ScottySurf TP3 Deliverable

Project Description: The name of the term project and a short description of what it will be.

Welcome to ScottySurf. ScottySurfer is a Web Browser that aims to tackle modern browsing's critical failure: TABS! Modern computers are usually more wide than they are tall, so why do we put our tabs on the top of the screen as opposed to the side? According to revolutionary research (3 people were asked), horizontal screen real-estate is much more valuable than vertical screen real estate, and it's time that our browsers reflected that.

Competitive Analysis: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.

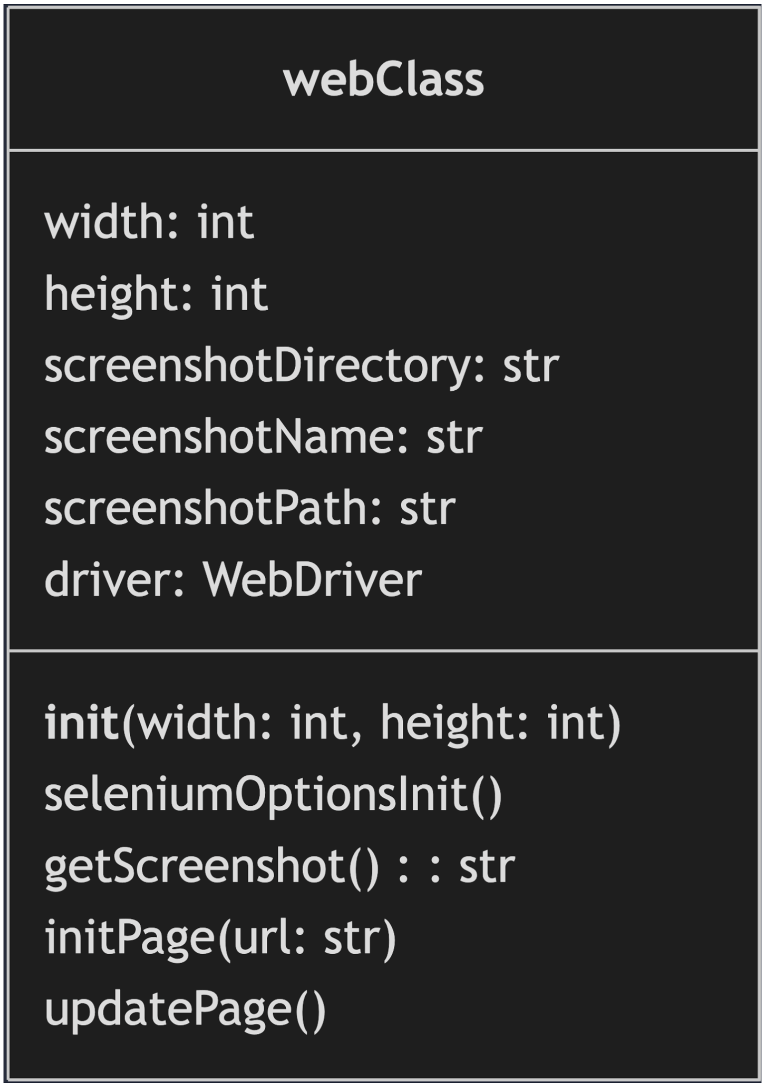
Arc Browser by the BCNY (Browser Company of New York) is similar to ScottySurf, but they no longer support it, and also have a whole stack of useless 'AI' features that actually harm the user experience. The Arc "MAX" features introduce friction inside of the UX and often harm UX. These features will be banned in ScottySurf. Additionally, past MVP, ScottySurf will include much more unique features tailored for students at CMU.

Structural Plan: A structural plan for how the finalized project will be organized in different functions, files and/or objects.

For MVP, I will have two main directories and a few subdirectories

```
Unset
      /draw
      ----/assets
      ----/code
      /backend
      ----/web-driver
      ----/rendering
      ----/jsons
```

This project will be designed using an Object Oriented framework, and will be designed to be modular and configurable. Here is the UML code for the most important class, the driver:



webClass

width: int
height: int
screenshotDirectory: str
screenshotName: str
screenshotPath: str
driver: WebDriver

init(width: int, height: int)
seleniumOptionsInit()
getScreenshot() : : str
initPage(url: str)
updatePage()

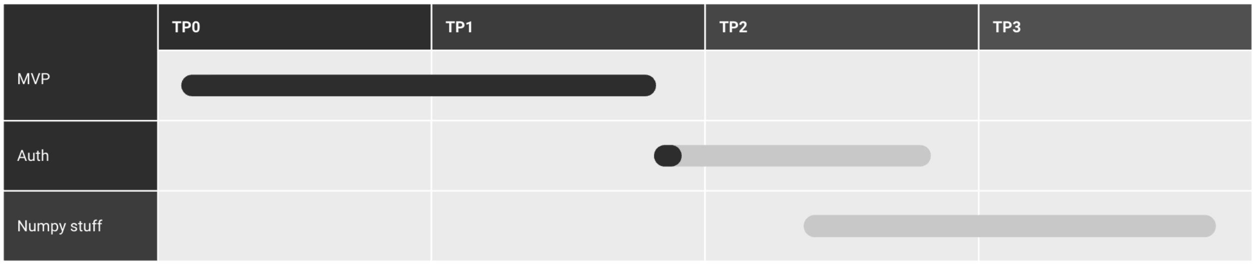
Algorithmic Plan:

Undoubtedly the trickiest part will be the rendering, which I will handle in a few clever ways. The actual site will be rendered in the backend using a fork of Selenium’s chrome driver, and will be updated on a frame-by-frame basis to reflect animations inside of the website. In terms of interaction, I will be using chromium’s headless mode through selenium, which allows me to simulate clicks and hovers using CMU_Graphic’s callback functions.

Other than that, I will be storing the tab data inside of a JSON which consists of three elements per tab, the name of the tab, the link, and the favicon to store the image of the website. The image will be stored as a CMU-Image for efficiency.

Timeline Plan:

By MVP deadline, I will be finishing the basic functionality of the website. By now (19 nov), I have already completed the rendering of the web-pages, and all that’s left is interactions for MVP.



Version Control Plan:

Github!

Since I am still a student of 112, I cannot make the repo public so email me to be added to the repo.

Module List:

For the MVP:

- BS4 (html parsing)
- Selenium (rendering)

Updated design plan (as of TP3):
No significant changes made ! 🙂

- I dialed in some Optimization fixes and worked on a few Post MVP super cool features