Comp - u - Lock

Parents achieve control

Documentation Written By:
Colten Rouska

Index

Executive Summary

Comp-u-Lock is a Windows application that is designed to keep parents in control of their children's computer use with a straightforward interface. With this accessible interface, information about what data is being used and when it is being accessed is clearly visible. Comp-u-Lock allows for the logging of application use and the ability to lock certain processes from running. This program also allows for set time periods of allowed computer use, set by the parent, locking when this allotted time has expired.

Project Description

Recently, Microsoft has released a Windows feature called "Family Safety". Family Safety was released in 2007 with many features that allowed parents to control their children's' computer activity. This program was a core feature of Windows Vista and 7. The software allowed for controls that limited each child's computer use for a set amount of time. The application could also be controlled through the web, which allowed universal access to settings and time limits. The downfall to this application was that there were no backwards compatibility before Windows Vista, making the program unavailable to 60% of the Windows market that was still using Windows XP.

Children are fighting with their parents to be on the computer for longer periods of time. Parents no longer have a recourse for action other than to be "the bad guy" and say no. In today's world, software has been able to solve problems within a vast range of topics. Very few programs have been able to satisfy child obedience with parental control.

Comp-u-Lock is a solution to this problem. This program builds upon Microsoft's Family Safety, adds Windows XP support, and adds a reward system. The reward system is present to support parents. After this project's completion, Comp-u-Lock can be improved with additional features that will satisfy the web portion of the requirements.

Comp-u-Lock creates a marriage between the desktop application and the web service. This allows the application to synchronize settings when internet access is available. The program enables user control from locations other than onsite, and the program manages all the computers associated with the account simultaneously.

Project Requirements

For Comp-u-Lock to be a successful solution, it has the following requirements:

- Authenticate users
    - Using an email address
        - Can not be 0 characters

        - Should follow valid email format
        - Max of 30 characters
    - Using a password
        - At least 6 characters
        - At most 30 characters
        - Can contain alphanumeric and '@', '?', '!', '.', and '#' symbols
- Limit computer users account access based on time
    - Time constraints are set by the user
        - Users can select which days of the week the account can be accessed
        - Users can select time ranges within a day of the week that the account can be accessed
        - Total amount of time a user can be on during the above constraints are specified via time tokens
    - Users can be issued time tokens
        - Time tokens are issued in the amount of 5, 10 and 15 mins
        - Time token amounts can be modified
            - Time tokens cannot hold a value less than or equal to 0
- Supports displaying information for the user to view
    - Logged account information per account
        - Processes ran
        - Websites visited
        - Programs ran
- Encrypt and decrypt sensitive data
    - Using SHA 256 encryption

External Specifications

The Comp-u-Lock interface is meant to get information to the user in an easy and efficient manner.

When the user first tries to use the application, the login screen will appear. The user would then input their credentials and login. If the user did not have an account, they can click the signup button near the bottom of the login window. This will open up a registration window.

The registration window is used for inputting user credentials to create an account. Once the user has inputted their desired credentials, they will click on the giant signup button near the bottom of the window.

The overview window is the the window that is displayed after the user has successfully logged into the system using the provided login screen. This window displays general information about all the accounts on the computer. An overview of processes, programs ran, and websites visited are displayed near the bottom of the window. Graphs above these areas give a visual representation of the usage for each user. The one near the left of the window displays the total

amount of time the user has been on the computer. The next graph on the right displays the general usage during a set time period.

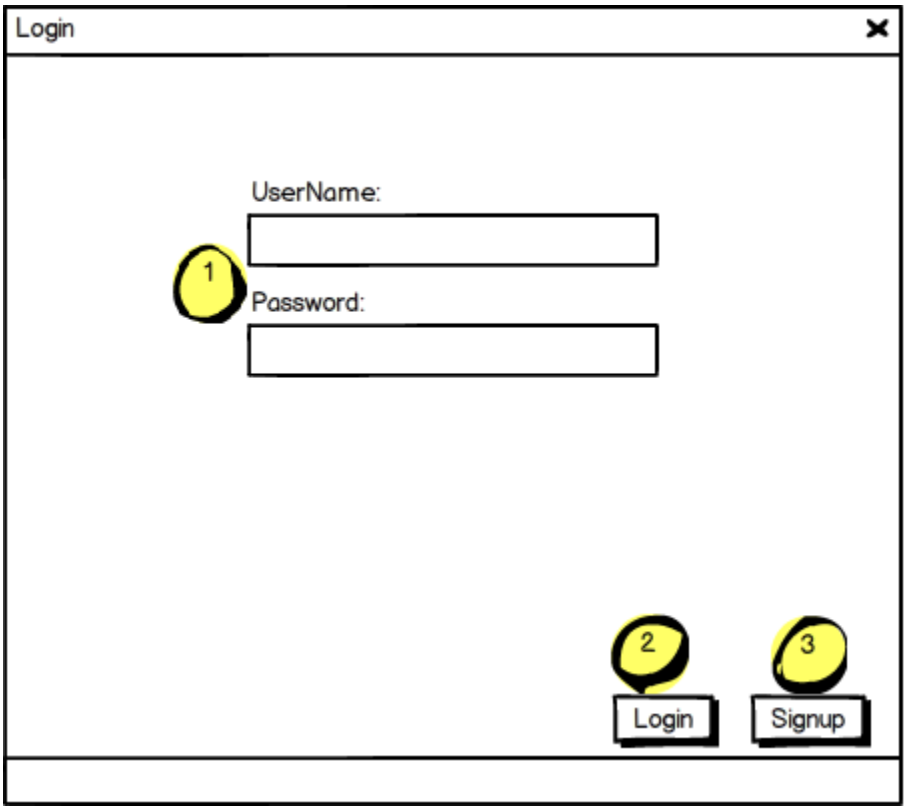
The logs window displays a lot of detail specific to a single user. In order to do that, the user has been provided a drop down menu to select a specific account. There is an area to show the user how many time tokens have been used and how many are left on the account. Much like the overview window, three sections display the processes, programs ran and the websites visited in the past 24 hours. The 24 hour range can be set using date picker interfaces so the user can look at other time ranges. There is also a display that shows usage in a linear fashion.

