

## Criterion B: Design

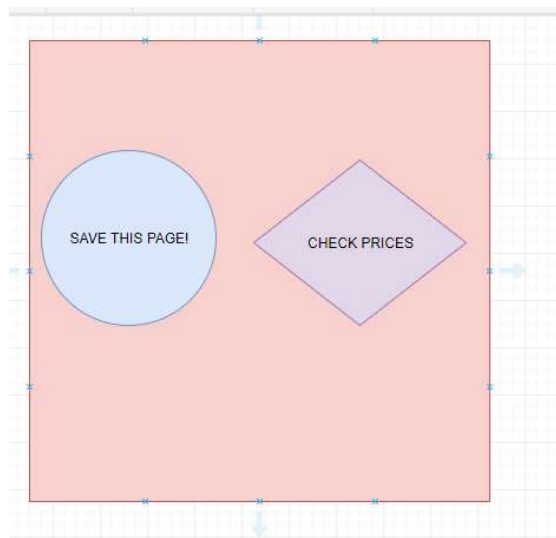
### The Chrome Extension

Since the project is a chrome extension it should be explained how chrome extension function in relation to my project. Chrome extensions the basic anatomy of the following:

- A "manifest.json" file which specifies which provides the chrome browser with the following information:
  - Which other files are being used in the chrome extension.
  - Which images to use for icons.
  - The version number of the extension.
  - The name of the extension.
  - The permissions the extension will require.
- A background JavaScript file ("background.js") which runs javascript code in the background of the browser.
- A popup file (popup.html) which serves as the user interface for the chrome extension. When the user clicks on the icon of the extension the popup.html file is provided.
- A style sheet (style.css) file for the popup.html for changing the user interface.
- An external JavaScript file for the popup file ("popup.js") this provides interactive programming elements which can communicate with the background.js file.

### The User Interface:

In order to meet the first success criteria outlined in section A I will need a clearly designed application. This will be done by keeping the design as simple as possible in order to make sure the client can operate the extension.



