

Shopaholic

Final Year Project Report

DT228

BSc in Computer Science

**Thuy Nguyen**

**Dr. Michael Collins**

School of Computing

Dublin Institute of Technology

**12/04/2019**



Abstract

The aim of this paper and application is to determine whether it was possible to build an intelligent system that allow users to get an accurate recommendation for a piece of clothing using different features and factors of an image.

The project consists of two parts. The first section is a React Native web application. This interface which allow users to keep track of the item. A user can register and login to view their dashboard. Each item in the dashboard is pull from the MongoDB Database. The application is ran on a server allowing the system to scrape the data hourly from the website in order to inspect its availability. Upon availability, the system notifies the user by sending an email to the provided information.

Second is a Python backend web service, hosted on Heroku. Using the VGG19 model the web service reuses the trained weights to build a new model. Furthermore the web service uses the K-NN classifier to plot the most similar item for the user.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Thuy Nguyen

12/04/2019

Acknowledgement

I would like to thank my supervisor Dr. Michael Collins for guiding me throughout my project. I’d also like to thank my mother and father for cooking food for me every day. Last but not least, I’d like to thank Zizo for encouraging me to do the best I can

Table of Contents

*(These are suggested contents only. You can add or modify if required)*

1. Introduction
   1. Overview of the project and the background behind it.
   2. Project Objectives
   3. Project Challenges
   4. Structure of the document
2. Research
   1. Eg. Research related to identifying the problem that this project solves, research into solution definition
   2. An overview of the technologies evaluated and selected or rejected and the rationale behind the key decisions.
3. Design
   1. Identification of a design methodology including why it was chosen
   2. Design of each of the project components eg: the UI, Network, Project Demonstration, source code layout
   3. Clearly identifying the list of features and use cases supported within the project.
4. Architecture & Development
   1. Overview of the system architecture and a diagram to represent all of the key elements within the architecture.
   2. Details of each component within the project, problems encountered and resolved, challenges overcome or worked around.
   3. Identify key development components;
   4. Identification/explanation of external APIs used versus own code ; List of classes of your code etc .
5. System Validation
   1. Testing
      1. What testing was performed, why it was selected and what are the key use cases within the project.
   2. Demonstration
      1. Identify what features can be demonstrated and show screen shots or reference a video online to show the project demonstration (for audience not at demo)
6. Project Plan
   1. Project Plan analysis and review of how it changed from the initial proposal including explanation of what changed and why, and suggestions on how to address this if the project was repeated.
7. Conclusion
   1. Analysis of the projects key elements identify the key learning obtained from the project and recommendations and suggestions for how the work can be improved on continued into the future.
8. Bibliography
9. Appendix
10. Introduction
11. Overview and Background of Project

There is no doubt that shopping play a big role in our lives. Whether you are grocery shopping or just buying a new outfit for an event. Most people these days find it easier to shop online rather than going to the store. Not only is it easier to shop onlinebut the process is much faster. Nowadays most people go online daily, 89% of the Irish population have access to the internet and an estimated 1.66 billion people worldwide purchased goods online in 2017. It is clear that the figures are only going to continue to increase, so why are we not making it more convenient for consumers to shop online?

Although online retail stores allow a consumer to select items and purchase it, they don’t allow the option to be notified when an item is back in stock of it is currently out of stock. Not only that but the consumer is not given the option to view the price history, so how are consumers able to choose the best time to purchase an item? All these functionalities will be beneficial for the consumer when online shopping as they are saving a lot of their time by not having to check if an item is back in stock or even just looking for something similar to the item they were looking at.

Despite the growth of e- commerce, none of the applications listed below cover the main functionalities such as image-base recommender, user notifications and price history viewing. Not only will this application potentially save the consumers time, but it allows consumers to enjoy and shop more efficiently.

A screenshot of a cell phone

Description automatically generated

<https://www.eshopworld.com/blog/ireland-ecommerce-insights-2018/>

<https://www.cso.ie/en/releasesandpublications/er/isshh/informationsocietystatistics-households2017/>

1. Project Objectives

The aim of this project was to accomplish the following objectives:

* Develop a React Native application that was fully functional. The application will be hosted on Heroku and accessible to everyone. The sign up and login functionality working for users to be able to register.
* The application will automatically scrape data daily to update the price history and also hourly to check the item’s availability. The price history will be displayed on a graph alongside the item details.
* The Python backend web service will be deployed into the cloud through Heroku. It can be served to the React Native application via HTTP requests.

Using the scraped images, the recommender system is able to generate relevant recommendations. These recommendations will be stored in the MongoDB database which can be accessed using Flask whenever it is needed. The recommendation will be displayed to the user when necessary.

1. Project Challenges

There is no doubt that issues and challenges will arise when developing a fully functional application. For this project, a few risks and challenges include:

The dataset used to train the machine learning model is very small. When training a model to classify images, a lot of dataset is needed to be accurate. Because there are no image dataset for clothes, the images for each class had to be scraped off the internet. This method is inefficient as images may duplicate. This is a common issue, it is sometimes known as a ‘Cold Start’ problem. This is where the circumstances are not yet optimal for the system to make an accurate recommendation. But as the project progress, more images can be retrieved and saved to the dataset to make the model more accurate and efficient.

Another issue includes the model being trained very slow. Some models take from a few days to weeks to finish the training process. This is inefficient as some models need to be retrained when parameters are tweaked. As this is a college project, the equipment may not be up to standards for a model to train quickly. This issue increases the time constraint of the project. A solution to this problem is to run the script on a cloud where the machine is more advanced and faster.