Next on the list is the hours window. This window displays an interactive chart and a drop down menu. The drop down menu is there so the user can select a computer account. The interactive chart allows the user to specify days of the week and the times for each of those days that the account can be accessed.

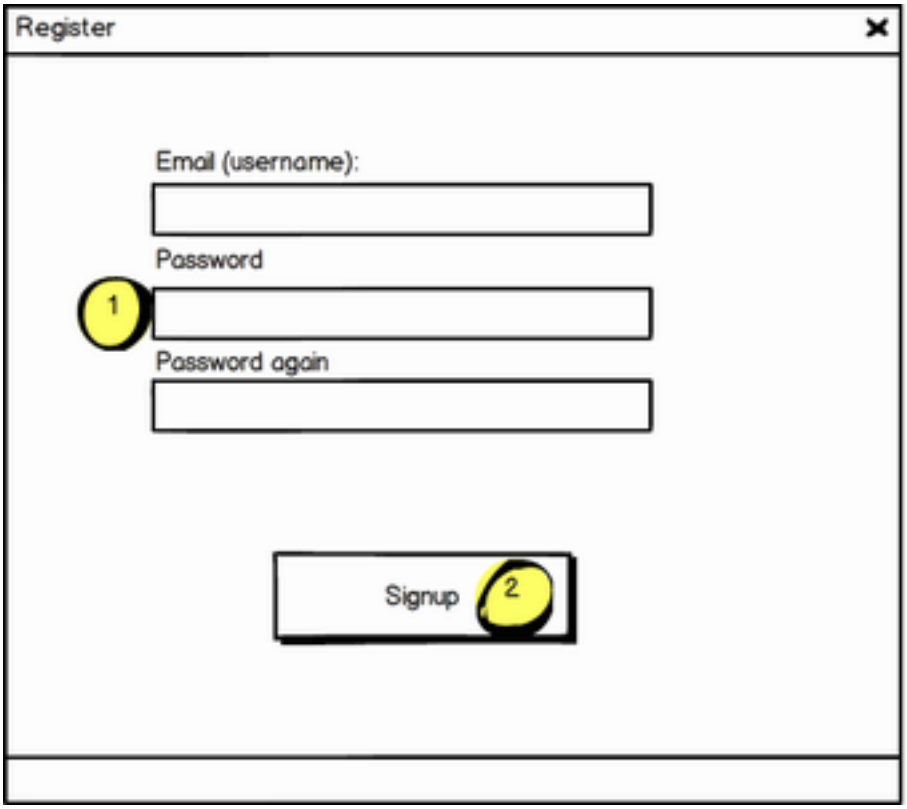
Last of all, the settings window. The settings window allows the user to constrain the application. This window provides information such as the account that is currently managing the computer. The user is able to change which account is associated by inputting their credentials in the provided field. The user is also able to specify what kind of information should be saved via each account. Last of all the user has the option to select which accounts should be tracked and managed.

Login and Register Screens

Login screens are primarily used for authenticating and registering a user.



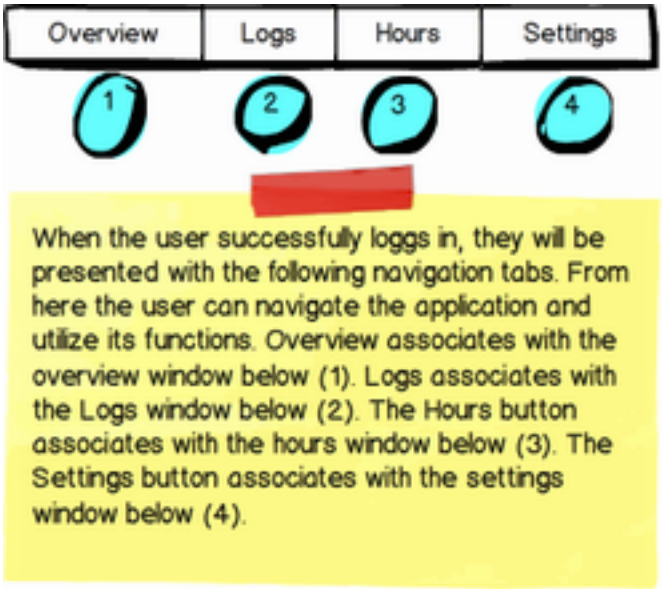
When the user clicks to open the application, the login screen displays. An example of the window is displayed to the left. The user can then enter in their web credentials to log into the system(1,2). If they do not have an account, they can click the sign up button below.(3)



After the user has clicked the sign up button from the login screen, this window will display. This window provides the user to register their account if one does not exist. The user inputs their username which is their email, and their password twice. After which they click the signup button to finish the registration.

