
        uploaded boolean
);

CREATE TABLE observations_server_paths (
        unzipfolder varchar(220),
        uploadfolder varchar(220),
        tempfolder varchar(220)
);
```

3. Installation Guide

- **Install Apache James server and Mailet**

  Apache James Server can be downloaded and installed from Apache James website (http://james.apache.org/). Server name and Mailet need to be added to the Apache James configuration file. Next is to create a user id on the James server. The next step is to add Mailet jar to James lib directory and to add mail.jar, mailet-2.3.jar, and mailet-api-2.3.jar to lib folder within lib folder. Next the image file location directory is set in the observations.ini file located in James bin folder. This will allow sending emails to James Server and will invoke the Mailet.

- **Install Observations Edit and Observations Server web applications**

  Observations Edit and Observations Server web applications are written in jsp. They need to be installed through Apache Tomcat.

- **Install Postgresql**

Postgresql can be downloaded and installed from postgresql website. (http://www.postgresql.org/)

- **Install jdbc driver**

  The Postgresql jdbc driver needs to be downloaded and installed from postgesql jdbc website (http://jdbc.postgresql.org/). The file name is postgresql-9.2-1002.jdbc4.jar. The file needs to be saved in the apache James main lib directory in order to connect the Mailet to database. Next is to save the file in apache tomcat lib directory in order to connect Observation Edit and Observation Server web applications to database.

- **Install Mozilla Thunderbird**

  Mozilla Thunderbird can be downloaded from Mozilla Thunderbird website (https://www.mozilla.org/en-US/thunderbird/). Mozilla Thunderbird needs to be configured with IP addresses and ports for the James server. An email address also needs to be provided. Mozilla Thunderbird proved to be a useful email client.

- **Hardware installation**

  Hardware installation involves connecting the wireless router and configuring the iPod Touch with wireless router and James Server settings.