For the price history, a lot of data have to be stored in the database. For MongoDB, the maximum BSON document size is 16 megabytes. This is an issue as this is not enough to store all the data that is required for this project. To solve this issue, instead of storing every single item detail, the system will only store data for men and women’s shoes. The system is coded in such a way that it is possible to added more items to the database when needed.

Due to the application having to scrape data from the ASOS website every day, the website blocked **… etc \*\*\*\* (Finish this) \*\*\*\*\***

Lastly, because this is a college project, it is unlikely to have a large user base. This can cause testing risks. The application is unable to test on all types of users. It will most likely be a few dummy users, therefore the application may not be as optimised as it can be. It is also impossible to test how the system will scale in a production environment with hundreds or thousands of users.

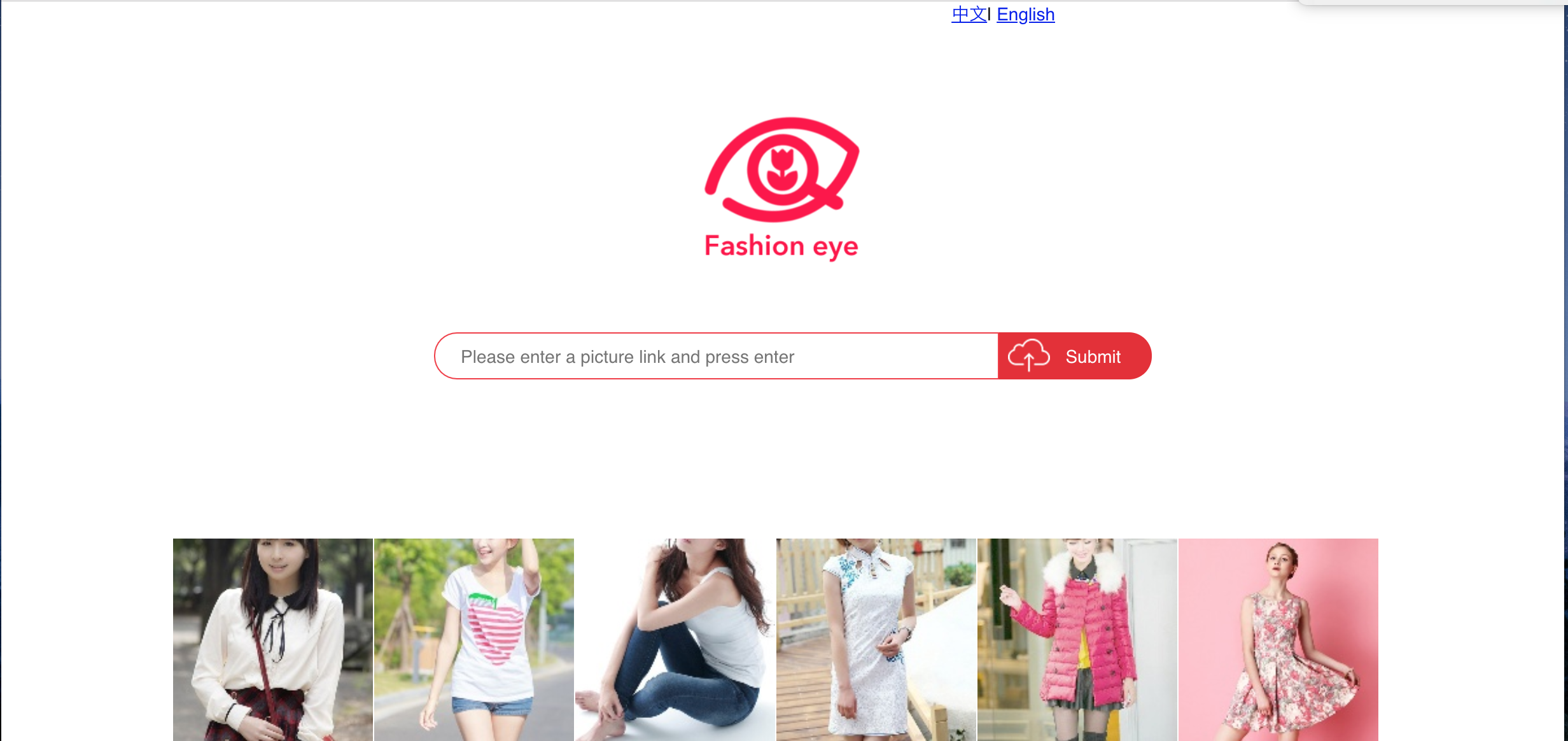
1. **\*\*\*\* Structure of the document ?????? \*\*\*\***
2. Research
3. Background Research

The background research consisted of finding a way to classify the input images into its own category as well as training the model so that it is able to find similarities between multiple images. Also considering alternative existing solutions to the project was part of the background research. Furthermore, different technologies were investigated in order to choose the best and most efficient one to use for the project.

