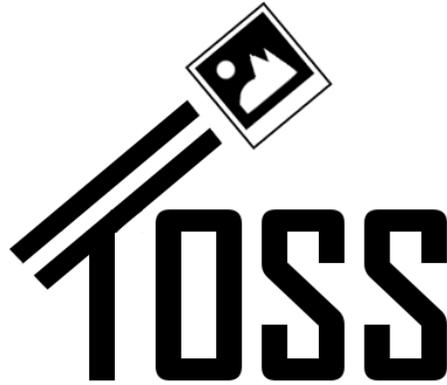


That One Special Shot



Project Proposal

TOSS Team
Gregory LaFlash
Patrick O'Loughlin
Zachary Snell
Joshua Howell
Hao Sun
Kira Jones

Department of Computer Science and Engineering
Texas A&M University

2/10/2014

Table of Contents

1	Executive summary.....	3
2	Introduction	4
2.1	Problem background.....	4
2.2	Needs statement.....	4
2.3	Goal and objectives.....	4
2.4	Design constraints and feasibility.....	5
3	Proposed work.....	5
3.1	Evaluation of alternative solutions.....	5
3.2	Design specifications.....	6
3.3	Approach for design validation.....	10
4	Engineering standards.....	11
4.1	Project management.....	11
4.2	Schedule of tasks, Pert and Gantt charts.....	13
4.3	Economic analysis.....	14
4.4	Societal, safety and environmental analysis.....	14
4.5	Itemized budget.....	14
5	Appendices.....	15
5.1	Product datasheets.....	16
7	Bios and CVs.....	16

1. Executive Summary

It has never been easier to take photos at public events. Yet ironically it remains difficult to gather all the photos taken at them. To this end, our group will create “That One Special Shot” (TOSS), an application which will facilitate the collection and distribution of photos. Using TOSS, users will be able to take a photo at a specific event and upload it in real time to an online, crowd-sourced photo database. After the event, TOSS will allow them to download the photos in the database for free.

In chronological order, TOSS will function in the following manner:

1. An event's host will create an account on our website. For each created event they will be provided with a public QR code. This QR code will be linked to a private Event ID (a unique, 25-digit number identifying the event in our system).
2. The host will then share the QR code with the event's attendees. Anyone with access to the QR code will be able to upload photos to our system for a period of time specified by the event's host.
3. While at the event, attendees will take photos. TOSS will automatically upload the photos to the server specified by the Event ID.
4. For a period of time after the event, only users with the Event ID can access photos on the database.

Downloading photos presents its own problems: in many scenarios a user will not want to access all of the photos taken at an event (imagine a football game where tens of thousands of photos might be taken). TOSS provides two solutions to this problem. First, the user will be able to access the photos in a gallery format. He can therefore download only those specific photos he appreciates. Second, TOSS will allow users to tag specific photos. This way, a user can search for only those photos which include certain people in it.

TOSS will also implement a number of security features to maintain our user's privacy. First, only those creating events will need to create accounts. As long as an event's attendees have access to the Event ID, they will not need to create an account with us. This way, they can maintain their privacy, even with our system.

Second, as photos come into our website, their metadata will be stripped so specific photos cannot be tracked to a specific photographer. Our website will also add a photo's tags into its metadata so photos can be searched. These tags are to be kept internally, and not shared with anyone. When the user downloads the photos, all tags are stripped.

Third, we will limit access to the tags. Only those with accounts can do searches by tags.

2. Introduction

At any large gathering – whether it be a wedding, convention, or sporting event – attendees will want to both take photos and see photos taken by others. With the proliferation of smartphones, taking photos has never been easier. Retrieving other’ photos, however, remains cumbersome. Because no one wants to engage in the time consuming task of contacting photographers individually (this assumes, of course, we have their contact information) a way to easily create and access a crowd-sourced photo database would be beneficial. The proposed application, upon completion, will be such a database. It will give users a variety of storage and access options as well as include numerous security features.

2.1 Needs Statement

Current software which attempts to create these databases are woefully inadequate: Once our project has been completed, none will come close to having its range of functionality. Our application will be distinct from current applications in three major ways.

First, storage comes with a variety of options. Current applications only support one type of storage, severely limiting the consumer’s choice and thus making it less likely that the application will suit.

Second, our program will implement numerous measures to ensure and maintain our users’ privacy and security. Our research into other applications has shown that, as of yet, this is not even a passing concern. However, as internet users become more concerned with their privacy online, any form of social media (including crowd-sourced photo databases) must take this concern into account.

Third, and perhaps most important, our application will be free. Any application currently on the market which comes close to offering our range of functionality is either inherently for profit or must charge the users’ a fee to recoup the money needed to store all of its users’ photo.

2.3 Goal and Objective

We plan to create an application which will support a crowd-sourced photo database. It will provide various methods of storage, privacy, and security for free. Using our app, anyone with the proper permissions will be able to upload a photo they have taken in real time to one of the numerous databases our application will support.

In order to keep the program free, our method of storage must be free as well. To do this, TOSS will leverage the APIs of a number of social media websites, which will service as our makeshift databases. These social media websites include Facebook, Photobucket, Twitter, Instagram, Pinterest. Users will also be allowed to store their photos using Dropbox, an FTP server, or private server that they must create and host.

2.4 Design Constraints and Feasibility

Our primary constraints derive mostly from the fact that it is our aim to keep our application free. For our application to be free, our storage must be free, and to keep our storage free we will leverage the APIs of social media websites. It is here where we will encounter the most constraints. Our constraints will fit into two categories: documentation problems and media throttle limits.

First, because the APIs for these social media sites change so quickly, proper documentation is often not kept. Many tutorials are therefore out of date and useless. Second, many APIs have method throttle limits, which restrict how many times we can call a function of the API. For instance, Photobucket will only allow our application to upload 10,000 photos a day to a specific album.

