

# Financial State of the Union

---

**Ryan Allenbach**

## **Executive Summary**

After further analyzing the situation, it became clear that the most important solution would be a time-saving solution.. Discussions with the boss revealed that there was a tremendous amount of time used gathering simple financial information in the effort to understand the “bottom line,” and that is where the idea for this project began.

The household where this project will ultimately reside is no different than many today – there are multiple sources of cash outflow (viz. credit cards) and a similar amount of cash reserves (e.g. bank & checking accounts). With recent legislation passing that effects the consumer credit industry – it is ever more important to be on top of monthly payment. Calendar dates when bills are due can change every month, so the idea behind this project is to create an easy to use tool that automatically gathers the account balances, outstanding charges, and most importantly – due dates of the major credit cards being used. Phase two of the project will mirror the functionality for banking accounts. Ultimately, this tool decreases the number of times during the month when a credit card user visits the online account center.

## **Implementation Documentation**

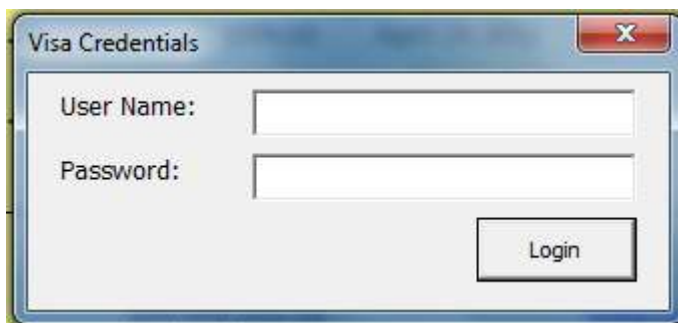
Technology development requires cooperation between the builder of the technology and the user of the technology. During the planning and analysis phase of my project, the most important focus I had was on understanding the high-level requirements would be for the user, and mentally laying out a gap analysis. It was useful for me to consider what functionality

would be in the “future state” of the tool, and what simple wouldn’t be feasible with my growing (but limited) experience and knowledge of VBA.

The actual implementation – or designing, building, and testing – was completed in a very iterative process, where I would build a component, test the component, and then modify the design or build as appropriate. Interestingly, the iterative process really helped me to identify key aspects of the tool that I had originally overlooked. I was also able to identify things that I had learned during class and leverage them as best as I could in order to have a successfully functioning tool.

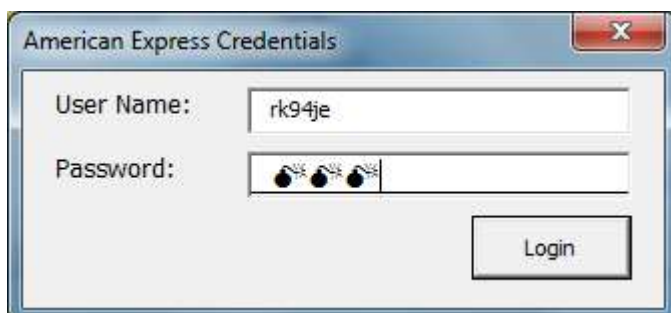
The modules that I developed (and plan to expand upon) are as follows:

- Credential Form – used to gather the User Name and Password. Clearly this is a simple,



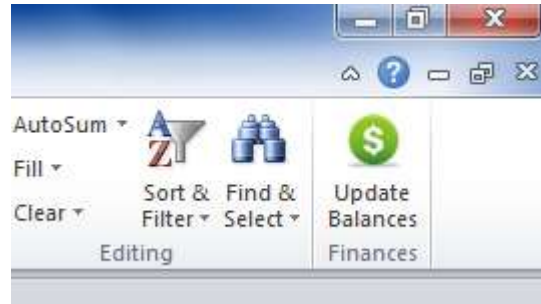
yet absolutely critical component in scraping information from the web – behind internet security measures. It is dynamic, meaning that the form is modified with each connection

made (i.e. “Visa” credentials, “American Express” credentials) in order to best support



the user in ease of interaction. You will do well to notice that the “Password” section has also been modified to protect privacy as well.

- Intuitive Ribbon Icon – any user of Microsoft Suite knows how valuable the Ribbon is in visually depicting the purpose of a function. The user of this program does not have experience with running macros, but certainly values the ease and attractive usability this feature adds. Thanks Dr. Allen!



- A simple summary report page simplifies the diverse information in the colors selected by the user – this most certainly with increase adoption. One of the key factors in a successful product or program launch is the confidence that users have in the system. The user interface will set the tone for an entire opinion, regardless of functionality. From the onset, my determined focus on color selection set the project up for success. Note: actual statement and outstanding balances have been modified to protect the innocent, and to provide a sense for the actual functionality. The Payment Due Date has been allowed to remain the actual date – to create the real effect of ownership and power that comes when using the tool – you are welcome.