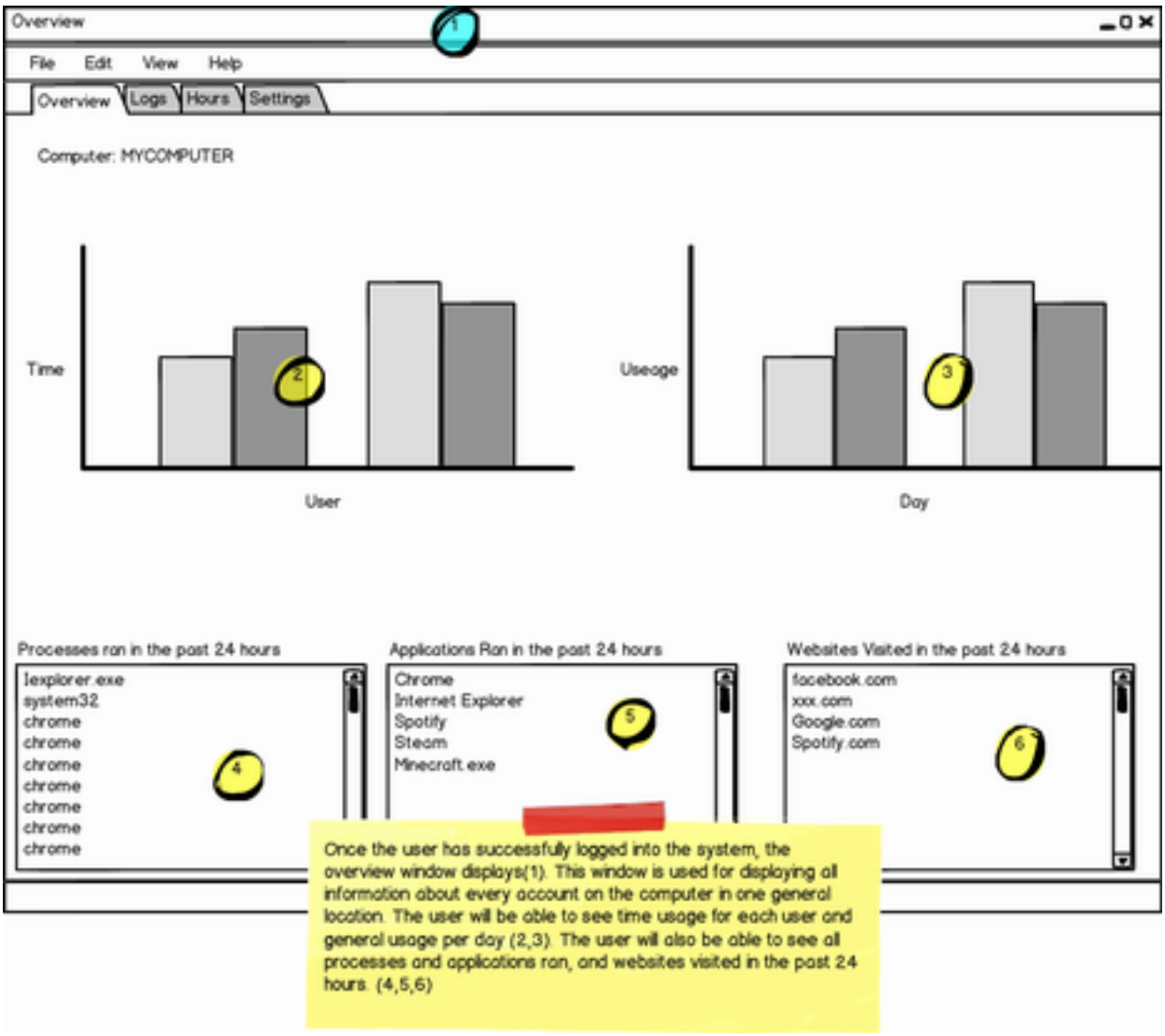
Navigation

The navigation is near the top of the main window which is represented in the overview, hours, logs, and settings windows. These tabs are the main flow for navigating the application.



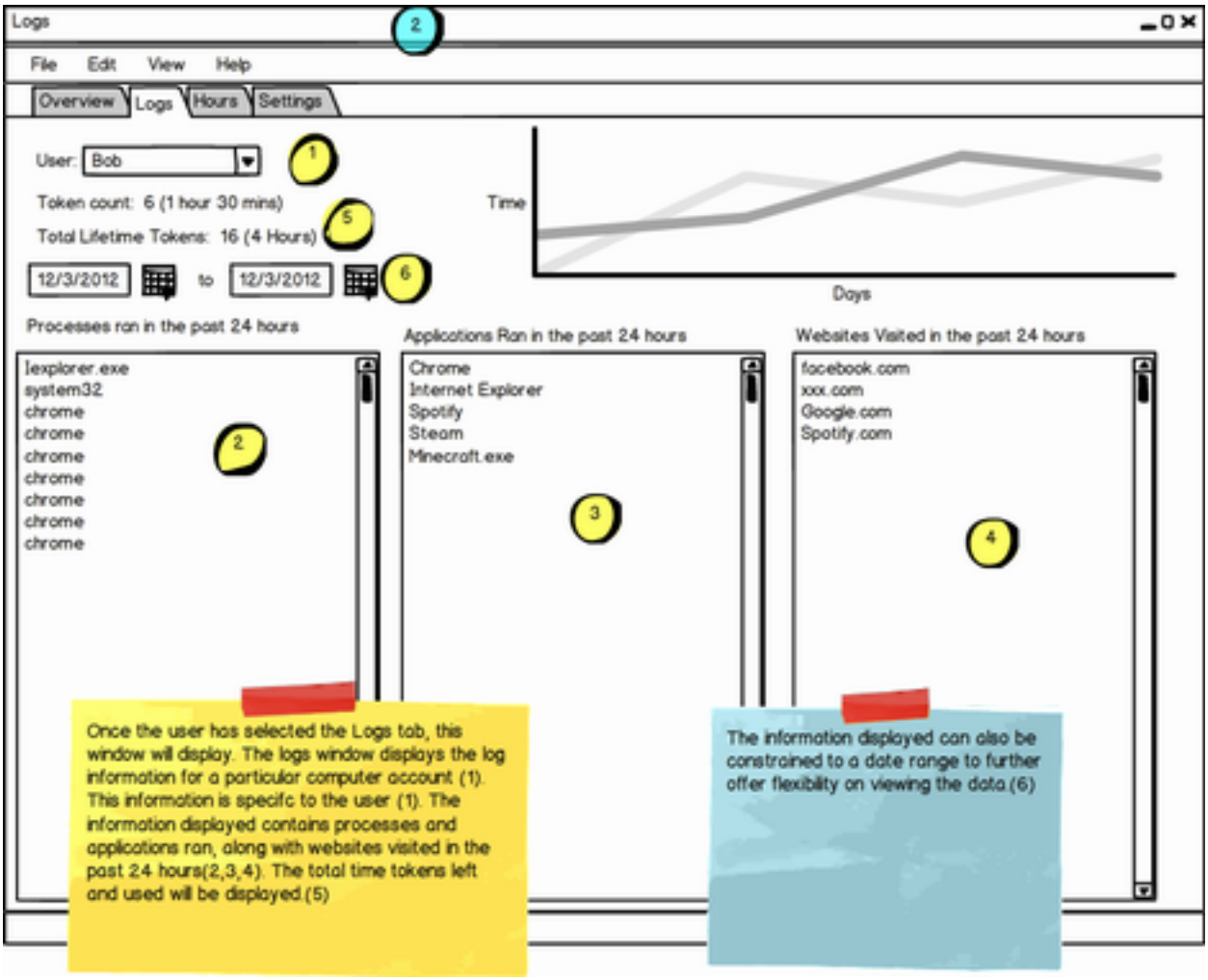
Overview Window

The overview window shows general information about all the accounts currently being tracked.