3 Proposed work

3.1 Evaluation of alternative solutions

Before starting our project, we have done a lot of research on alternative solutions so that we can choose the best idea to satisfy our goal. Although there are currently many software which can allow users to download or upload photos in storage none of them have advantages in terms of application, privacy, and expense compared to our Toss.

Firstly, “Wedding Snap” is one of the applications to collect all guests’ wedding snaps in packages. This software can be used in both Android and IOS systems and it is free. However, hosts need to purchase one of three packages which cost at least \$129. Since Wedding Snap has already had its own package, hosts cannot use other storages such as Facebook, Dropbox or private servers.

Second application is “Sharypic” that allows for smartphone-to-app photo uploading and sharing. Also, it can collect photos from other storage such as Facebook and Dropbox into just one application. Besides, users can choose the free version for some small events. For large events such as weddings or graduation ceremonies, the cost is as low as \$9.99 per event.

Third alternative solution is “Bonfyre” which also has both IOS and Android applications. Users are not only able to share photos, but also they can create and join albums, and enter private chats so that they can talk about the event. The chatting feature of Sharypic can provide users with privacy since people must participate in the chat before they can view the photo. Also, users can mute the notification of chat to prevent any interpretation during the time people are viewing photos. This application is free. However, it does not support a variety of storages like private servers.

Another solution is to use “Fotojelly” that is mostly used for importing the details of a person’s Facebook events and let them add photos separately from the social network. People also can share the final albums privately. Besides, users are able to fine tune their photos by using editing tools before they want to share them. Even though the application is free, the disadvantage is that it can only be used in Windows 8, and it is basically not an event photo collection software.

The last application we surveyed about is called “Napa”. This application has better security since it allows users to share photos privately among friends instead of hosting them publicly on a website like Facebook. It will automatically display photos taken at the same time. The software cost only \$0.99, but it can be only use for IOS system.

Based on these alternative solutions, we decide to create “Toss” since “Toss” have more benefits. Firstly, it comes with a variety of options such as Facebook, Dropbox, private server and so on, but other applications will only support one of these. Second, we will implement measures to maintain our users' privacy and security. This is not even a passing concern for other applications. However, as internet users become more concerned with their privacy online, this is something we must take into account. Third, our application will be free. Any application currently on the market which comes close to offering our range functionality is either a) inherently for profit or b) needs to recoup the money needed to store all users' photos.

3.2 Design specifications

TOSS will be comprised of five main parts: an Android phone, two web services, a storage medium, and a database. The Android phone will be the main method of interaction for users. The first web service will be the other method of interaction for the user (frontend), and will control user account creation, login, and photo displays. The second web service will be strictly backend-oriented, handling incoming photos, authenticating device IDs and event IDs, and communicating with the frontend. The storage medium will be decided by the user at the creation of each event. Storage mediums may be Photobucket, Dropbox, Facebook, or a user's private server. The database will store user accounts, event IDs, and links to photos.