## Financial State of the Union

| Account                     | Statement Balance | Outstanding Balance    | Payment Due Date | Link   |
|-----------------------------|-------------------|------------------------|------------------|--|
| American Express - Skymiles | \$5,000.00        | \$9,000.00             | May 1, 2012      | <a href="http://www.americanexpress.com">www.americanexpress.com</a> |
| American Express - Costco   | \$45,000.00       | \$90,000.00            | N/A              |  |
| Visa - Marriott Rewards     | \$0.00            | \$999.00               | April 19, 2012   | <a href="http://www.chase.com/marriott">www.chase.com/marriott</a>   |
| Discover Card               | \$150,000.00      | \$41,018.00            | May 14, 2012     | <a href="http://www.discovercard.com">www.discovercard.com</a>       |
| <b>Total Liabilities</b>    |                   | <b>\$141,017.00</b>    |                  |  |
| ING Direct                  |                   |                        |                  |  |
| Checking                    |                   | \$XX,XXX,XXX.XX        |                  | <a href="http://www.ingdirect.com">www.ingdirect.com</a>             |
| Savings                     |                   | \$XX,XXX,XXX.XX        |                  |  |
| <b>Total Assets</b>         |                   | <b>\$XX,XXX,XXX.XX</b> |                  |  |

Last Update 4/12/2012 3:56:10 PM

- Modules for each financial institution were created with the intent of gathering the appropriate login credentials, and then retrieving the desired information to be manipulated in MS Excel. Currently, American Express, Visa (through Chase Bank), and Discover are the supported credit card companies. In due time, the financial institution ING Direct will have a functioning connection. The connections were built upon the “agent” class variables which were developed by Dr. Gove Allen. The tool has an invaluable aspect of allowing the developer to identify the source code of a webpage as it is currently rendered – liberating him or her from the basic or original source code that is commonly provided by the web browser.
- Finally, modules were written to facilitate the parsing of data deposited into MS Excel. While data may be imported from websites decreases the need for manual data entry and reduces the time required to receive such information, the data is rarely received in

a consistent format (across data sources) and so the need to manipulate it becomes crucial.

## **Learning and Conceptual Difficulties**

The most time consuming challenges I faced in this project have dealt with security. One example that I am still dealing with (and why it is now “phase 2” of the project) is with the security questions asked of the bank when the computer is not previously registered by the user. Additionally, there is a key-pad like utility that accepts either keystrokes or mouse clicks to advance.

Working with [www.chase.com/marriott](http://www.chase.com/marriott) (the VISA card) posed some security measures that were difficult to assess. During the development phase, I successfully was able to enter American Express’ site though trial and error. Finding the “submit” button name was frustrating – but I eventually succeeded. I followed a similar path for the VISA card and because I had tried to log on so many times, the website just refused or blocked any traffic that appeared to come from a computer. I double checked the code – triple checked it. Again and again I would submit my login credentials through the user form and it would attempt to submit them via the internet browser. If I manually moved the mouse and clicked the submit button, my password would appear (from the previously run, and stalled VBA code) and the would work.

As though the clouds parted and the heavens shined down upon me, I remembered a conversation I overheard (maybe even in class) where computers could block requests if they were coming from computers, and the program had to be modified to appear as a human.

Brilliant! Now how could I make it executable. I researched several locations, from the text book to the internet, and eventually found the following solution:

```
Application.Wait Now + TimeValue("00:00:05")
```

Conquering my coding challenge helped me to feel extremely victorious and rewarded from doing the project.

As was briefly mentioned before, the American Express and Visa cards are fully functional; however, the Discover card connection still has bugs, as does the financial institution ING Direct. I continue to make progress, but I find that I come up with a solution that may work only to find out that it doesn't fully meet what I need to do. You may refer to the following sub in my code as an example: Sub findInfo ()

## Assistance

I do not believe that my project is the most challenging or complex of the projects that I have heard; however, I am more satisfied with the results of the project because I completed nearly all of it on my own, using only tools that I received in class. I say nearly, because I spoke with one other person regarding my project – Joel Riddle. He took the VBA class last year and ultimately selected a similar project.

My conversation with Joel was about his approach and how he ran into problems with websites that were not compatible with the agent functionality. While I did not encounter that problem, I was aware of what I may encounter. Also, because of the conversation, I feel that my mind

was beginning to plan for a certain series of events that I otherwise would have had to develop and react to on my own.

Ultimately, my interactions with Joel have been insightful, but have not contributed directly to my code. I feel a lot of satisfaction knowing that I was able to manipulate the Agent class effectively, and that includes finding the appropriate tags and ID's of the web objects that needed to be modified. I look forward to addressing the open questions that have arisen from the financial institution that I am working with and am pleased to see my tool being used.