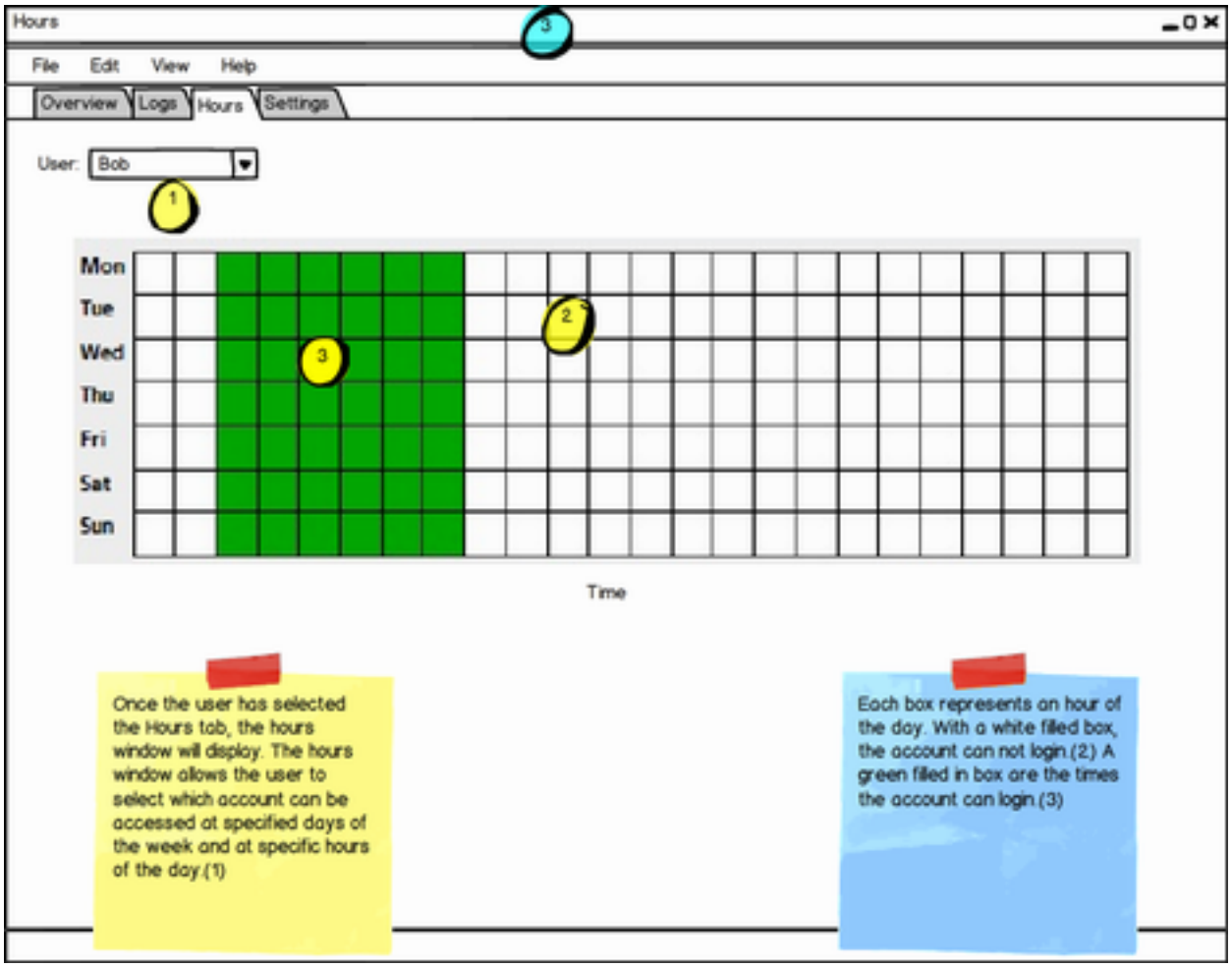
Log Window

The log window displays specific information about a specific user account that is being tracked.