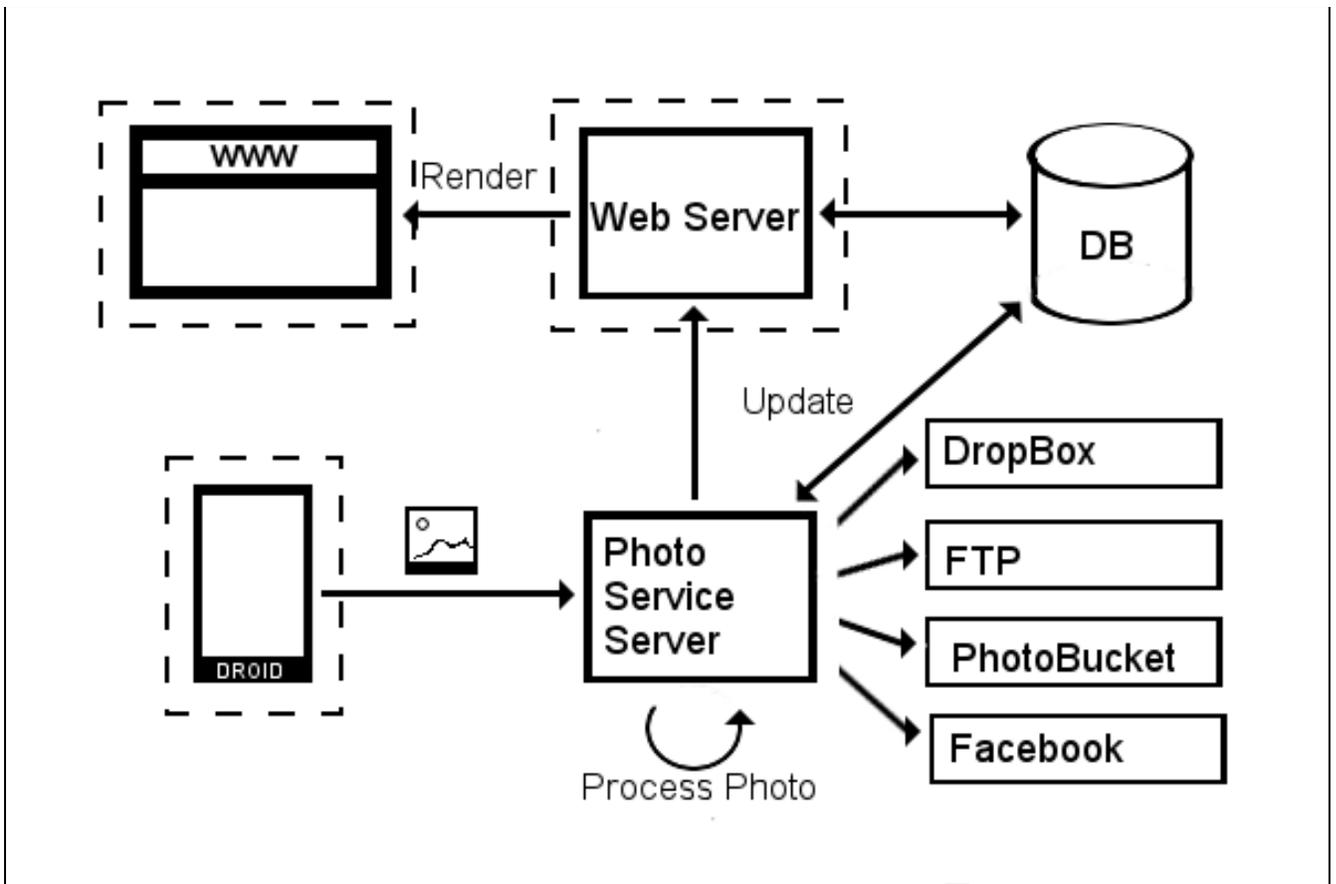
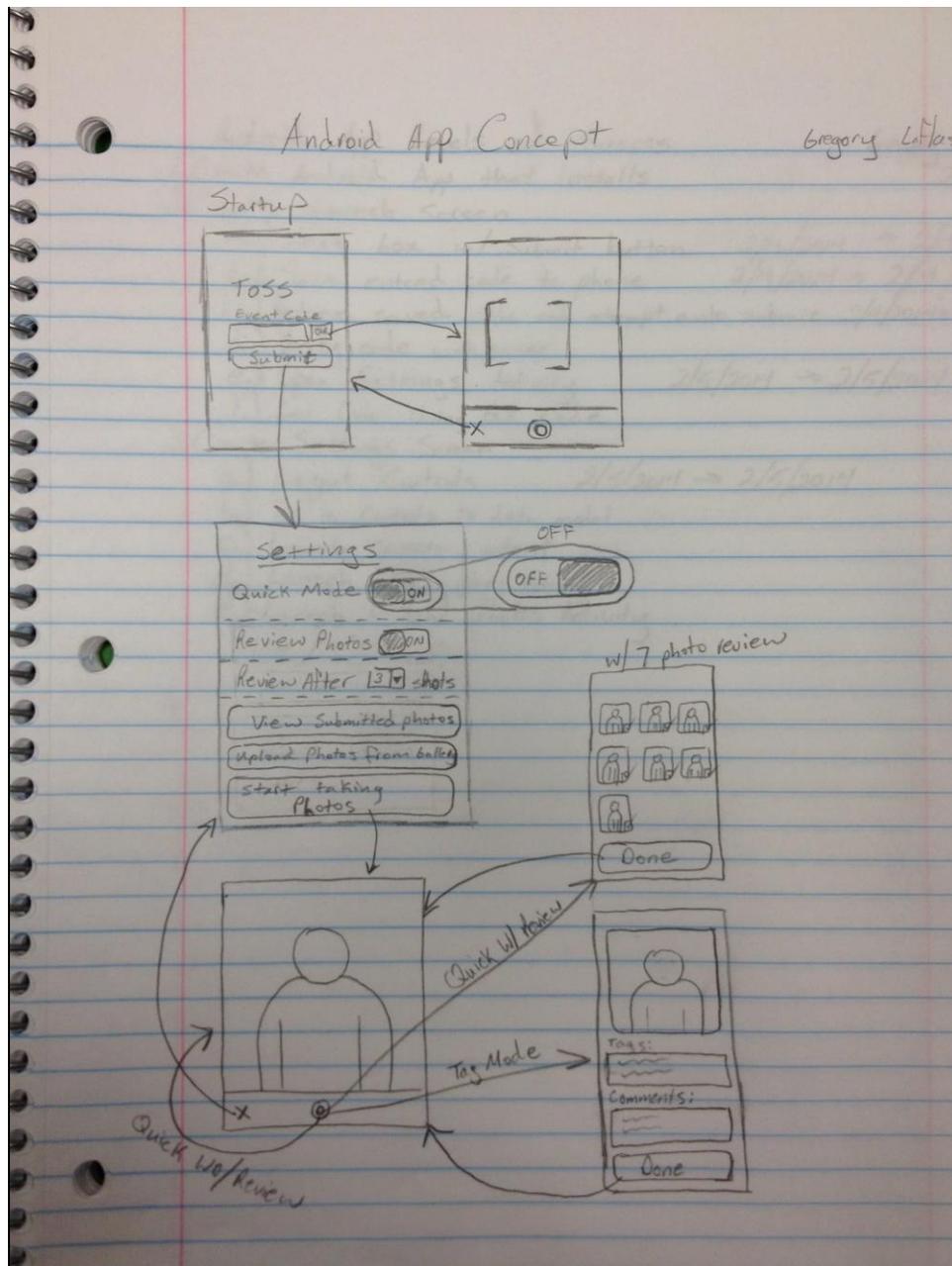


Illustration of the basic System Architecture. Note the two separate web services.

The Android application will be developed by Gregory. The app will consist of the default camera component, a screen to input an event ID code, and a screen to choose which pictures should be sent to the web service using REST. The Android application will send each photo as a REST call, with image data, tags and comments, device ID, event ID, and time taken present.



Gregory's first draft design of Android application user interface.

The frontend web service will be built from Python and the Django Web Framework. It will handle the creation of user accounts and events, the display of photos, and the downloading of all pictures taken at an event. Kira will handle the technical design of the web service frontend. Hao will be in charge of 'branding' the website and creating a uniform look and feel throughout the project. The Python web service will interact with the database to manage account and event creation, view photos, and retrieve these photos. Event creation will assign a unique identification code to each event, generated through an md5 algorithm. The host of an event will distribute a QR code to his guests. The QR code will translate into an event code, which maps to the unique event ID.

