1. Executive Summary

1.1 The Problem

Alpine Credit Union is a great institution for my families banking needs. However, for some reason my wife doesn't regularly check our account balances. She frequently asks me what they are looking like, but that generally requires me to stop doing my homework, log into our accounts and sit down with her to go through our monthly expenses. I generally want to go through a much more in depth look into our finances than she does. Also, neither me nor my wife have a checkbook, yet we have a few monthly expenses that we would rather not pay in cash. My wife is frequently uses excel at her job, so she is comfortable with navigating through several Excel sheets.

1.2 The Solution

With one click of the mouse, this Excel Spreadsheet can greatly reduce both our anxieties of going through our finances and taking care of our monthly expenses. One button accesses our bank accounts (recognizing both our passwords and extra security questions) and downloads the financial information for our checking, savings, credit card, and debit card accounts. Then it creates a summary page including our top 10 credit card expenses, our assets, liabilities, and net cash available. Also, with another button it logs into our bill pay account and opens to the exact page to start the payment process. This spreadsheet is meant to give us a "finance at a glance" view of our financial situation. This spreadsheet only saves probably 5-6 minutes, however, it's ability to put me and my wife on the same financial page is worth much more.

2. Overview

This projects main goal is ease of use. On the Main info page there are three buttons for functionality. First the "Update Account Info" button. This button starts the login, retrieval of data, and data manipulation process. 11 sub procedures are used to carry out this task. 6 internet pages are imported and analyzed. This is the bread and butter of this project. Next, we can either pay any bill or specifically pay rent. Both of these sub procedures log into the bill pay sight and get to the user interface quicker than could be done by hand. They skip 4 pages to arrive at the final destination much quicker than someone can by hand.

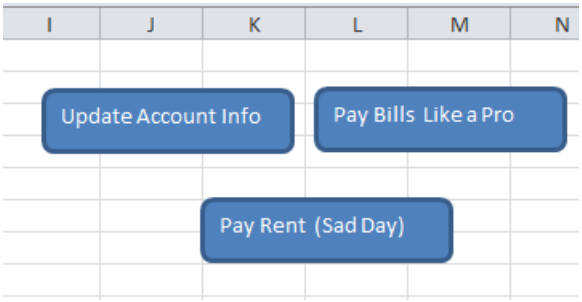

Figure 1: 3 Buttons for data retrieval, bill pay, and rent

3. Project Explanation

3.1 Log into Alpine Credit Union

The process of logging into Alpine Credit Union (ACU) has a main login page and then a secondary security question to make sure the user is really who they say they are.

3.1a Main Log-in Page

The main Login page is a fairly standard login page. First, excel opens Internet Explorer and directs it to "alpinecu.com". I downloaded the HTML file and found the form with "AcctNum" and "Pin" as two of the tags. From a text file stored on my hard drive, Excel assigns both of these the values. See figure 2 (below) with the account and password fields filled out, ready to be submitted.

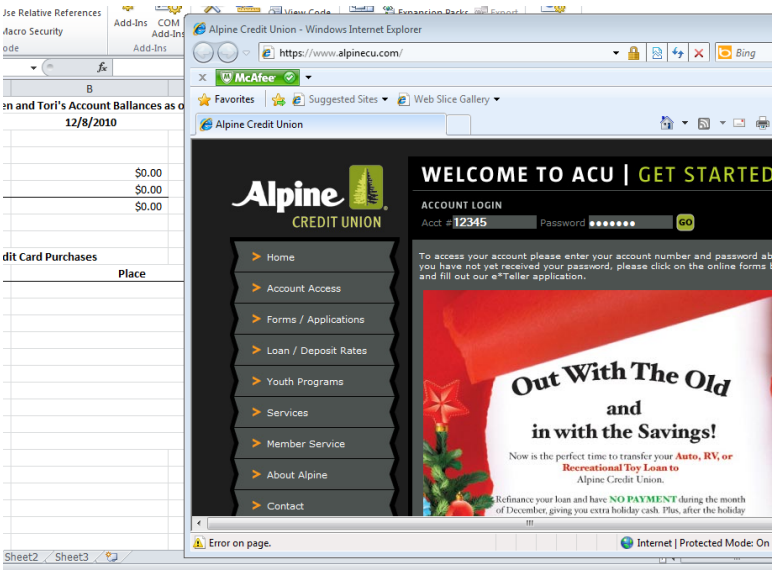

Figure 2: Main Log-in Page for Alpine Credit Union with Account Number and Password Entered

After figure 2, the form is submitted and the first step of our login should be successful.