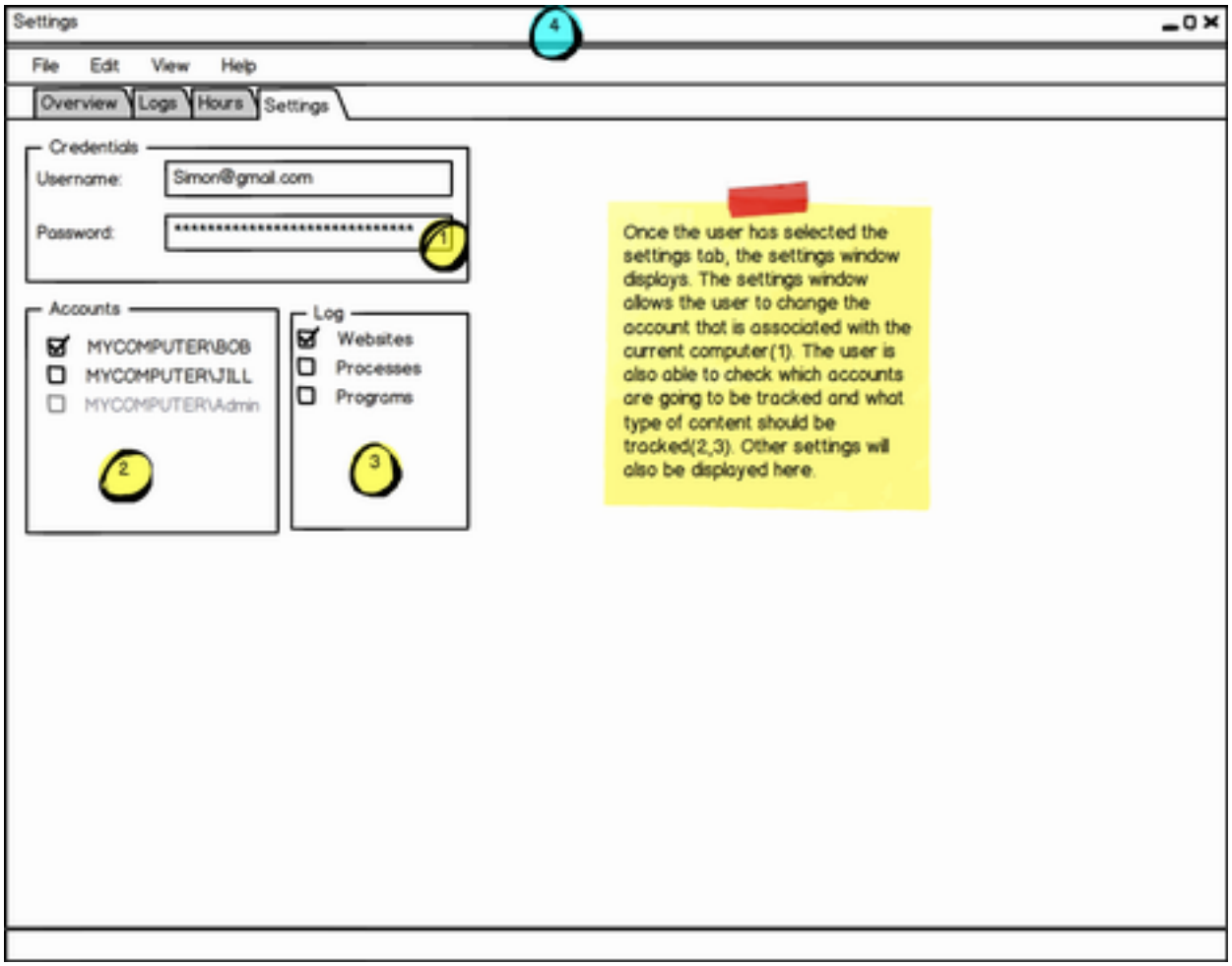
Hours Window

The hours window allows the user to set up constraints on when the user account can be accessed.



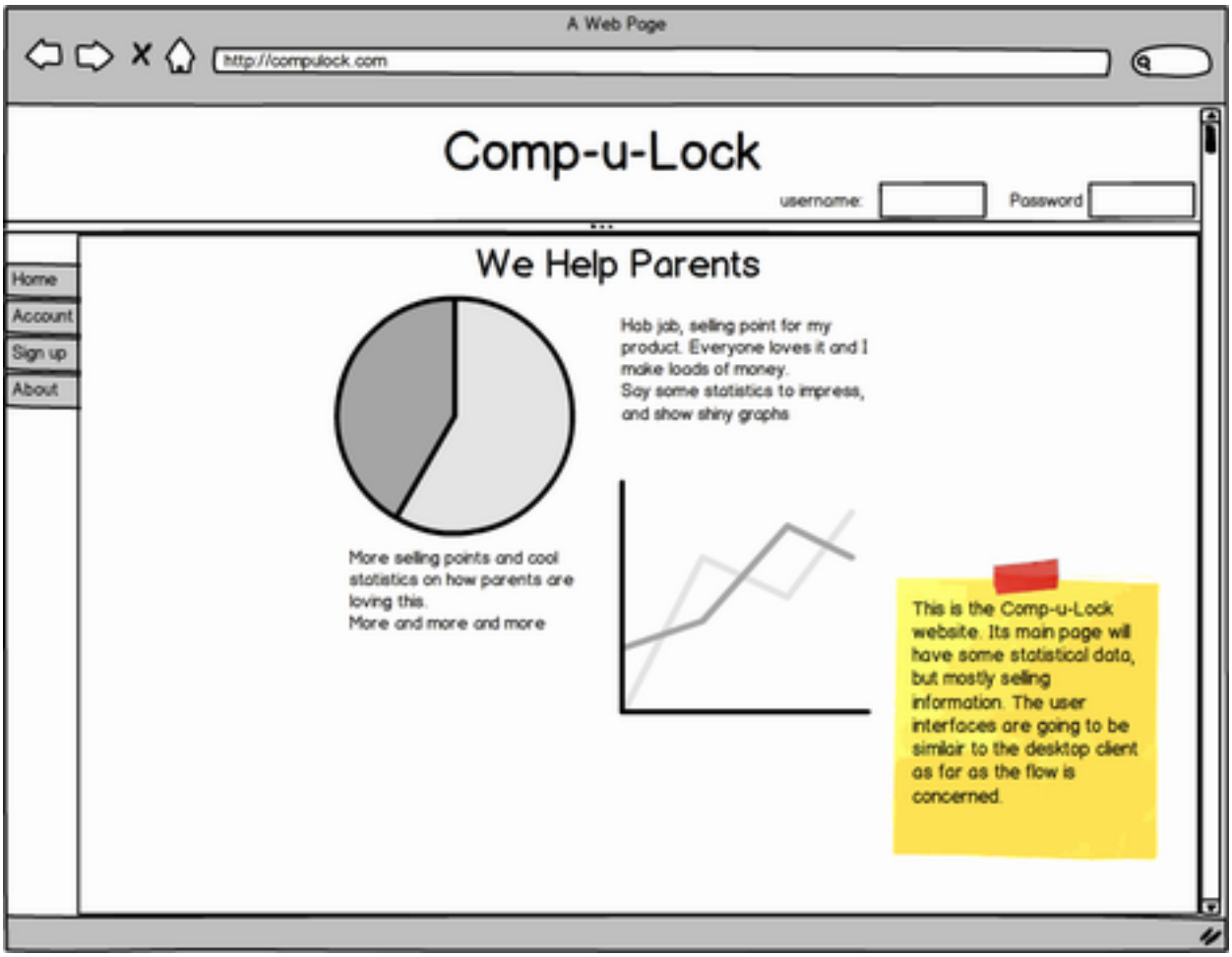
Settings Window

The settings window allows the user to set settings that are available to be modified for the application.



Webpage

The Comp-u-Lock website offers account control online. The user can use this website to control a computer from a distance, and view log information on the go.


A Web Page
http://compulock.com
Comp-u-Lock
username:
Password
Home
Account
Sign up
About
We Help Parents
Hab jab, selling point for my product. Everyone loves it and I make loads of money.
Say some statistics to impress, and show shiny graphs
More selling points and cool statistics on how parents are loving this.
More and more and more
This is the Comp-u-Lock website. Its main page will have some statistical data, but mostly selling information. The user interfaces are going to be similair to the desktop client as far as the flow is concerned.