Hao's basic layout for the website.

(The other parts of the site are basically the same format as the above pictures. From the top with function buttons to the bottom. In the middle part, users can input their personal information)

The photo service (backend) will be implemented in Java and Jersey. Jersey is a framework for creating REST applications. Maven will be used to manage dependencies and builds. The web service will be created using Grizzly, a lightweight Java server. Patrick will be working on the photo acceptance and event ID authentication. Josh will be managing the storage of photos in the user's Dropbox, Facebook, or Photobucket account. The decision for splitting the photo-handling service and the user interface service was made by Gregory, to reduce load on the UI while many pictures were being uploaded. This service will communicate with the Android application, database, and Python service. The Android app will send the image, device ID, and event ID, which will be read by the Java service. We will look up the event ID in the database to determine its authenticity. The device ID may be logged to prevent spammers from abusing our system. After the event ID has been verified, the photo will be uploaded to a storage medium and the Python web service notified of changes.

The inclusion of several different storage options for the user allows for more flexibility and privacy than other services may allow. Josh will be in charge of integrating our web services with Facebook's, Dropbox's, and Photobucket's APIs. The list of planned storages is as follows:

- Facebook
- Photobucket
- Dropbox
- Private Server

The standard services (Facebook, Dropbox...) will be simple to set up. The host for the event will provide his credentials to the specific account to allow our services to upload the guests'

photos directly to the account. The private server option is a more technical option for the truly privacy-conscious user. This option will allow the users to bypass all of our services. Instead of giving his guests an event ID, he will distribute a link to his private server, which will collect and display photos sent to it.

The MySQL database will contain information related to user accounts, event IDs and time constraints, and photo links. Zach will set up and maintain the database. The Python web service and the Java web service will communicate with the database. The Android apps will never interact directly with the database.

Zach has purchased a server from NFOservers.com and installed 12.04 LTS Ubuntu distribution. Unicorn, Engine X, Supervisor, and Virtual Environment are installed on the Linux machine. Most of our development work will be done through this server. Both the Python/Django service and Java/ Jersey service will run on Zach's server.

3.3 Approach for design validation

The overall system must have several different points tested. Technical points as well as user interaction points must be evaluated.

- Android application usability
- Python web service (frontend) usability
- Python web service response time
 - Under normal conditions
 - Under heavy load on web service (many users browsing existing event albums)
 - Under heavy load on photo service (many users uploading photos)
- Java web service response time
 - Under normal conditions
 - Under heavy load (many users uploading photos)
- Database security
 - Are usernames and passwords stored properly?
 - Are unauthorized users prevented from viewing photo details? or event details?
 - Are the event IDs secure and being recycled correctly?
- Are the users' Facebook, Photobucket, and Dropbox account credentials secure?

Obviously, the most important feature to test would be the basic functionality of the system. Does it perform the task it is meant to do? Our team will take pictures with several Android phones simultaneously and upload them using an event ID. We will test uploading them to Facebook, Photobucket, Dropbox, and a private server. If this test passes, we shall test uploading photos concurrently to multiple events.

Usability testing will be performed by polling a large portion of the CSCE 482 Senior Design students. Students will have a chance to vote on a Lickert scale just how usable the interface is, in addition to providing a short description of what they found intuitive and what was difficult. This method for usability testing can be done for both the Android application and the main website.

Load testing will be organized by measuring response time of both web services as well as resource allocation of the server while many users are uploading photos. The large number of users will again come from the students of CSCE 482. Multiple users will upload multiple photos to different events while we record response time of the services.

Security testing will be accomplished through attempts to break into our system. Examination of permissions concerning the database will be crucial.

4 Engineering standards

4.1 Project management

Project management will consist of the team leader creating the overall architecture for the system and breaking it down into tasks for each team member. Each team member will break down their large tasks into smaller tasks. Continual progress checks at meeting will be performed in order for the team lead to keep track of the status of the project and make adjustments as needed. Implementation decisions are based on group discussion. We will use Gantt charts, version numbering and checklists to ensure that we continually have a working system and keep development on schedule. We will meet on Mondays and Wednesdays from 4:10 to about 6:30 and additional meetings will be planned on a per need basis.

Gregory

Responsibilities

- Project lead
- Develop the Android app

Qualifications

- Over a year of experience of working at a software consulting company
- Android app Experience
- Experience building multiple web applications

Patrick

Responsibilities

- Handle the interfacing with the Android app from the web service side.
- Handle the interfacing with the website front end web service.
- Live viewing of photos

Qualifications

- Interest in graphics, which will be needed in creating a custom live preview of the photos

Joshua

Responsibilities

- Handle saving photo saving to all storage options
 - Handle Retrieving photo using APIs

- Social media integration
- Creating the schema for what information from the photo will be saved and how it will be saved

Qualifications

- Interest in working with the various APIs needed to save the photos to various storage options such as Dropbox, Facebook, Photobucket and others

Hao

Responsibilities

- Branding the website
- Creating the website excluding the parts specifically tied to a backend process.

