­

**Team 2  
Project Honeycomb**

Project 1

Sprint 2

June 11, 2013

by

Alex Veit

Damon Heard

Karl Kamdem

Paul Gimou

A project report submitted for

Software Systems Engineering

Summer 2013

Department of Computer Science and Software Engineering

Southern Polytechnic State University

Marietta, Georgia

# Acknowledgements

**Damon Heard**, for enabling the website layout

**Karl Kamdem**, for implementing critical database functionality and file structure

**Paul Gimou,** for the hard work and helping with the documentation

**Alex Veit**, for implementing user login functionality and data validation

# TABLE OF CONTENTS

Acknowledgements 2

TABLE OF CONTENTS 3

1. Introduction 4

1.1 Executive Summary 4

1.2 Project Goals 5

1.3 Cycle GoalS 5

2. REQUIREMENTS 7

2.1 Project Environment, Technology, Hardware, Etc. 7

2.2 User stories 7

3. DESIGN 10

3.1 System Architecture 10

4. Management Plan 16

4.1 Planned Assignments and Schedule for Current Cycle 16

4.2 Actual Assignments and Schedule for Current Cycle 16

4.3 Planned Assignments and Schedule for Next Cycle 16

5. CYCLE POSTMORTEM ANALYSIS 17

5.1 Management Plan Post-Morem Analysis 17

5.2 Successes 17

5.3 Failures 17

5.4 Lessons Learned 17

APPENDIX A: Supporting Documents 18

APPENDIX B: Cycle Presentation 19

APPENDIX C: Source Code 20

# Introduction

## Executive Summary

In this executive summary, it will discuss the purpose and the goals for the project. The projectwill go into more details in the next sections. This will show the idea we had when had for Honeycomb and not essentially what was created. A person should have visualization in mind, and think on how they can over exceed expectation, or predict how short they will come from reaching their goal. Therefore, what was created will be im

In this executive summary it will explain the entire project, its purpose and its goals. It will go into further details in the next two subsections. Note that this is the idea that we have in mind for The Honeycomb and not necessarily what was created. One should always have a vision in mind and see how far they can surpass there expectation or how short they fall of it. With that being said, what was actually implemented will be in the Management Plan Section.

The Honeycomb, almost as its name suggest will encompass SPSU's server (eventually) to house online storage places for Students and Teachers. Thus meaning they will be able to upload and files such as docs, pdf, mp3, power points and down the road, video formats. Of course the user will be able to download the files as well to any computer that they choose. This will also become a collaborative site where students can search for and share folders with each other. The teachers will be able to do the same plus set up a folder for his or her specific classes that students may access. In fact with the Honeycomb being user friendly, it will make anyone’s life easier.

Students will no longer have to worry about losing their flash drives and deal with the consequences. Whether it will be from the school computers or their home computers, The Honeycomb will be accessible. Teachers will also feel at ease with the ability to upload files specifically for that entire class without the worry of it not reaching everyone. With The Honeycomb linked with students SPSU account, signing up will be an easy process as well as for Faculty. This greatly improves the odds of an account holder forgetting there password. To get the full grasp of The Honeycomb and it many functions and features, one must use it for themselves.

## Project Goals

The goal of this project is to have these few great features and functions in this version of The Honeycomb. First is the main point of The Honeycomb, uploading files onto our online storage. Without that simple function nothing else will matter. One other feature is having the ability to work on all operating systems. The previous point is very important as to the main operating system used by students and teachers are all never the same. Therefore have it work on Windows, Mac and Linux greatly increases the amount of users and usage it will get. Another is having functions such as copy, cut, paste and delete. Once again this creates familiarity and customization. An email verification system that sends a confirmation to the new users email to ensure only SPSU has access. Administrators will be able to log in and remove anyone not abiding by the terms and condition of The Honeycomb in order to keep a clean environment. Lastly a search function enabled for anyone, which will allow any user to search for another and send a request. This request has the ability to be declined or accepted thus allowing the requestor to see the content of the requested. Having this ability will create the share function, the second main point of the Honeycomb.

## Cycle GoalS

During the first Cycle, we intended to get the lay out off the storage system done, meaning the user will be able to view the actual pages as he or she clicks from link to link. The user will be able create a username choose a password but the email that they use MUST be a Southern Polytechnic State University email containing an edu. The honeycomb will be linked to our hard drive to act as if it were a server/database. We will have the terms and conditions page up, and also try to implement a copy and cut function if we are able to get to it.