Use Cases

Web Cases

WUC1 - Creating an account

Actor	User
Preconditions	The user has visited the main page

Basic Flow	1. The user selects the “Register Here” link
	2. The user inputs their username
	3. The user inputs their email address
	4. The user inputs their chosen password and confirms it
	  i. The service confirms the email is not already registered to an existing account.
	  ii. The system confirms the passwords are the same
	  iii. The service confirms the username is not already taken by a previously registered account
	5. The service sends a confirmation email to the user
	  i. The user clicks the link in the email to confirm their account

Basic Postconditions	The user has an account

Alternative Flows	5. The user provides an email address that is taken
	  i. The service informs the user the email address is taken
	  ii. The system asks if the user to recover their password
	  a. If the user confirms, the system sends a recovery email to the registered email account.
	5. The user provides a username that is taken
	  iii. The service informs the user that username is taken
	  iv. If the user does not change the username the dialog fails.
	5. The user doesn’t provide the same password and confirmation
	  v. The service informs the user that the password and confirmation were not the same
	  vi. The user must change the password to match
	  a. If the user fails to match the password the dialog fails
	6. The user does not click the confirmation email
	  vii. The users account is not activated until the verification link is clicked.

Alternative Postconditions	Success() - Failure() -
Notes	

WUC2 - Logging in

Actor	User
Preconditions	The user has visited the main page
Basic Flow	1. The user selects “Login” 2. The user enters their username and password 3. The service verifies the username/password combination 4. The service logs in the user
Basic Postconditions	The user is logged into the service
Alternative Flows	3. The username is not valid i. The service informs the user that the username or password is incorrect ii. The service asks if the user to register or recover their password i. If the user selects to register go to WUC1 ii. If the user selects to recover password go to WUC3
Alternative Postconditions	Success() - Failure() -
Notes	See: WUC1 - Create an account

WUC3 - Recover Account Password

Actor	User
Preconditions	The user has tried to login to the system
Basic Flow	1. The user clicks recover password 2. The user enters in the account email 3. The service verifies the email is valid 4. The service sends an email to the specified email i. The user clicks the reset password link ii. The user puts in a new password and confirms
Basic Postconditions	The user gets an email with a reset password link
Alternative Flows	3. The entered email is not associated with an account i. The service notifies the user that the email is not

	associated with an account. a. The service asks if they would like to register b. If the user clicks to register go to WUC1 4. The user fails to click the reset password link provided in the email i. The users password remains unchanged
Alternative Postconditions	Success(4i-ii) - The user has changed their password Failure(2,4) - The service remains unchanged
Notes	See: WUC1 - Account an account

WUC4 - Downloading the application

Actor	User
Preconditions	The user is logged in
Basic Flow	1) The user clicks the download link 2) The user clicks the confirm for downloading the application
Basic Postconditions	The user has downloaded the application
Alternative Flows	1. The download doesn't start i. The user clicks the start download link
Alternative Postconditions	Success(1) - The user downloads the application Failure(1 - 2) - The user doesn't download the application
Notes	See:

WUC5 - Changing account information

Actor	User
Preconditions	The user is logged in
Basic Flow	1) The user clicks on the "My account" link 2) The user clicks "Edit Info" link 3) The user inputs a new email i. The system confirms the email is not already in use 4) The user inputs a new password and confirmation ii. The system confirms the passwords match 5) The user clicks save
Basic Postconditions	The user changes their account information

Alternative Flows	3) The user inputs an invalid email i. The system tells the user the email is already taken
Alternative Postconditions	Success(5) - The user has changed account information Failure(2-5) - The service is unchanged
Notes	See:

WUC6 - Managing associated computers

Actor	User
Preconditions	The user is logged in
Basic Flow	1) The user clicks on the “Manage Account” link 2) The user clicks delete computer icon next to the associated computer 3) The user clicks edit computer icon next to the associated computer i. The user renames the computer ii. The user clicks save
Basic Postconditions	An associated computer is edited from the account
Alternative Flows	1. There are no computers to manage i. The dialog fails
Alternative Postconditions	Success() - Failure(2-3) - A dialog displays saying there are no computer to manage.
Notes	See:

WUC7 - Managing associated computer accounts

Actor	User
Preconditions	The user is logged in
Basic Flow	1) The user clicks on the “Manage Account” link 2) The user clicks on the desired computer that contains the account 3) The user clicks the edit account icon i. The user selects a new time range ii. The user clicks the delete account icon iii. The user clicks save
Basic Postconditions	The user has modified the associated account

Alternative Flows	3) The time range is out of bounds i. The dialog fails
Alternative Postconditions	Success() - Failure() -
Notes	See:

WUC8 - Viewing Logs

Actor	User
Preconditions	The user is logged in
Basic Flow	1) The user clicks on the “My Dashboard” link
Basic Postconditions	The logs are displayed to the user
Alternative Flows	1) There is no information to display i. A dialog is displayed informing the user
Alternative Postconditions	Success() - Failure() -
Notes	See:

Desktop Use Cases

DUC1 - Creating an account

Actor	User
Preconditions	The application is running
Basic Flow	1. The user selects the “Register Here” link 2. The user inputs their username 3. The user inputs their email address 4. The user inputs their chosen password and confirms it ii. The service confirms the email is not already registered to an existing account. iii. The system confirms the passwords are the same iv. The service confirms the username is not already taken by a previously registered account 5. The service sends a confirmation email to the user v. The user clicks the link in the email to confirm their account

Basic Postconditions	The user has created an account
Alternative Flows	5. The user provides an email address that is taken viii. The service informs the user the email address is taken ix. The system asks if the user to recover their password a. If the user confirms, the system sends a recovery email to the registered email account. 5. The user provides a username that is taken x. The service informs the user that username is taken xi. If the user does not change the username the dialog fails. 5. The user doesn't provide the same password and confirmation xii. The service informs the user that the password and confirmation were not the same xiii. The user must change the password to match a. If the user fails to match the password the dialog fails 6. The user does not click the confirmation email xiv. The users account is not activated until the verification link is clicked.
Alternative Postconditions	Success() - Failure() -
Notes	See:

DUC2 - Logging In

Actor	User
Preconditions	The user has an account
Basic Flow	1) The user opens the application 2) The user types in their username or email 3) The user types in their password 4) The user clicks login
Basic Postconditions	The user has logged into their account
Alternative Flows	2) The user has put in an invalid username or email i. The system replies with a response 3) The user has put in an invalid password i. The system replies with a response 4) The user doesn't have an account i. The user clicks the register link a. Go to DUC1

Alternative Postconditions	Success() - Failure() -
Notes	See: DUC1 - Create an account

DUC3 - Logs

Actor	User
Preconditions	The user has logged in
Basic Flow	1) The user clicks on the “Logs” tab 2) From the dropdown menu the user selects an account to view logs for
Basic Postconditions	The user views specific log information
Alternative Flows	
Alternative Postconditions	Success() - Failure() -
Notes	See:

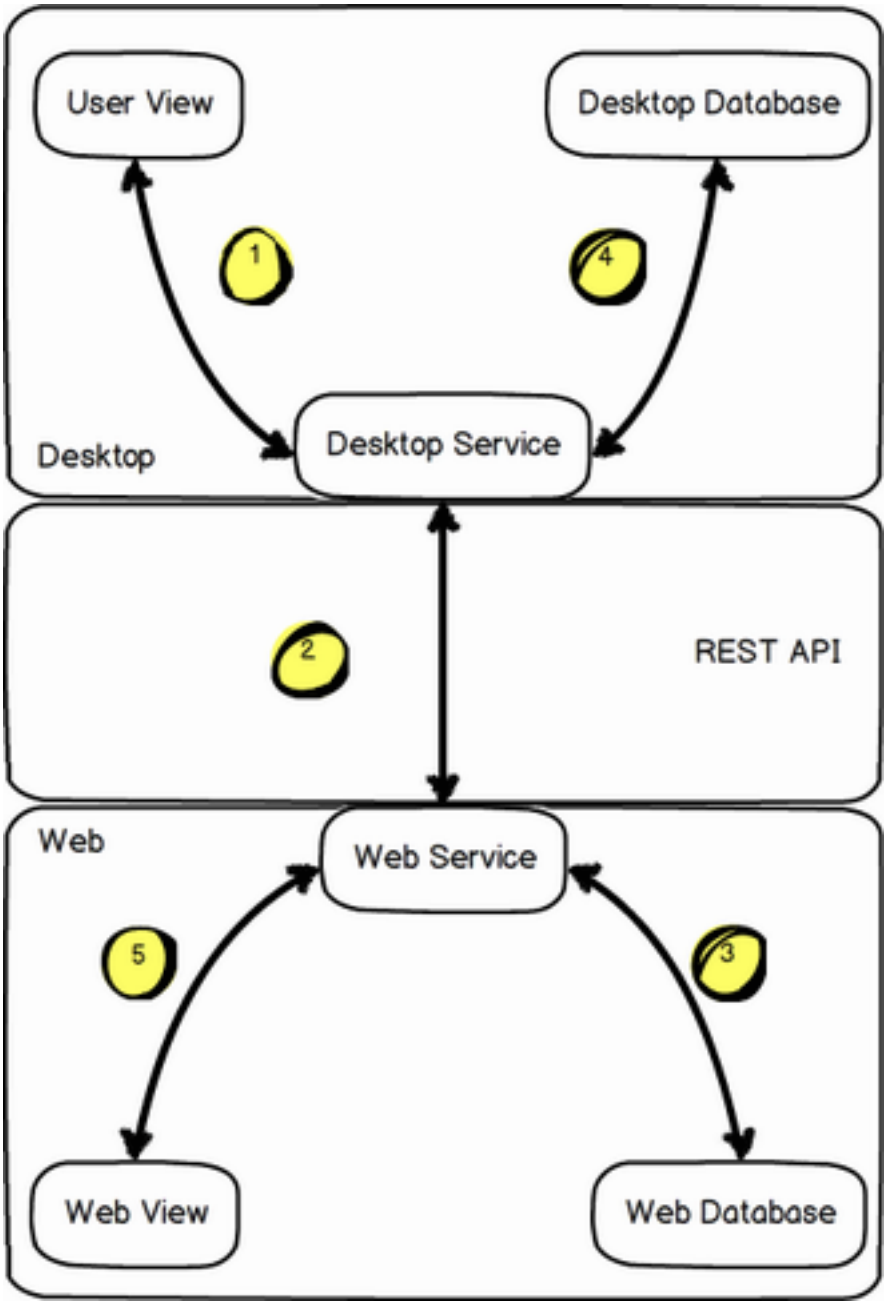
DUC4 - Hours

Actor	User
Preconditions	The user is logged in
Basic Flow	1) The user clicks the Hours tab 2) From the drop down the user selects an account 3) The user selects days that the account is able to login 4) The user specifies a time frame for each day 5) The user clicks save
Basic Postconditions	The user has modified the time frames
Alternative Flows	
Alternative Postconditions	Success() - Failure() -
Notes	See:

DUC5 - Settings

Actor	User
Preconditions	The user is logged in
Basic Flow	1) The user clicks the “Settings” tab 2) The user specified a new log location 3) The user clicks save
Basic Postconditions	The user has modified the settings
Alternative Flows	
Alternative Postconditions	Success() - Failure() -
Notes	See:

Project Architecture



Comp-u-Lock is designed in a service-oriented architecture. This allows the addition of other devices and clients to the structure with minimal effort. The desktop and web services are the “middlemen” between their databases and views.

In the diagram above, the desktop client creates a request that sends user’s information to the server (1). The web browser performs the same action but using the web view (5). The database retrieves or sets the information(5). The requested information is then sent back to the

view that originally sent the request.(2,5) The service processes the information, and displays it back to the user using the views provided.

From the desktop client, the desktop service pulls information from the database and sends this information to the web service using the REST API (4,2). The web service processes the information it receives. The web service stores the information into the database (3). The web service returns a response confirming the storage of the data or returns an error and what state it is at (2).