3.1b Secondary Security Question Log-in

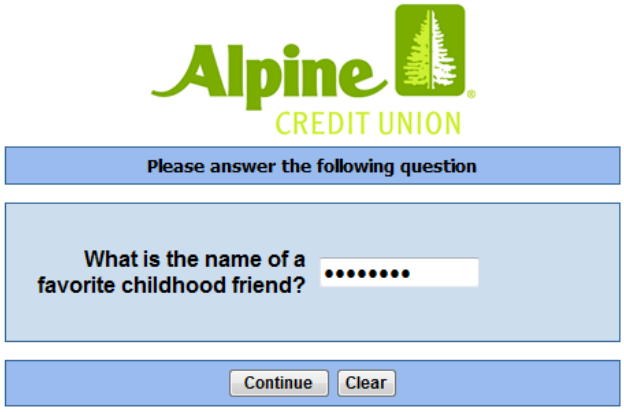

Figure 3: One of Many Secondary Security Questions with appropriate Password

The secondary security question (Figure 3, above) can be one of four questions. Excel imports the picture into a excel sheet and searches through it for one of the 4 possible questions. After it finds which question is begin asked, it pulls the correct answer from text files. After this form is submitted, we have passed the authentication process.

3.2 Downloading Account info

Within ACU's website, there are 4 accounts that I look at; Checking, Savings, Credit Card, and Debit Card. Each has it's own specific url with a 30 day history displayed. This sub-procedure directs internet explorer to each of the account's url's and imports the pages text information. This process is significantly quicker than a regular login experience because excel can skip any redundant pages that a user must click through to get the correct information. A user would have to see at least 9 pages to get the same information while excel only visits 4 pages.

3.3 Bill Pay Sight

The bill pay sight at ACU is quite convenient after you go through up to 6 pages to finally get to the right page. This bill pay skips all of these unnecessary pages. In creating this page, I had a dilemma on my hands. I knew I could automate this process and send checks throughout the country without any major problem. However, I don't think I want an automatic process that could drain my bank account without so little human interaction. I chose to simply take the user to the point of filling out the information, while not automating the whole process.

3.3a Miscellaneous Bill Payment

This sub-procedure logs into our accounts (similar to above, Figures 2 and 3). Then navigates to a page to create a new bill pay session. I'm not entirely sure why, but there is an intermediary page that must be

visited. I believe this is so that you actually have an active bill pay session running. The user can then enter information for a bill, and quickly press send.

3.3b Paying Rent

Rent can be simplified a little bit more than other bills to various procedures. Rent is the only bill that we write every month to the same person for the same amount. This sub procedure goes to the correct bill pay sight, then checks a box (held on the left hand part of the webpage) that holds all the information for rent and auto fills the information. The user can double check the information, and fill out the amount needed, then push the send button. Again, this would not be difficult to be completely automated, however, with actually dollars going out of our account, I felt more comfortable with more user interaction and room for double checking the information.

3.4 Housekeeping and Data Manipulation Sub-Procedures

3.4a Formula Manipulation and Data Consolidation

Specifically when the account info is downloaded through internet explorer, the data is not easily displayed. The Whole purpose of importing the account information was to just give a snapshot of the data that my wife can look at quickly and easily.

	A	B	C	D	E	F	G	H
1	Ben and Tori's Account Ballances as of							
2	12/8/2010							
3								
4	**Assets**			**Liabilities**				
5	**Checking**	$525.15		**Debit Card**	$0.00			
6	**Savings**	$1,025.25		**Credit Card**	$265.00			
7	**Total Assets:**	$1,550.40		**Total Liabilites:**	$265.00		**Net:**	$1,285.40
8								

Figure 4: Summary Information of Different Accounts

Figure 4 is the quick rundown of our accounts with assets and liabilities. These numbers are pulled off the new sheets that where imported. Total assets are our checking and savings accounts combined. The total liabilities are our debit card and credit card balances combined. Our net worth (as far as this bank account goes) is calculated by total assets minus total liabilities.

9			
10	**10 Largest Credit Card Purchases**		
11	**Date**	**Place**	**Amount**
12	11/25/2010	EXPRESS # 0553 OREM UT US OREM UT 23Nov1(	$324.18
13	11/15/2010	CASCADE COSMETIC SU RGEOREM UT UOREM (	$276.00
14	12/06/2010	EXPRESS # 0553 OREM UT US OREM UT 04Dec1(	$121.68
15	11/30/2010	WALKER 21 10020 907PLEASANT GROVUT UPLE	$45.81
16	11/11/2010	WALKERS 12 00310 292LINDON UT ULINDON LI	$43.45
17	11/29/2010	PAPA JOHNS 1283 8017620400 UT US 8017620(	$43.14
18	11/24/2010	MEGAPLEX 8 THANKS GIVLEHI UT ULEHI LEHI UT	$42.00
19	11/15/2010	WM SUPERCENTER LINDON UT US LINDON UT 1	$40.71
20	11/25/2010	WM SUPERCENTER LINDON UT US LINDON UT 2	$38.55
21	12/07/2010	WM SUPERCENTER LINDON UT US LINDON UT (	$37.09
22			

Figure 5: 10 Largest Credit Card Purchases in the Last 30 Days

This information is taken from the imported credit card information. First, the blank spaces between the data must be removed. Then a filter is applied that sorts the data largest to smallest. The 10 largest purchases in the last 30 days are selected and moved over to the information page.

