DisciPlan

Adam Brostowicz, Jeff Garnier, James Ross, Danny Thomson

Description of Project:

Our project will be a browser plugin that we intend to make for both Firefox and Google Chrome. The goal of the plugin is to collect information about the user's internet usage, use charts and graphs to display that information to the user, and allow him or her to create custom plans to change their internet usage however they see fit. The plugin's main job will be to collect internet usage statistics from its users, mainly what websites they visit and how much time they spend on each. It will also categorize the websites that the user is visiting (social media, entertainment, academic, etc). If we cannot determine the category of any website that the user frequents, the user will be able to "tag" a website as a certain category. The plugin will also have an interface that displays graphical information about the user's internet usage. The graphs will split the information based on website categories so that the user can see exactly how much time he or she is spending on both specific websites and in each category. The graphs will also display category trends over time, so that the user can see any progress he might make.

In addition to data collection and graphical displays, the plugin will also allow category based goal setting for the user. For example: if a user wants to spend only 1 hour on social media per day, he or she would specify that within the plugin's goals page and the plugin would take certain actions after he or she reaches the limit. The plugin will have a gradient of strictness for the actions it takes, ranging from blocking popular social media sites like Facebook or Twitter after one hour has been spent with a social media site on an active tab, or simply sending a notification when the time is up. Within the goals page, the user will also be able to add or remove specific websites from the block list, or change the categories as they apply for the user.

Each user will have an account with the plugin. When a user logs into the app from a different device or browser, all of the user's settings and information will be preserved. This will ideally allow for a seamless experience, and won't let the user cheat the

system by switching to another browser, as long as the plugin is installed on both of them.

Need For the Product:

Our product is needed because of the immense number of distractions available on the internet. It's really easy to get distracted, lose track of time, and end up spending time where one should have been very productive getting absolutely nothing done. Oftentimes people only need a small reminder to keep them on task, and any sort of barrier introduced between the user and a website like Facebook would probably be enough to stop them from wasting hours reading through all of their friends' posts. In addition, it's one thing to know that you have wasted some time on social media sites, but it's entirely another to have an app show you exactly how much time was spent, when it was spent, and graph it next to the "productive" time. Being shown not only how much time a user has spent unproductively, but also improvements will help keep people motivated and eventually learn to spend their time on the internet more productively.

Potential Audience:

Our audience will likely be high school and college students who want to improve their productivity. There is a fairly large audience who recognizes that they spend too much time being unproductive online, and our tool would help alleviate that problem for them. The level of technical sophistication required is quite minimal. All the user needs to know is how to search for plugins on Firefox, or chrome extensions on Chrome. From there, it does not require a high level of sophistication to find and change settings within that plugin. Furthermore, the graphs should be visually pleasing and easy to understand for someone with any level of technical sophistication. While our plugin caters to the young more than the old, that is likely the same population that spends a lot of time browsing the internet.

Discussion of Competing Products:

DisciPlan is a combination of two different Internet browser extensions. In Google Chrome, these extensions are called StayFocusd and Time Tracker. StayFocusd is an extension that allows users to visit unproductive websites for a set amount of time everyday. These unproductive websites are defined by the user. After exhausting the allowed time for that day, the extension will prevent the user from visiting these websites. By thwarting quick access to these websites, the extension stops users from easily getting off track from their work.

DisciPlan will take many of the features of StayFocusd and build upon them. StayFocusd does not provide many controls to adjust the behavior of the extension. A user can only define what websites should be restricted and how much time allowed visiting these websites in a day. To improve the extension, DisciPlan will allow different amounts of time for different websites or categories, provide different punishments for visiting a unproductive website after exhausting the user's allowed time, and offer a more clean interface to interact with these settings.

Since DisciPlan will be tracking the user's web usage, DisciPlan will be able to use this data to make recommendations about the user's web usage. Our tracking features will provide more in-depth information about the user's usage than the other web usage statistics extensions. Unlike Time Tracker and the other various web usage statistics extensions, we will integrate this information into the new tab page of the browser to ensure that this information will seen and used by the user.