1. Alternative Existing Solution to Your Problem

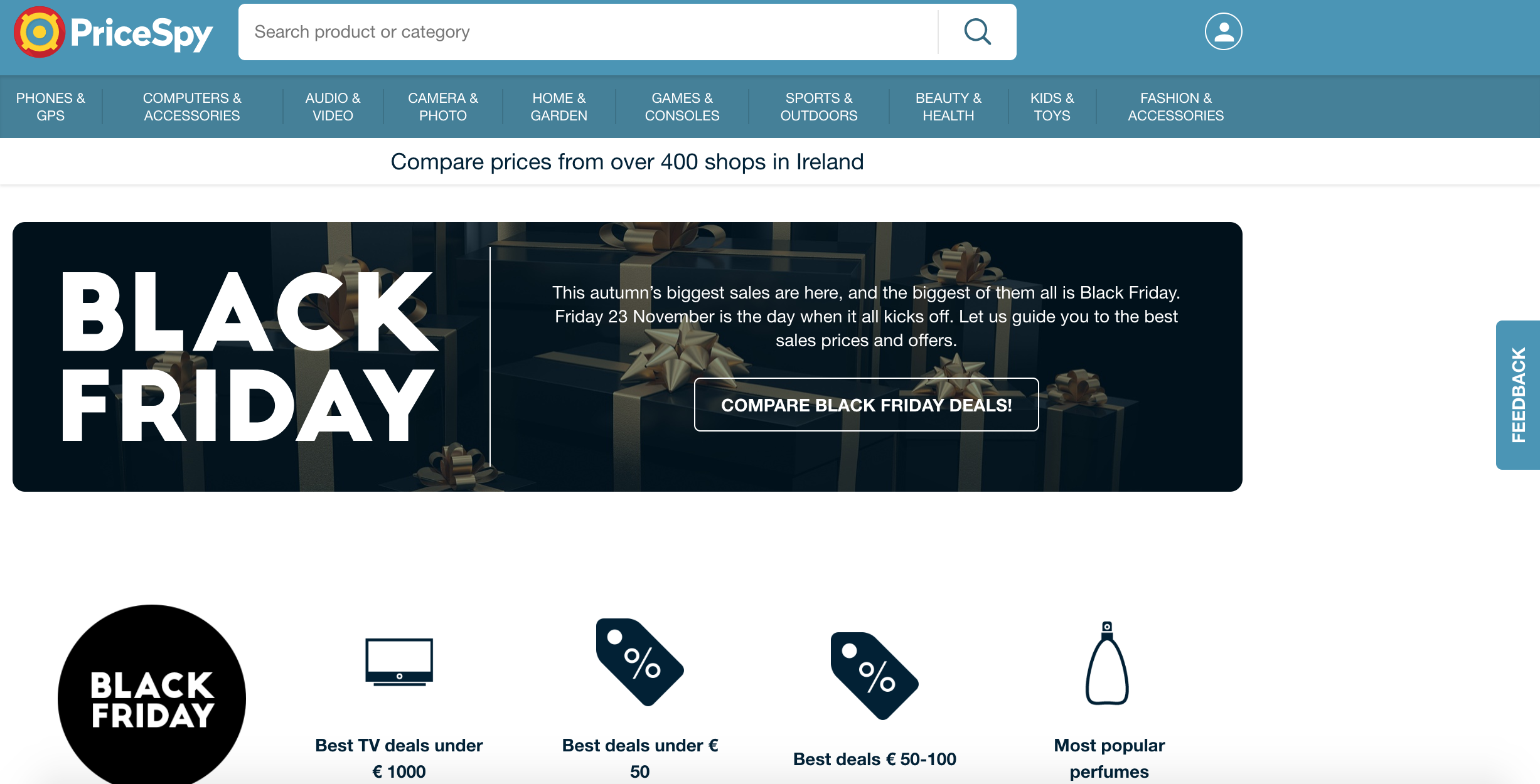
**1. Fashion Eye - http://fashion.sensetime.com/**

Fashion Eye is a website, implemented by college students. The website allows the user to upload and image and the website will recommend a location of where to buy this similar item. Although the system was implemented well, from time to time the system didn’t work. The website keeps on loading and no recommendations are made. This potentially can make the user very frustrated from waiting and disappointed from the lack of recommendation. Not only that but the website is marked as Not Secure, which is a big turn off when using the website as users might be paranoid over the security.