After the failure we had with cycle one of not being able to upload a file and save it to a directory we decided to rework. The rework was also due to limitations that were set on the team about having no database. It was in continuance until The Honeycomb could save usernames and passwords to a text file. Therefore the first priority to accomplish during cycle two is reworking our login system to function by having it write(ref) to a text file(ref). At the same time another team member will be working on the upload function since one is not needed by the other. This should get us to where we are supposed to be. Through hard work during this cycle we will use the PHP(abv) language to get every function done with a flat file and if there is time input a database. We will mainly focus on a few core functions at first, dividing the work in order to be effective with our time. With four members the key focus points will be file sharing with one another, email verification to ensure only SPSU(abv) users, meaning one user per account and uploading, deleting and downloading multiple files(not at once). With those functions achieving optimum results, meaning everything works and tested for errors, this version of The Honeycomb will be ready for the real world.

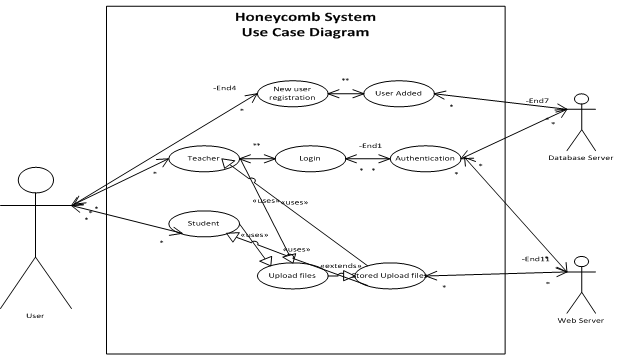
# DESIGN

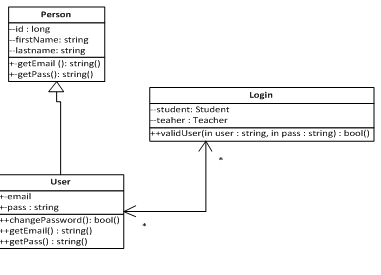
## System Architecture

**Use case diagram**

* The use case diagram of the online storage for The Honeycomb processing system contains the scope of the system, the list of system functionalities and the vision of all the interaction with the system and the actor of the system. The major actors here are:

1. User: User symbolizes the main actor for the system. The Honeycomb offers storage service online for the user in many different locations.
2. Database Server: A file server stores user’s authentications information and it adds new user information when a new user registration begins.
3. Web Server: The server handles new user registration, user authentication, and uploaded files.