Qualifications

- Experience with website design

Zachary

Responsibilities

- Database design
- Schema for event code generation and maintenance
- Purchasing the web server hosting

Qualifications

- Full time employee at Texas A&M - CIS - Software Developer
- Created and maintained web applications based on Django
- Experience in Database maintenance
- Already had a hosting account and has experience with web server maintenance

Kira

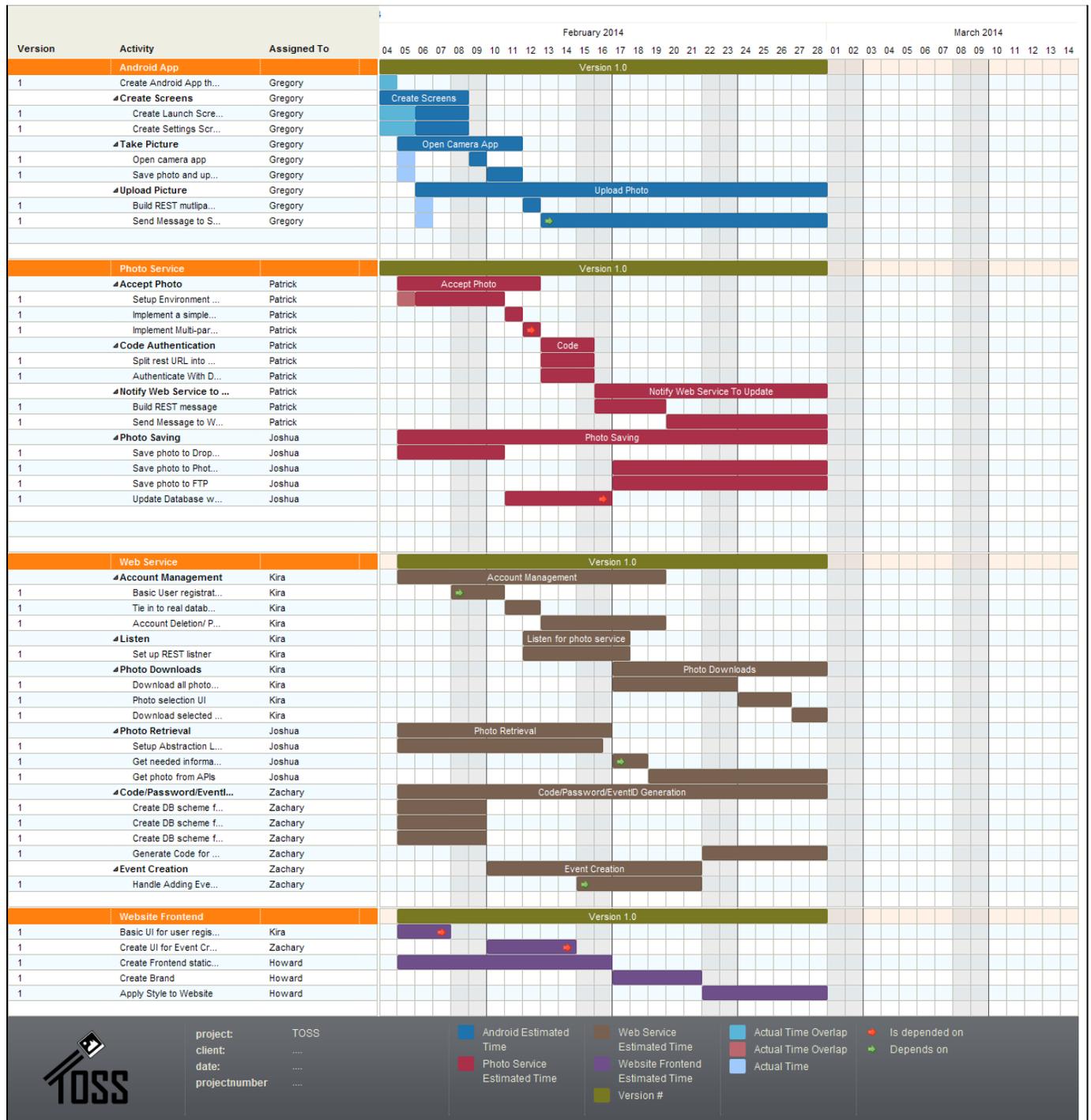
Responsibilities

- Handle the interfacing with the photo upload web service.
- Creating the schema and implementing user accounts
- Handle both the frontend and backend needed to retrieve and download stored photos

Qualifications

- Experience developing a Django web app

4.2 Schedule of tasks, Pert and Gantt charts



4.3 Economic analysis

- Economic viability:
 - The marketability of our system has already been shown within the public in the use of this style of system for, specifically, weddings. In addition to offering the use of this application for weddings, we are allowing the use for any event. In this perspective we can offer everything that currently exists, plus improvements. We can monetize any area we desire but are choosing to offer many things for free. Things that have been shown to be monetizable are products such as cards which include the codes pre-printed, which we will offer a template for free but can sale the actual printed cards if desired. Additionally, we can offer links to various services which return portions of the profit observed from the link. The area of marketability for webapps/applications is very high along with the marketability of photo printing on various objects.
- Sustainability:
 - Overall the only physical aspect of our service is a server. Servers are widely available and easy to obtain. In the area of software, everything we are using is highly maintained languages and api's which are unlikely to be removed and are easy to maintain. The survivability and sustainability of this application overall is very well structured.
- Manufacturability:
 - The tolerance of our application should be high as the products and services that are being used are, in general, already well tolerated. The use of photos and sharing of them will have various privacy options with well documented usage notes. The software is easily testable since the only testing necessary is usage, this should be easily sourceable. Any regulations surrounding photos and their privacy, usage, market use, and other concerns will be covered in the use policy of the application and service upon registering or downloading.

4.4 Societal, safety and environmental analysis

We are creating a product for use by the everyday person at event gatherings. The project has the potential to have a large and varied user base. Our hopes is that our product will have a beneficial impact on its users. It is our intention that the product will be used to consolidate event attendees by allowing them common access to uploading and downloading event photos. The product will allow event attendees the ability to share and connect with each other through photos. Although it is not our intention, the product could also be detrimental to event attendee privacy.