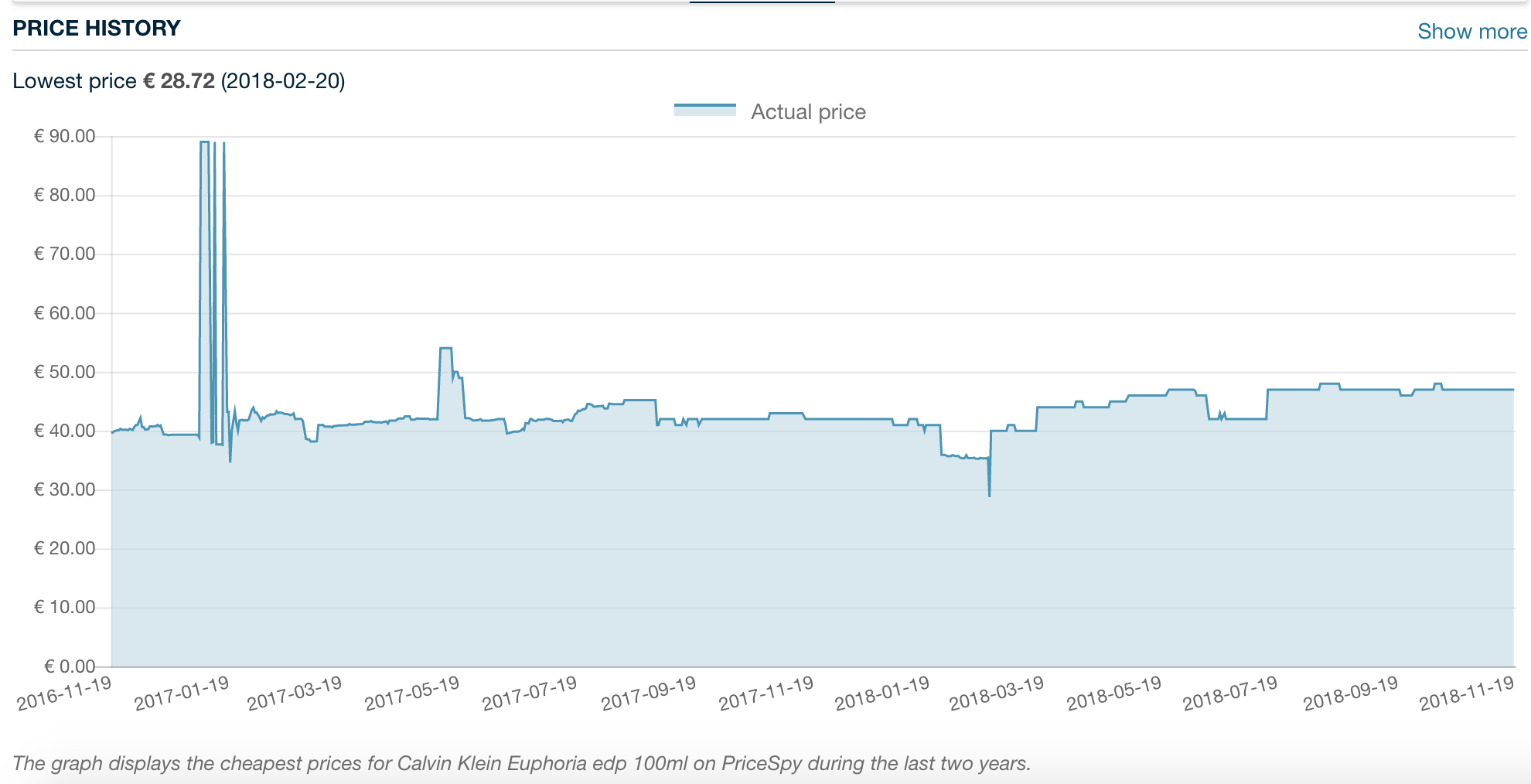


**2. PriceSpy - https://pricespy.ie/**

Price Spy is an application that allows users to compare prices. A user can search for an item and a list of locations and their prices are displayed allowing users to choose the best place to purchase the product.



Price Spy is an application that allows users to compare prices. A user can search for an item and a list of locations and their prices are displayed allowing users to choose the best place to purchase the product.



Another functionality that this application has is displaying price history. The prices are displayed on a graph with the y-axis being the prices and the x-axis displaying the dates.

This application also displays the price of the item with shipping and without shipping and the shop ratings. It is clear that this application was very well developed with many useful functionalities. The only disadvantage to this application is that it doesn’t cover some of the more popular online sites that users tend to shop in. This means that the user will have to quit this application to go compare it to the stores they normally shop in which defeats the whole purpose of the system.

**3. Back in Stock**

*Figure SEQ Figure \\* ARABIC 2.4 – Back in Stock Application*

*Figure SEQ Figure \\* ARABIC 2.5 – Back in Stock PricesFigure SEQ Figure \\* ARABIC 2.4 – Back in Stock Application*