Major Technologies Used:

We're looking to make our plugin first for Chrome, and then for Firefox. This will require to know the differences between extensions development for these browsers; Chrome, for example, has easier developer environment setup and debugging thanks to its developer dashboard. As a browser plugin, our project will be largely coded in JavaScript. The first set of JavaScript code we'll be writing will deal with monitoring

browser usage. Since we'll be storing statistics on internet usage so that the user can later display them, these numbers will be sent back to a server which we will choose to write in either Java (in which case, we'd probably use Apache Tomcat) or Ruby on Rails. This server will be responsible for writing stats to (and later reading stats from) a SQL database (if we end up using AWS, this will probably be a PostgreSQL or MySQL DB). When users want to view their usage stats, we'll produce pages using HTML and CSS in tandem with JavaScript. Since these pages will include graphics like charts comparing time spent on different categories of websites, we'll use an open source JavaScript charts and graphs library (this may be D3.js, Google Charts, or another charting library).

Resource Requirements:

For hosting our server while building and testing, we'll either use the Corn machines or Amazon Web Services (AWS), in which case we'll need some AWS credits.

Potential Approaches:

DisciPlan could be released as an independent desktop application or directly integrated into the browser's source code. While both of these options provide more control, these options will be difficult to develop and may raise security issues. Separate desktop applications provide control over all the applications running on the user's computer, but DisciPlan focus is on the browser. A desktop application created to micromanage a browser will be significantly more difficult to develop over a browser extension. Unlike a desktop application, a browser extension will have easy access to the browser and compatible between different operating systems making development significantly easier.

Meanwhile, modifying the browser's source code will provide much more control over the browser. However, we could introduce security issues by changing the source code of the actual browser. Web browsers sandbox extensions and plugins giving them

only authorization to the resources they need. Thus, if an extension or plugin were hacked, the hacker would not have access to all of the browser's resources. However, if the code were directly integrated into the browser, a hacker would obtain those resources. With this access, a hacker could potentially learn a user's personal information. Aside from the security concerns, users are more likely to download an extension/plugin to their browser over a separate browser. A separate browser is not likely to be compatible with other browser extension/plugins and will be more difficult to update. As a result, a web extension is the most efficient approach.

Assessment of Risks:

One potential risk comes from the fact that none of us have ever built a browser extension. There seems to be good documentation on the Internet about how to build extensions for both Chrome and Firefox, but learning how to do this may take longer than we expect, or we might not be able to get it working at all. If we have trouble setting up the extensions, then we are losing valuable time that could be spent building a better extension with more complete and aesthetic features. Part of this risk also comes from that we are trying to develop an extension for two different web browsers. We might have no trouble getting the extension set up on Chrome, but if we cannot get it set up on Firefox, then we lose a lot of the core functionality that we were planning, because we would not need the server to have the different browsers share the user's data. In order to reduce this risk, we need to be diligent in quickly learning how to set up an extension for both Firefox and Chrome so that we can actually start building our extensions as early as possible.

Another possible risk is that we will not be able to get the server talking to the browsers. We have little background setting up a real server outside of class assignments. If we cannot get the server working properly we will not be able to have the different browsers share their information about the user's browsing usage. One of

the key features of DisciPlan is that the user can get the extension for Chrome and Firefox and it will keep track of all of their Internet usage statistics so that they will be aware of their statistics even if they switch browsers/computers, and without the server this will not be possible. In order to reduce this risk, we need to have a clear plan for developing the server, and have milestones throughout the quarter to make sure it is getting set up properly. If we have trouble setting up the server we might need to get guidance from someone who has more experience than us. The major risks in this project are learning how to set up extensions in Chrome and Firefox and setting up a server that we can access from these browsers. We need to start these tasks early so that if we run into any problems they can be handled, so that we have more time to actually work the extensions themselves.

Next Steps:

First, we need to research how to build extensions in Chrome and Firefox. Once we figure out how to get started building our extensions we can start figuring out how to collect user data and how we want to handle it. Along with building the extensions, we also need to get our server set up. We need to figure out if we are going to test it on the Corn machines, or figure out how to get the server set up on AWS. Once we get the extensions and the server set up, we need to get the extensions to query/store data in the servers. After we have all of this working, we need to design the UI for the extensions.