As with all large publically-accessible online services, the privacy of individuals involved is a primary concern. The product has the ability to breach the privacy of all event attendees. By attending the event, each attendee can potentially be captured on photo and shared with all other attendees or even those not physically attending the event if they have been given proper access codes. Additionally, the service in which the event creator designates as the storage facility leads to other privacy concerns not directly tied to our service. We aim to mitigate potential privacy

violations by providing a variety of privacy options to both event creators and attendees. We allow the event creator the ability to choose between a variety of photo storage services, specify access permissions for each event code generated and also the ability to strip meta-data off all photos taken with each code. The event attendees will also have the ability to strip their meta-data off every photo they upload.

The service also has the potential to be detrimental to both the psychological safety of the attendees and the safety of the event creator's storage service account. The photos can be uploaded and shared with any individual containing the access codes which can be detrimental to the beneficial aspect of the product. We are primarily concerned with malicious users who intend to upload inappropriate photos not suitable to the event and users who flood our photo services throttling other users from using our service or by utilizing all the storage available in the event creator's storage service account. Also, some photo storage services may have policies regarding the content of the photos which could result in the deletion or banning of the event creator's account.

Our product is completely software based and therefore has no direct impact on the environment or natural resources.

4.5 Itemized budget (1 page; 3 points)

- 6 Months of Virtual Server from NFOServers \$52.45
- 3 Months of Tomsplanner Subscription (Gant charts) \$27.00
- Possible Server Upgrade Varies
- Final Party to Test Application Negotiable

5 Appendices

5.1 Product datasheets

1. *Linux only Single core*

- One **full, dedicated** HT CPU core (Nehalem or better)
- **1024 MB** of RAM
- **100 GB** of RAID-protected storage
- **1500 GB** of InterNAP bandwidth transfer
- Restricted to Linux installs

6 Bios and CVs

Zachary Snell is a soon to be (December 2014) graduate of Texas A&M with a bachelors of science in Computer Science. He has experience with engineering including 2 years of classes within nuclear engineering. Additionally, he has worked with Texas A&M for 4+ years. Here he works as a software and applications developer, developing web applications for various functions within the university system. In addition to this, he has worked in the support of applications with thousands of users involving active enhancements and developmental staging. He also has experience with game development under an indie company as well as various courses. He has a passion for game development and may pursue a future within this. He has extensive experience in C++/C, Java, C#, Python, PHP, SQL, and HTML/CSS.

Zachary Snell

zach20snell110@gmail.com

Mobile: (325) – 280 – 1973

Current Address

400 Marion Pugh Drive Apt#2130
College Station, TX 77840

Permanent Address

9205 Fm. 2228
Baird, TX 79504

OBJECTIVE

Seeking active development position in advanced Software or Game development projects.

EDUCATION

Texas A&M University – College Station, TX

Major GPA: 3.73

Overall GPA: 3.04

B.S. Computer Science

December 2014

RELEVANT EXPERIENCE

Texas A&M University: Computing and Information Services – College Station, TX

Student Worker: Software and Applications Development

May 2012 – Present

- Participated with a programming team implementing advances and repairing ongoing bugs with a problem tracking system, Keystone.
- Inspected ongoing requests and current issues with application and instated advancements with through PHP, MariaDB, and Python.

Student Worker: Help Desk Central

May 2010 – May 2012

- Worked in Technical Assistance through over the phone assistance to Faculty/Staff and Students.
- Provided physical and software based computer repair services for Faculty/Staff and Students.

- Developed working advancements in a USB-Flash-Drive deployment of operating systems and technical software.
- Learned valuable interpersonal communication skills by collaborating with students from many fields.
- Trained and created documentation over various operations.

Archetype Global – Fort Worth, TX

Internship: Game Development

June 2013 – August 2013

- Worked in Unity development to create an Indie game from crowd-funding towards final production.
- Small Team environment helped to better interpersonal skills and team focused programming.

HydroGraphics by Design – Cross Plains, TX

Web Development Associate

May 2009

- Worked in a small team-based environment designing and implementing a basic marketing website.
- Maintain website/server and instate changes as requested by owner.

SKILLS

- Programming Knowledge – PHP, Python, SQL, C++, Java, Haskell, django, HTML, CSS, C#, UnityScript.
- Operating Systems – Linux, UNIX, Windows, administered web server.
- Skills Advancement – Self trained PHP and UNIX, implementing a reporting solution on schedule. Fast learner.
- Writing Skills – Concise, organized reports, complete documentation and release notes.

Kira Jones is a student at Texas A&M University who will be graduating in May 2014 with a bachelor of science degree in Computer Science with a minor in Business Administration. She currently works for Texas A&M University as a peer teacher and a student worker under Dr. Valerie Taylor. She peer teaches several undergraduate computer science courses, helping students understand material and assisting in running labs. She also works under Dr. Taylor in regards to the Center for Minorities and Disabilities in IT (CMD-IT) maintaining the organization website as well as producing workshop materials. After graduation she will be working as a software consultant for Pariveda Solutions starting in August 2014.

Kira Jones

800 Marion Pugh Apt. 409 College Station, TX 77840 | (713) 906-5733 | kmystic524@tamu.edu

I am a Texas A&M undergraduate student majoring in Computer Science committed to developing both my computer programming skills as well as my peers, through peer mentoring, interested in a software development career.