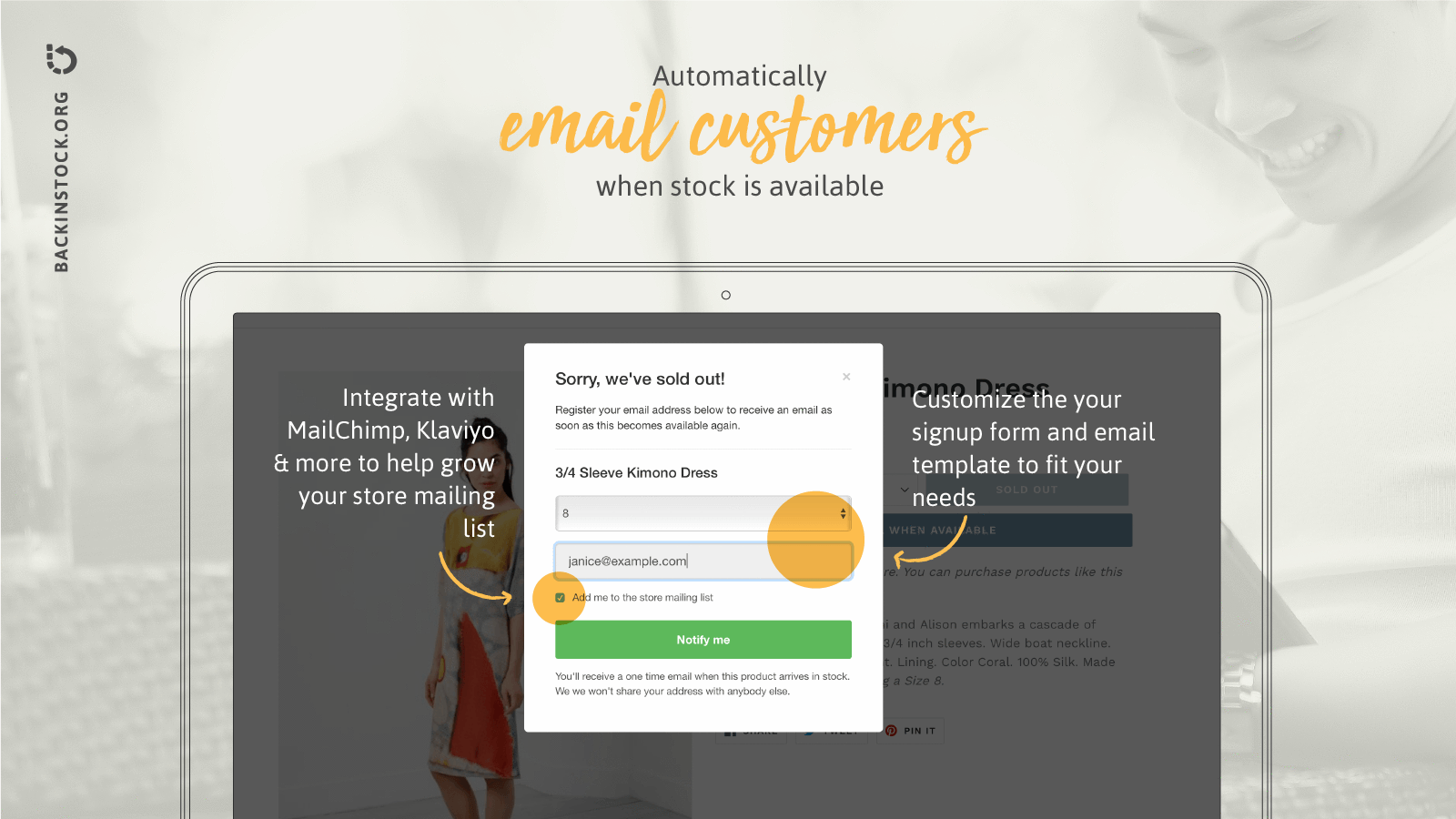
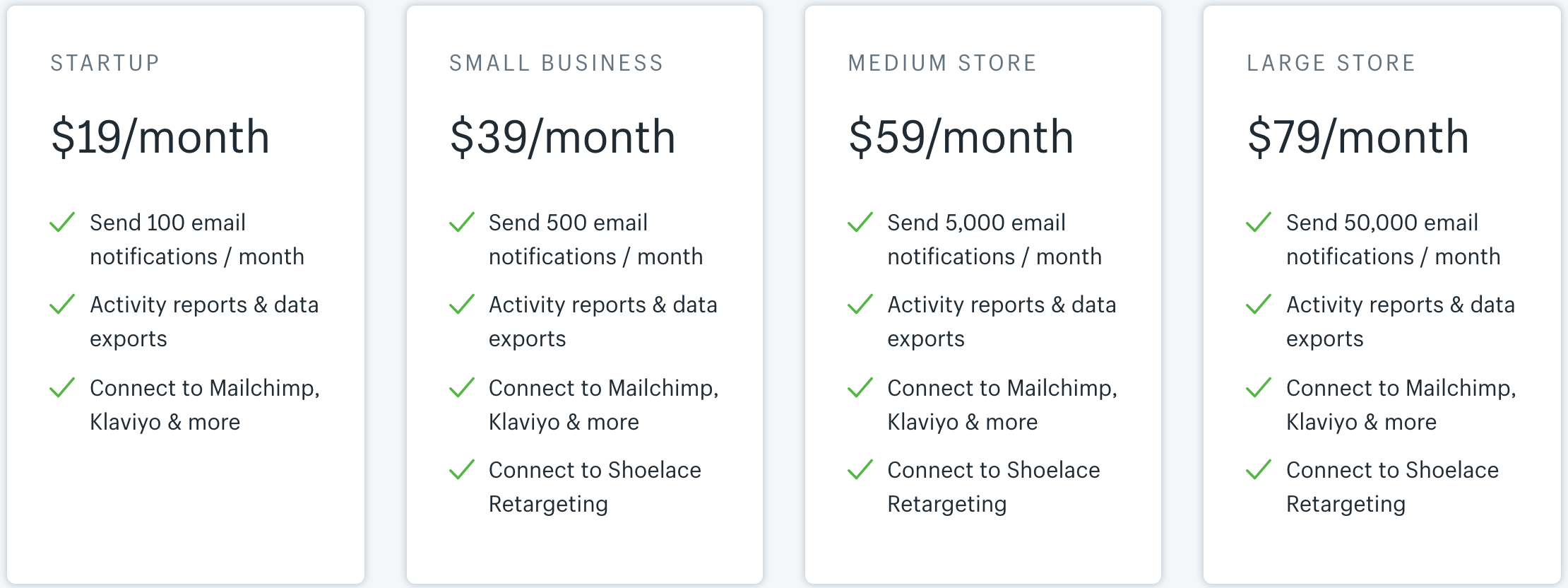
Back in Stock notifies a user when an item is back in stock. It is a simple application with no login or sign up required. A user has to install the application to the laptop/computer then when something the customer likes is out of stock, they enter their size and email address and the application will send out an email to them when the item is back in stock.

*Figure SEQ Figure \\* ARABIC 2.5 – Back in Stock Prices*

Although this application is useful, some drawbacks include a monthly fee of $19 for a simple 100 emails sent per month. The price increases if the customer chooses bigger plans.