3.4b Deleting Pages

Throughout the process of logging in, downloading information, and manipulating the data, there are a few pages that get created that the user never needs to see. I created a couple subs that get rid of only certain pages. They get used throughout all of the processes to keep the sheets from becoming a mess over time.

3.4c Creating Graphs

Creating graphs in excel isn't difficult, but it can be annoying if it needs to be done by hand all the time. The main graph is our savings account over time. Really we just want to see an upward trend to the data.

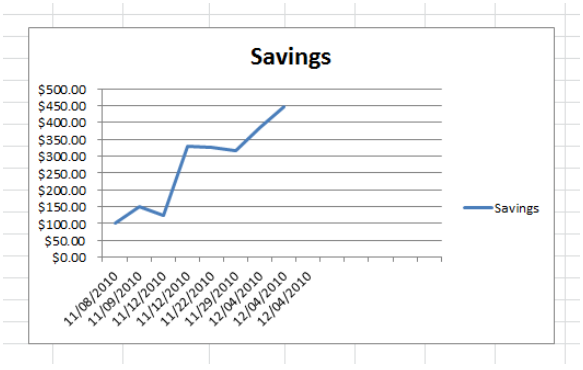

Figure 6: Savings Account Graph (hopefully it grows)

Using VBA this is created new, every time the data is imported. It's probably the quickest way that one can tell what's happening to our savings account, whether it's going up (YAY!) or going down (sad day).

4. Learning and Conceptual Difficulties

4.1 Difficulties

Difficulties abounded while creating this excel spreadsheet with it's sub procedures. Here are a couple that I overcame.

4.1a Internet Explorer Frames

I was completely unfamiliar with "Frames" inside of web pages. They completely destroy the "importPage" method in VBA. When I tried to import a page that had data on it, nothing would show up on the new excel sheet. Finally I found out that the 3 frames that appear are really basically 3 separate web pages displayed at once. If I find the address to a particular frame, that frame can be imported without any major problems. After the fact, this is a fairly easy concept, but it took a long time trying to figure out why the main page woudn't import. If I ran into this again, it wouldnt take longer than 5 minutes to fix.

4.1b Session ID

Each of the webpages after authentication into ACU's sight has a unique "session ID" that changes every time. To overcome this, I needed to find the current URL and extract the session ID. Then whenever I directed Internet Explorer to a new page, I included the session id with it's appropriate placement within the URL.

4.1c Checkbox on Billpay Sight

Ugh, this issue was frustrating. The bill pay sight had so much HTML that I didn't think I could actually find the checkbox. I quickly figured out that the checkbox I was looking for didn't have a "name" property. Finally I figured out that I could find and use the checkbox with it's "ID" using "getElementById". After I figured that out, it was smooth sailing!

4.2 Learning

4.2a Security of Passwords to Financial Institutions

I was initially worried about how I would keep my passwords and information safe. I was worried that someone could get my workbook and immediately access my account information. I think my solution is quite simple and extremely safe. It's interesting that my solution for digital security is a physical one. First, my excel file is on my USB flash drive. The password files that it reads are on the windows partition of my macbook. For example, if someone wanted to get to my account, they would have to steal both my macbook and my flash drive. After stealing both of those, they would have to log into my macbook with one password, then log into my windows partition with yet another password. It was a good learning experience for me to think through my security. I didn't want it to be any easier for someone else to get into my account. I, on the other hand, am frequently working in windows with my usb connected. It's simple and quick for me to access my data.

4.2b HTML Tagging and Internet Explorer Interfacing

Before this project I had never tried to decode html. Now I know the difference between inner and outer html. I can manipulate data both by "ID" and "name." I know how to submit forms, read basic html, check boxes, and navigate through frames. What I learned is so valuable. Excel is no longer just a program, it's my slave!!! (insert evil laughing here) Data doesn't just have to be in a nice little neat file anymore. The amount of data is unlimited because I have access to the entire internet as long as I'm patient enough to create the code to retrieve the information. This skill has already made me more employable.