Education

- **Bachelor of Science | May 2014 | Texas A&M University, College Station, TX**
 - Major: Computer Science
 - Minor: Business Administration
 - GPA: 3.844
- **High School Diploma | 2009 | The High School for the Performing and Visual Arts, Houston, TX**

Projects & Course Work

- Implemented a Restaurant Order Management System with a simple character oriented menu-driven user interface in C++
- Implemented Dijkstra's algorithm for finding shortest paths algorithm in C++
- Implemented a small parser and type checker in Haskell
- Implemented a Database class in C++
- Implemented a Keyframe animation, including character posing using inverse kinematics and keyframe interpolation using OpenGL

Skills & Abilities

- **Programming languages:**
 - Proficient: C++, C, Html
 - Familiar: LaTeX, Java, Python, Haskell, CSS, JavaScript,
- **Programs:**
 - Microsoft Office Suite (Word, Excel, etc.), TeX, Adobe Publisher, Adobe Photoshop
- **Art Mediums:**
 - Graphite, Watercolor, Acrylic, Charcoal, Polymer Clay

Experience:

- **Peer Teacher | Texas A&M University | August 2012 – Present**
 - Assist with student's understanding of lecture and assignment material in the following courses:
 - CSCE 121: Introduction to Programming Design and Concepts (C++)
 - CSCE 206: Structured Programming in C
 - CSCE 221: Data Structures and Algorithms (C++)
 - CSCE 314: Programming Languages (Haskell, Java)
- **Student Worker | Texas A&M University | May 2012 – Present**

- Works under Dr. Valerie Taylor, Senior Associate Dean for Academic Affairs, in regards to the Center for Minorities and People with Disabilities in Information Technology (CMD-IT)
 - Maintain organizational websites (CMD-IT, CMD-IT DIR, Tapia 2014)
 - Develop organizational workshop documents and materials
 - Achieve workshop success through data management and communication outlet maintenance
- **Sales Associate | Kohls | May 2011 – October 2011**
 - Maintained sales floor, organized fitting rooms
 - Answered customer's questions about merchandise and advised customers on merchandise selection
- **Billing Clerk | Life Transitions Psychotherapy | September 2010 – May 2011**
 - Prepared and submitted private practice therapist claims to third party insurance carriers
 - Accomplished timely insurance payments through session data management

Awards and Achievements

- Texas A&M Department of Computer Science and Engineering's Industrial Affiliate Program scholarship recipient for the 2012 – 2013 academic year
- Texas A&M Dwight Look College of Engineering Dean's Honor Award recipient for the spring 2011 semester
- Graduated from the High School for the Performing and Visual Arts with honors in art

Gregory LaFlash is a 2014 soon to be graduate of Texas A&M University. He is currently earning his Bachelor of Science degree in Computer Science with minors in Business Administration and Math. He has worked part time at Capsher, a software consulting company in College Station, Texas since January of 2013. He has worked on several web apps using Ruby on Rails and DotNetNuke. He also has extensive development experience with C++ and source control across a team. After graduation he will be going to work for Goldman Sachs.

Gregory R. LaFlash

1501 Copperfield Pkwy
Apt 313
College Station, TX 77845
(817) 313-6659

gregorylaflash@gmail.com

6203 Patriotic Ct
Arlington, TX 76002
(817) 468-2854

Objective I am looking for a Full Time position for May 2014. I would like to be involved in software consulting, financial tool development, business solutions, cloud computing, or automation. However, I am open to other areas as well.

Education Texas A&M University, College Station, TX **Graduation: May 2014**
Bachelor of Science in Computer Science
Minors: Business and Math
Overall GPA: 3.33

Intended Courses Database Design(Spring 2014), Cryptography(Spring 2014)

Experience **CAPSHER Technology, Inc.** **January 2013 - Present**
CAPSHER is a software consulting company. At CAPSHER I work with a team of four to design and create software for a client. This involves interacting with the client, designing software, and writing C++ code.
Word Report Generator
Wrote C++ code that gathered data and created a formatted word document with the data and images.
Created a DotNetNuke website
Created a website using DotNetNuke based on a client specifications, which required creating custom modules and meeting with the client in person.
Created a complete MFC user interface for exporting data
Using MFC, I created an application that can be used to select and export data from the main application. This involved custom drawing, grid manipulation, and tree based data structure manipulation.

Activities **Member of IEEE** **Fall 2010-Spring 2011, Fall 2012-Spring 2013**
Professional Society of Electrical and Computer Engineers and Computer Scientists. They work to improve technology for all mankind.
Member of CCDC **Fall 2012-Spring 2013**
Collegiate Cyber Defense Competition, Competition that tests students' ability to defend a network in a business setting.
Member of TAMULUG **Fall 2010-Spring 2011, Fall 2012**
The Texas A&M Unix and Linux Users Group on campus; talk about and learn features of Unix and Linux systems.
Member of TACS **Fall 2010-Spring 2011, Fall 2012**
Texas A&M Computing Society promotes students to learn and advance their

computing skills through competitions as well as professional development by sponsoring industrial speakers.

Big Event at Texas A&M

Spring 2012

Volunteered for eight hours with a group of students to rebuild a fence for an elderly woman in the community.

Skills

Java	C++
MFC	DotNetNuke
PHP	JavaScript
HTML/CSS	Ruby on Rails
Haskell	Android Dev
Coffee Script	SASS
Ruby	TCP/IP Protocol
XML Serialization	Microsoft Office
Visual Studio	Unix/Linux

Patrick O'Loughlin is a soon-to-be graduate of Texas A&M University. Originally part of the class of 2013, an eight-month co-op with Advanced Micro Devices (AMD) bumped him back to the class of 2014. While at AMD, Patrick wrote several applications in Java, including a frustratingly complex Verigy 93k parser and a program to visualize large Test Flows, which are used to verify CPU integrity and correctness. Patrick has experience with C++, Java, Python, Git, and Linux. He prefers Slackware Linux, because automatic dependency resolution breaks more things than it solves. After graduation, he plans to try the entrepreneurial life.

Cell: (830)-456-3957

Patrick O'Loughlin

200 Marion Pugh Rd Apt

patrickoloutamu@neo.tamu.edu

College Station, TX

Objective

Seeking an after-graduation job related to the field of Computer Science. I am looking for work in programming, systems maintenance, and/or information technology.