*Figure SEQ Figure \\* ARABIC 2.2 - Price Spy Website*

*Figure SEQ Figure \\* ARABIC 2.3 – Price Spy Price HistoryFigure SEQ Figure \\* ARABIC 2.2 - Price Spy Website*



1. Technology Researched

**Programming Languages:**

**1. NodeJs**

NodeJs is a server-side platform built on Google Chrome’s JavaScript runtime. Everything that one may need to execute a program written is JavaScript is included in NodeJs. [(1)](https://www.zotero.org/google-docs/?8Keikk)

Before NodeJs was create, JavaScript was made so that it could only be ran on the browser but now with NodeJs, it is possible to run on a machine as a stand-alone application. This allows developers to use JavaScript like other scripting languages such as Python. [(1)](https://www.zotero.org/google-docs/?5UOG1y)

An event-driven, non-blocking I/O model is used by NodeJs which makes it lightweight and efficient. This is useful to real time applications that run across distributed devices and those that are data-intensive. [(1)](https://www.zotero.org/google-docs/?DoNhvy)

Not only that but rich library of various JavaScript modules are provided by NodeJs. Furthermore, for developing server side and network applications, NodeJs can be used as it is an open source, cross platform runtime environment. JavaScript can be ran within the NodeJs runtime on various operating systems such as Microsoft Windows, Linux and OS X. [(2)](https://www.zotero.org/google-docs/?JOGrfb)

NodeJs comes with many advantages such as:

**High Speed:** The code execution for NodeJs library is very fast as it is built on Google Chrome’s V8 JavaScript Engine. [(2)](https://www.zotero.org/google-docs/?fjY7Lo)

**Asynchronous and Event Driven:** A large advantage that NodeJs has is that all the APIs of NodeJs Library are asynchronous. Because the server moves onto the next API before even receiving the previous data to be returned. A notification mechanism of Events is used to get the response from the previous API call. [(2)](https://www.zotero.org/google-docs/?y5jPtf)

**Highly Scalable:** Although NodeJs uses a single threaded model but because it uses the model with event looping, the event mechanism can be used to help the server reposing in a non-blocking way. This essentially makes the server highly scalable as opposed to the limited threads created to handle requests. [(2)](https://www.zotero.org/google-docs/?V1O1La)

**No Buffering:** Instead of buffering all the datas, NodeJs applications outputs the data in chunks. [(2)](https://www.zotero.org/google-docs/?65bktL)

1. **ReactJs**

ReactJs is an open source JavaScript library. Developers use ReactJs to implement user interfaces. It is mostly used for single page applications and handling view layer for web and mobile applications. By using ReactJs, developers are able to create and reuse UI components. A component takes in parameters called props and returns a hierarchy of views to display via the render method. A description of what you want to render is returned and React takes that description and renders it to the screen. [(3)](https://www.zotero.org/google-docs/?CZNN4l)

Furthermore, ReactJs is handy for when developers wish to create large applications as React allows developers to change data without the stress of reloading the page. Other JavaScript libraries, such as AngularJs, can be used alongside React in MVC. [(3)](https://www.zotero.org/google-docs/?wOh5og)

1. **HTML**

HTML also known as Hyper Text Markup Language is a computer language that is used to create web pages. HTML is made up of elements and each element can be applied different pieces of text to give it different meaning in a document. Images and link can also be added to this document. Each element is represented by tags. [(4)](https://www.zotero.org/google-docs/?SlQCVh)

The tags are named and encapsulated by angle brackets. Each tag has a start and end tag to distinguish where the start and ending of the element is.

Using markup, a structure of web pages can be described by HTML. The browser renders the content of the body by using the HTML tags. The browser does not display the HTML tags but uses them to determine how the content should be displayed.[(5)](https://www.zotero.org/google-docs/?aBISnQ)

HTML elements are rendered alongside an external file called CSS. CSS is a cascading style sheet that describes how HTML elements should be displayed. It is used to define styles for webpages which includes layout, designs and also variations in display for different devices that has different screen sizes. The layout of multiple web pages can be controlled by CSS all at once. [(6)](https://www.zotero.org/google-docs/?tjg6QY)

**Database:**

**4. MongoDB**

MongoDB is a document database meaning is stores data in flexible, JSON-like documents. Each field can vary from document to document. Overtime, the data structure can be changed. The data is easy to work with as in the application code, the document model maps to the objects. Built in functions such as geographic distribution and high availability, horizontal scaling is easy to use as MongoDB is a distributed database at its core. Furthermore, MongoDB is free and open-source. [(7)](https://www.zotero.org/google-docs/?EP6nVy)

In MongoDB, a record is known as a document. It is a data structure which is composed of field and value pairs. It is similar to the JavaScript Object Notation objects but instead uses a variant called Binary JSON (BSON), which more data types are accommodated. The fields in the MongoDB document parallel to the columns in a relational database and the values that they contain can be a variety of data types. According to the MongoDB user manual, they contain data types such as other documents, arrays and arrays of documents. [(8)](https://www.zotero.org/google-docs/?scrriz)

A primary key must be included in the documents and used as a unique identifier. They are the basic unit of data in MongoDB. Equivalent to the relational database tables are collections which carry sets of documents and functions. Although data of any type can be stored in a collection, the data cannot be spread across different databases. [(8)](https://www.zotero.org/google-docs/?7EVJ1n)

Users are able to query, update data and conduct administrative operations by using the mongo shell. The mongo shell is an interactive JavaScript interface to MongoDB. MongoDB’s open source distributions contains standard components such as shell. Users are able to connect their running MongoDB instances to the mongo shell once the MongoDB is installed. [(8)](https://www.zotero.org/google-docs/?ilxm2n)