From the web client, settings are sent to the web service (5). The web service processes the information and stores the data (3). The configuration information is then packaged and sent to the desktop service (2). The desktop service processes the information and saves it to its database (4).

Desktop Database: Stores information about settings for that computer. It also stores log information that is recorded.
Desktop Service: Handles requests sent and received from the web service. It acts as one of the middlemen between the web and its own interface.
Desktop view: The application that provides an interactive interface for the user to use. Settings are pushed to the desktop service from this view.

Web Service: Handles requests received and sent from the desktop service. It acts as the other middleman between the desktop service and the web service.
Web Database: Stores information about settings for associated computers, associated computer accounts, login credentials, and log information sent from the desktop service.
Web View: The interface that provides an interactive interface for the user to use. Settings about accounts and computers can be modified using this view.

Configuration Settings: These settings are for the desktop client to use. These settings will be sent as a model, so the client can easily process the received information.
General information: This information is primarily user credentials and computer registration information.

Technological Description

The following set of tools will be used to complete Comp-u-Lock:

- WPF
    - Desktop user interface
- C#
    - Managing and creating users
    - Creating a windows service
    - Accessing rest APIs
- Visual Studio

    - .NET development
- Ruby on Rails
    - Web platform
    - REST service
- Rspec
    - Ruby unit testing
- MsUnit
    - C# unit testing
- HostGator.com
    - Hosting site for websites and databases
- Version control
    - Code versioning software
- CSS and JQuery
    - Personalization
- Postgres and/or SQLite
    - Database software

Novelty

Summary

Comp-u-Lock is important to me because I have a younger brother and sister who love to be on the computer. There have been several studies that have shown that prolonged use of the computer can actually have negative side effects on children (mainly related to social behavior). Within my family, the computer is a source of stress for my parents because they do not know how to use it as well as they would like. The younger generations are also becoming adept at using computers and therefore helpless parents do not know how to control their children's computer use. I am making this software for my parents, but the project also presents an opportunity to make money outside of school. Capstone is also an opportunity for me to be able to finally create one of my ideas and show it to the world. This project will also challenge my ability to learn because I will be delving into new topics that I have not yet had much time to explore. Because of these challenges and complexities, creating this program will allow me to demonstrate how much I have learned in my studies. This project will allow me to present an outstanding program that I can proudly claim as my own.

Learning Deliverables

- Display proper testing for rails
- Display good time management skills
- Display proper MVC practices
- Display successful/good practices for implementing REST and the data it sends.
- Show that I have learned beyond the scope of my .NET classes with the account control and application logging.

Development Process

For this project to be completed the following deliverables have been laid out.

- **Week 1** (Jan 7 - Jan 11)
    - Desktop Module
        - Record processes
        - Record programs
        - Record browser history
            - Internet explorer
            - Chrome
            - Firefox
        - Error handling for the above
        - Unit testing for above
- **Week 2** (Jan 14 - Jan 18)
    - Desktop Module
        - Change a selected user's password
        - Change a selected users password based on a time limit
        - Unlock an account with a master password
        - Error handling for the above
        - Unit testing for password changes and master password
- **Week 3** (Jan 21 - Jan 25)
    - Desktop Module
        - Save information from week 1 to the database
        - Save information from week 2 to the database
    - Web Module
        - Create the database
            - Login information
            - Web user computers
            - Web user computer accounts
            - Log information for computer accounts
        - Register and login authentication
            - Controller
                - Authentication
            - Views
                - Login
                - Register
        - Unit testing for authentication
        - Error handling for authentication
- **Week 4** (Jan 28 - Feb 1)
    - Web Module
        - Create controllers
            - Account

                - Computers
                - Computer accounts
            - Create views
                - Account
                    - Overview
                    - Logs
                    - Associated computers
                - Computers
                    - Add
                    - Remove
                    - Associated computer accounts
                - Computer accounts
                    - Add remove
                    - Time tokens
                    - Hours
- **Week 5** (Feb 4 - Feb 8)
    - Web Module
        - Develop and implement architecture for REST
            - Sending/receiving settings
            - Sending/receiving time constraints
        - Processing and storing the information
        - Unit testing for the above
        - Error handling for the above
- **Week 6** (Feb 11 - Feb 15)
    - Web Module
        - Develop and implement architecture for REST
            - Sending/receiving login/register information
    - Desktop Module
        - Develop and implement architecture for REST
            - Sending/receiving login/register information
            - Sending/receiving settings
            - Sending/receiving time constraints
        - Processing and storing the information
        - Error handling for the above
        - Unit testing for the above
- **Week 7** (Feb 18 - Feb 22)
    - Desktop Module
        - Create WPF application
            - Login
            - Register
            - Overview
            - Logs
            - Hours
            - Settings

        - Unit testing for GUI
        - Any error handling needed
- **Week 8** (Feb 25 - Mar 1)
    - Web Module
        - Apply cosmetics to the site
            - Layout base
            - Css mask
            - Javascript/Coffeescript toning
        - Any unit testing needed
        - Error handling via javascript
- **Week 9** (Mar 4 - Mar 8)
    - Demo Module
        - Create test accounts
        - Pre populate the web and desktop database
        - Create any recordings of processes for presentations
        - Create powerpoint presentation
        - Finish any polish
- **Week 10** (Mar 11 - Mar 15)
    - Finish Demo Module
    - Present and receive grade

Resources

Comp-u-Lock uses the following resources:

- .NET
    - C# - Language of choice
    - MsUnit - Unit testing
    - WPF - Desktop application interface
    - Visual Studio - .NET development software
    - MSDN - Reference
- Ruby
    - Rails - Web framework of choice
    - Rspec - Unit testing
    - Rails for Zombies - Reference
    - RailsCasts - Reference
    - Sublime - Ruby development software
- References
    - Jamison Greeley - Best practices for Ruby on Rails
    - StackOverflow - Reference
- Hosting
    - Hostgator.com - Web hosting
    - Postgres - Database engine
    - SQLite - Desktop database engine

- Version control
    - Git
    - Bitbucket - Code source hosting site
- Beautify
    - CSS - Layout and design
    - Jquery/Coffeescript - UI integration
    - Bootstrap - adaptive layout/interface