**Figure 1: Use Case Diagram**

****

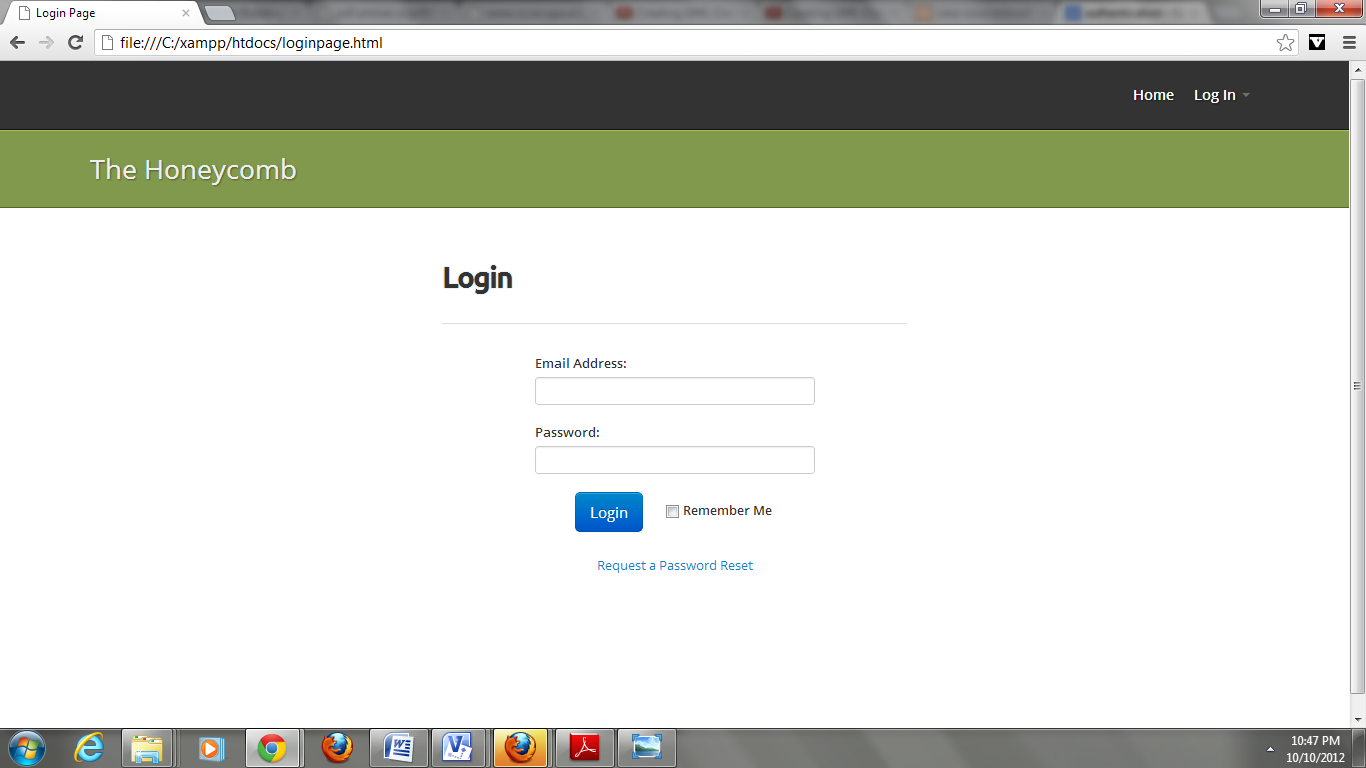
**Figure 2 UML Diagram**

1. **Prototype of the System (GUI)**

* We wanted the new users to better understand the Honeycomb online storage processing system easily. Therefore, it is dynamic webpage users will be able to upload their files from their computer with ease. They also will be able to upload files and save them online and share them to with other selected users. It the system point of user view, we created the header& footer because we felt like we would need them in the entire application. We chose the color and green for background to match SPSU colors.
* We created a big button called “Sign up”, so the user would know where to register for Honeycomb. In the first page we have Home, Login and Signup link button. The home button is to call the home page, Signup to register new user, and Login for the users who already have an account. We used JavaScript to create a drop tab for Login button to call Teacher Account, Student Account, and to Request Passwords. We used CSS from Twitter Bootstrap to improve content accessibility to provide more flexibility and control in the specification. Using the CSS from Bootstrap made the site look more appealing to the user. We ran our GUI on Apache server which supports a variety of features. It allows high performance web servers which provide variety of Multi Processing Modules that gives a better match the demands of each particular infrastructure.



**Figure 3: Home page of the online store system**



**Login page**

Environment setup

1. An operation system as Windows or Linux
2. Apache Server installed on hard drive

Following are the steps required to use Apache:

1. Down the software from <http://www.apachefriends.org/en/xampp.html>.

2. Save it to your local C drive for ease of use

3. Go to your local C drive and find xampp, and open it and enter the folder called htdocs.

4. Save all of your HTML, PHP, and Javascript into this folder.

6. Open a web browser , and type localhost/” the name of your file ” to view your files.

# Management Plan

# CYCLE POSTMORTEM ANALYSIS

## Management Plan Post-Morem Analysis

As a team we all worked hard to develop the features envisioned, but we committed to develop more functionality then the time allowed us. In other words we were unable to implement all the functionality from the user stories.

Our time allocation for each functionality was unrealistic.

## Successes

We have successfully implemented the Honeycomb interface, with signup, login, file upload/download, and sharing functionalities.

## Failures

We were unable to account for storage space for each type of user, e.g., 2GB student and 10GB professor. We were unable to implement password retrieval functionality. The file structure has no depth. We did not implement administrator privileges. We were unable to send out email confirmation due to resource limitations.

## Lessons Learned

Most of our failure was not due to lack of knowledge of the task at hand, but rather the lack of time and familiarity with SCRUM core values. As time passed and we had more lecture hours we begun to meet more frequently, but we still were faced with confusion in terms of time, resource management, back log, and other documentations. Each person worked mostly in isolation making integration nearly impossible.

1. Supporting Documents
2. Cycle Presentation
3. Source Code