**Data Science Languages:**

Because a large area of this project involves Data Science and Machine Learning, it is important to choose the correct tools and suitable language when dealing with this part. Data Science is a concept used to tackle big data including preparation, data cleansing and also analysis. Whereas the practice of using algorithms to use data, learn from it and be able to predict trends for that topic is known as Machine Learning.[(9)](https://www.zotero.org/google-docs/?GUKMmJ)

**5. Python**

Due to the simplicity of the Python language, it is considered to be the best language for AI development. It is east to learn as Python’s syntaxes are very uncomplicated. Furthermore, Python allows users to implement many AI algorithms without any hassle. The development time is shorter compared to other languages such as Ruby, C++ or Java. Not just that but Python provide plenty of libraries such as Numpy which solves many scientific computations. [(10)](https://www.zotero.org/google-docs/?tCrPaq)Other useful libraries for Machine Learning include:

**5.1. SciPy**

SciPy contains a various number of mathematical functions and algorithms built on the Numpy extension of Python. Using SciPy the user is able to visualise and manipulate the data using high level classes and commands.[(11)](https://www.zotero.org/google-docs/?QyF9tG)

**5.2. NumPy**

When dealing with scientific computing alongside Python, NumPy is essential package to use. Not only does it contain a powerful N-dimensional array object, but it also has tools for integrating C/C++ and Fortran code. Furthermore, it contains useful linear algebra and random number capabilities. [(12)](https://www.zotero.org/google-docs/?JRmgjl)

NumPy is often used as an efficient multi-dimensional container of generic data in cases where scientific use is not needed. Data that are random can be defined which allows a large mix of databases to be integrated by NumPy. [(12)](https://www.zotero.org/google-docs/?A1of0w)

**5.3. Pandas**

When using the Python programming language, Pandas is a BSD-licensed library which includes easy-to-use data structures and data analysis tool.[(13)](https://www.zotero.org/google-docs/?QMDzSx)

While the Python language is useful for data “munging” and preparation, it lacks in data analysis and modelling. Therefore, Pandas allows users to complete data analysis in Python instead of changing it to a different language. [(13)](https://www.zotero.org/google-docs/?pgpYFZ)

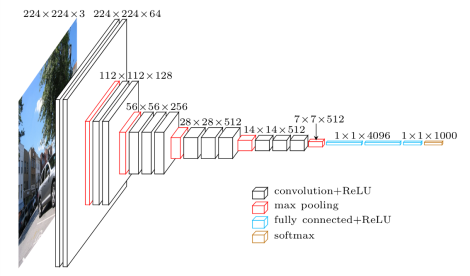
The highlights of the library are:

* + - *A fast and efficient* ***DataFrame*** *object for data manipulation with integrated indexing;*
    - *Tools for* ***reading and writing data*** *between in-memory data structures and different formats: CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format;*
    - *Intelligent* ***data alignment*** *and integrated handling of* ***missing data****: gain automatic label-based alignment in computations and easily manipulate messy data into an orderly form;*
    - *Flexible* ***reshaping*** *and pivoting of data sets;*
    - *Intelligent label-based* ***slicing, fancy indexing,*** *and* ***subsetting*** *of large data sets;*
    - *Columns can be inserted and deleted from data structures for* ***size mutability;***
    - *Aggregating or transforming data with a powerful* ***group by*** *engine allowing split-apply-combine operations on data sets;*
    - *High performance* ***merging and joining*** *of data sets;*
    - ***Hierarchical axis*** *indexing provides an intuitive way of working with high-dimensional data in a lower-dimensional data structure;*
    - ***Time series-****functionality: date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging. Even create domain-specific time offsets and join time series without losing data;*
    - *Highly* ***optimized for performance,*** *with critical code paths written in* [*Cython*](http://www.cython.org/) *or C.*
    - *Python with pandas is in use in a wide variety of* ***academic and commercial*** *domains, including Finance, Neuroscience, Economics, Statistics, Advertising, Web Analytics, and more.*

[(13)](https://www.zotero.org/google-docs/?uhZLKs)

According to David Himrod, Director of Optimisation & Analytics for AppNexus “Pandas is the perfect tool for bridging the gap between rapid iterations of ad-hoc analysis and production quality code. If you want one tool to be used across a multi-disciplined organisation of engineers, mathematicians and analysts, look no further.”

**5.4. VGG16 architecture**



VGG16 is a convolutional neural network architecture. This architecture is handy to use due to the visual presentation. The VGG16 architecture uses only 3x3 convolutional layers placed on top of each other increasing the depth, showing the simplicity of the network.(14)

*Figure SEQ Figure \\* ARABIC 2.6 – VGG Architecture*

Max pooling manages the reducing of volume size. Followed by a softmax classifier are tow fully connected layers, each with 4,096 nodes. Drawbacks include the training time being “*painfully slow*” and the weights being quite large, in terms of disc and bandwidth.[(14)](https://www.zotero.org/google-docs/?N8SuQX)

**5.5. Tensorflow 1.3.0**

TensorFlow is an open source library for numerical computational and large-scale Machine learning. A large amount of Machine Learning and Deep Learning models and algorithms are bundled together by TensorFlow, then made useful by way of a common metaphor. TensorFlow can train and run deep neural networks for image recognition. TensorFlow also supports production prediction at scale with the same model used for training.[(15)](https://www.zotero.org/google-docs/?9e1hkY)