Skills

- C++ (6 years of experience)
- Java (5 years of experience)
- Python (2 years of experience)
- Bash (1 year of scripting experience, 6 years CLI experience)
- Perl (1 year of experience)
- Linux Distribution (6 years experience)
 - Slackware
 - Debian-based (Debian, Ubuntu, Linux Mint)

Education

- Texas A&M University (Fall 2009 – Spring 2014)
 - Computer Science Major
 - Major GPA: 3.49 Overall GPA: 3.23

Job History

- Advanced Micro Devices (Co-op Engineer) (January 9, 2012 – August 15, 2012)
 - Product Development Group – Automated Test Equipment Team
 - Was the developer for FlowViewer, an internal Java program used by AMD Engineers worldwide to visualize Test Flow programs for new CPUs.
 - Part of the initial three-man team developing the Offline Tester Emulator, an internal Java program designed to emulate Credence Sapphire and Verigy 93k CPU test equipment. I wrote the Verigy interpreter, which would build a Test Flow structure after parsing Verigy's proprietary filetypes.
 - Wrote GUI program for batching STIL compilation jobs. The program would use the company's LSF Grid to quickly and automatically compile STIL Test Vectors.
 - Helped move the Product Development Group to Git

- Wrote most of the internal Wiki documentation for Git usage
 - Wrote Python scripts to collect users' SSH keys for authentication (Gitolite)
 - Wrote Bash and NSIS install scripts to ease installation of third-party Git modifications (Git Flow) for Unix and Windows
- City of Fredericksburg (Information Technology Intern) (May 20, 2013 – August 9, 2013)
 - Worked as an intern alongside the two other members of the IT Department
 - Reimaged computers with Windows 7
 - Repaired broken PCs, scanners, and printers
 - In charge of city-wide upgrade to Microsoft Office 2010
 - Helped migrate city's ArcGIS information from Shapefiles to Geodatabases
 - Mapped the city's network infrastructure using Microsoft Visio

References

- Russel Immel (Information Systems Manager at City of Fredericksburg)
rimmel@fbgtx.org
- Lea Feuge (Engineering Technician at City of Fredericksburg)
lfeuge@fbgtx.org
- Justin Schwartzbeck (Cisco)
justinmschw@gmail.com
- Jacob Petersen (Hewlett-Packard) (<http://jacob-petersen.com/>)
jacob@jacob-petersen.com

Hao Sun is a student who will be graduating in August with a bachelor degree in computer science and secondary major in applied mathematics. I am interested in software development and algorithm design. I have worked on several projects which involves database implementation, game design, android application, network security, parallel programming and so on. I am familiar with C, C++, Java. After graduation, I plan to work in a software company to gain more experience.

Education

2011-2014 Texas A&M University College Station, TX

- Expected Degree: Bachelor degree in Computer Science and applied mathematics
- Expected Graduation Date: August, 2014
- Current GPA of 3.35

2008-2010 Lone Star College Houston, TX

- Associate of Science
- Outstanding student in Mathematics
- Average GPA of 3.6

Scholarship

2011-2012 International Transfer Scholarship

Experience

Main Presentative

2008-2010 ICX International Culture Exchange Organization Houston, TX

- Hold meetings for Culture Exchange Program
- Organized trips to Austin in spring break
- Help international students adapt to American culture

Math Tutor

2006-2008 Fukien Secondary School Guan Tang, HK

- Educated students in pre-calculus and calculus
- Taught students problem solving skills in mathematics

Waiter

2009-2011 Bei Jing Chinese Restaurant Houston, TX

School Activities

2011-2014 Member of Tennis Club College Station

2011-2014 Member of Texas A&M Computing Society College Station

Skills

Language: English, Mandarin, Cantonese

Programming: C, C++, Java, Matlab, SQL

Joshua Howell is currently a senior at Texas A&M University with a major in computer science and a minor in business. During his time at A&M, he has done graphics research with the computer science and visualization departments, written for The Battalion and Maroon Weekly, and volunteered with the Brazos Valley Sexual Assault Resource Center. After graduation, he will relocate to Chicago.

Education

Graduation Date: May 2014

Major/Minor: Computer Science/Mathematics

WORK EXPERIENCE

REU Student, Texas A&M University

June 2012 - August 2012

- Documented compilation processes for graduate student software used in the research process.
- Reduced download and compilation time from days to hours.

Student Researcher, Texas A&M University

January 2012 - May 2012

- Edited and proofread research papers

Columnist, The Battalion, Texas A&M University

August 2011 - February 2012

- Wrote commentary on politics, culture and economics for Texas A&M's student newspaper, The Battalion
- Edited columns for fellow writers on the staff

Journalist and Columnist, Maroon Weekly

August 2010 - August 2011

- Reported on college clubs that did not fit the "Aggie stereotype": Muslim Student Organization, Atheist Agnostic Student Group, etc.
- Wrote commentary on national politics and culture.

VOLUNTEER EXPERIENCE

Agency Volunteer, Brazos Valley Sexual Assault Resource Center

August 2013 - Present

- Made care packages for survivors of sexual assault
- Assisted staff during presentations and health fairs
- Sorted through press clippings for topics

General Worker, Aggie Con 44

May 2012

- Prepared food for guests of Aggie Con 44, a local gaming, science fiction and fantasy convention.

Concessions, Cepheid Variable (Student Group)

- Helped raise money for Cepheid Variable by working a concessions during at Texas A&M basketball games.

LEADERSHIP EXPERIENCE

Advertising Officer, Texas A&M Computing Society
Fall 2012 Semester

- Created advertisements for the organization's meetings

Program Book Officer, Aggie Con 45
September 2013 - March 2014

- Contacting businesses and selling ad space in the Program Book
- Create and edit the Program Book which will be distributed to guests
- Create the schedule for the convention