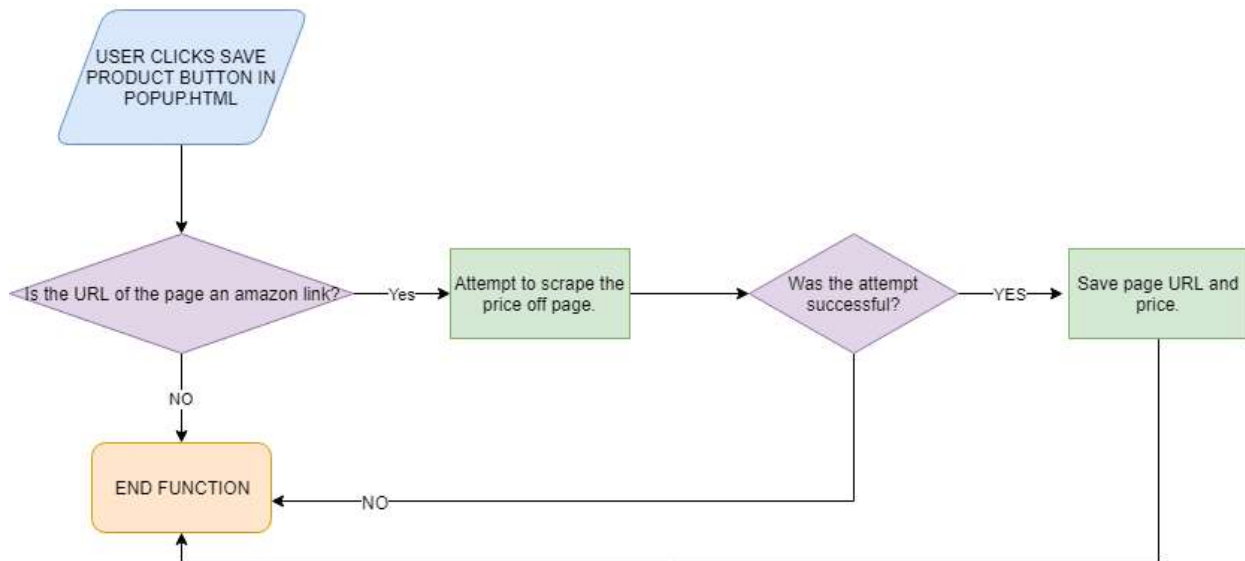
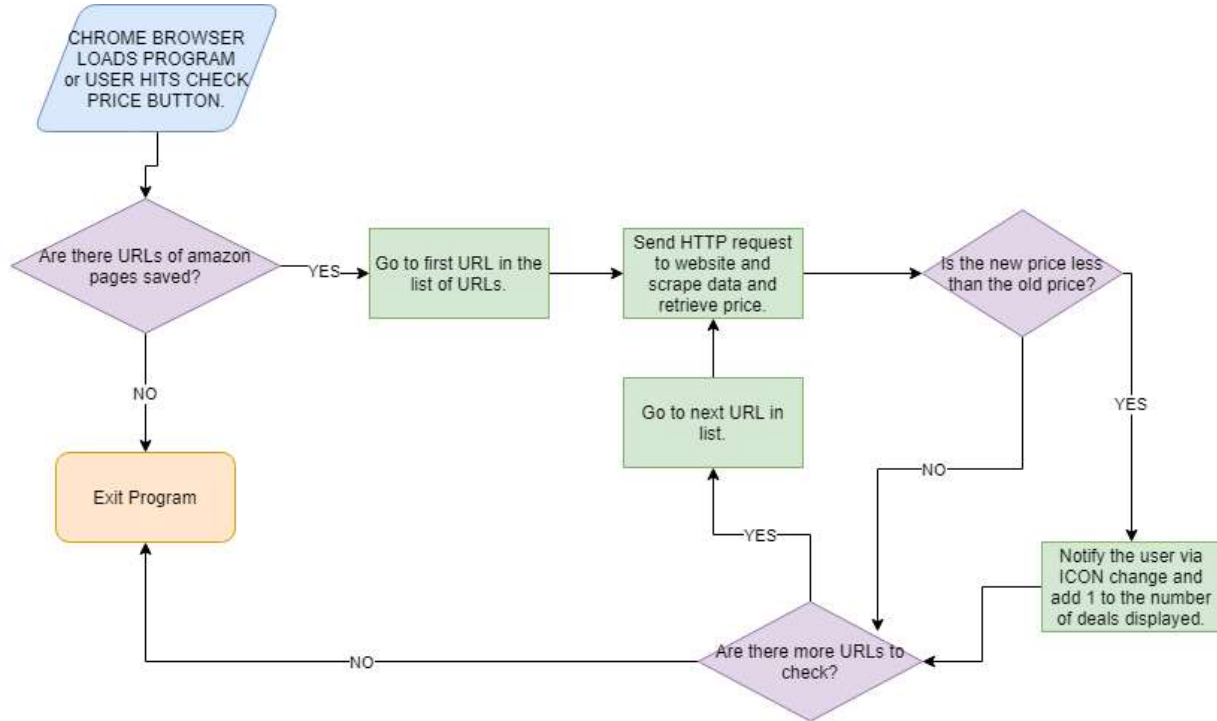
### The Algorithms

There are two main logical portions of the chrome extension which I will be building in JavaScript. The user interface will be done in HTML/CSS which doesn't use an algorithm to display information. As such the only main application of a flowchart would be the JavaScript portion of the extension.

The extension has two main functions:

- Saving Amazon product pages with the price.
- Check Amazon product pages for a new price and comparing it with the old price.

The flowcharts below showcase the algorithms that will be used in those two functions.



### Testing Success Criteria From Criteria A

Success Criteria	Test
The client can save products from viewing an amazon page.	The client can easily find the save button on the chrome extension. Tested in Evaluation.
The application save save the product page URLs across browsers as well as previously found price.	Tested by loading extension on the same account across multiple computers using the same browser.
The product can scrape the price of the product off of an Amazon product page.	Price scrapped is printed in console then compared with the price on the actual web page.
The application can notify the client of price changes in desired products.	The icon of the chrome extension can change.
The application can check product price on chrome browser start up.	The price of the product is printed in the console on extension loading as to insure that the price is actually checked.
The application can compare the new price with the old price.	The outcome is printed from the test. Either, "too expensive" or "too cheap".