Using TensorFlow, developers are able to create data flow graphs-like structures, which developers use to see how the data moves through a graph. TensorFlow can be implemented using the language Python, which is an advantage as the language Python is easy to learn and work with. The graph represents each connection or edge between nodes in a multidimensional data array along with mathematical operations. In TensorFlow, nodes and tensors are Python objects. [(15)](https://www.zotero.org/google-docs/?LF0plK)

One of the biggest advantages of TensorFlow is that TensorFlow deals with the small details behind the code. Developers don’t have to waste time figuring out the small details that don’t matter or figuring out how to link the output of one function to the input of another function. Instead developers can focus on the overall logic of the application. This is why abstraction is the biggest advantage that TensorFlow provides for Machine Learning development. [(15)](https://www.zotero.org/google-docs/?Rt8RNU)

**5.6. Keras - 2.0.7**

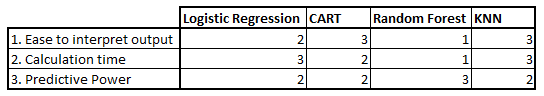
Although TensorFlow is a very powerful library, it can be difficult to use directly for creating deep learning models. The Keras Python library can provide a clean and convenient way to create the models on top of TensorFlow. Keras is high level neural networks API, written in Python. You can create models layer by layer by using the sequential API. Keras was developed to help implementing deep learning models fast and easy. [(16)](https://www.zotero.org/google-docs/?f6CuWk)

Benefits include:

* + - * + ***User friendliness.*** *Keras is an API designed for human beings, not machines. It puts user experience front and centre. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimises the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.*
        + ***Modularity.*** *A model is understood as a sequence or a graph of standalone, fully-configurable modules that can be plugged together with as few restrictions as possible. In particular, neural layers, cost functions, optimisers, initialisation schemes, activation functions, regularisation schemes are all standalone modules that you can combine to create new models.*
        + ***Easy extensibility.*** *New modules are simple to add (as new classes and functions), and existing modules provide ample examples. To be able to easily create new modules allows for total expressiveness, making Keras suitable for advanced research.*
        + ***Work with Python****. No separate models configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility.*

**5.7. K-Nearest Neighbours**

KNN, known as K-Nearest Neighbours, are generally used for both classification and regression. Three aspects are evaluated when considering which techniques to use. Based on a similarity measure, KNN classifies new cases and stores all the available cases. [(18)](https://www.zotero.org/google-docs/?YI4Nht)

* + - Ease to interpret output
    - Calculation time
    - Predictive Power

*Table 2.1 - K- Nearest Neighbour Comaprison*

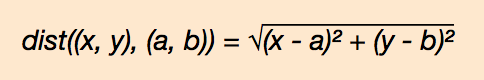
The KNN algorithm classifies a case by evaluating the majority vote of its neighbours. The case is allocated to its class based on how many neighbours it had in common during the measurement. For example, if K=3, the case is then allocated to the class of its nearest three or two neighbours. [(19)](https://www.zotero.org/google-docs/?k3rMgR)

In order to choose the best value for K, the data should first be inspected. Usually, the larger the value of K is, the more precise it is as the noise is reduced. Developers can use an independent dataset and cross-validation to decide on a suitable K value. [(19)](https://www.zotero.org/google-docs/?GINOCW)

**5.8. Euclidean Distance**

The Euclidean Distance algorithm is used when users are trying to determine the distance between two points. Using the coordinates (x, y) and (a, b), users are able to apply the Euclidean Distance algorithm and receiving the distance value. The Euclidean Distance algorithm is based on the Pythagoras Theorem.[(20)](https://www.zotero.org/google-docs/?XTXwB3)

The Euclidean Distance mathematical representation is represented as:



*Figure SEQ Figure \\* ARABIC 2.8 – Euclidean Distance Algorithm*

**5.9. K-Means**

When dealing with unlabeled data (ie. data without defined groups or categories). Developers normally uses K-means to determine the groups in the dataset. The number of groups are represented by K. Based on the similarities of different features for each data, data points are clustered together. The K-mean algorithm returns the centroid of the K clusters and also the labels for the training data. [(21)](https://www.zotero.org/google-docs/?pdcKxl)

**5.10. t-SNE for dimensionality reduction**

t-SNE also known as T-distributed Stochastic Neighbour Embedding, is a dimensional reduction technique that aids developers to identify relevant patterns.[(22)](https://www.zotero.org/google-docs/?Ikmb0Z) Dimensional reductions are a technique developer use to view a multi-dimensional data in two or more dimensions. Using t-SNE, developers are able to explore high-dimensional data. This algorithm can be used to map multi-dimensional data to two or row dimensions. t-SNE is able to preserve local structure giving it a high advantage.[(23)](https://www.zotero.org/google-docs/?Uwzwin) t-SNE is generally used for visualisation of high-dimensional datasets. Barnes-Hut approximations can be used to implement with this technique. [(24)](https://www.zotero.org/google-docs/?8tfKwH) Although it is not difficult to understand the patterns in regular charts, people with no knowledge of statistic can still find it hard to read them. Furthermore, if there are hundreds of features in a chart, users will have to study the charts before they are able to make sense of the data. Using t-SNE for dimension reduction, users are able to display the data in a detailed and clear manner. [(23)](https://www.zotero.org/google-docs/?KCLf8E)

**6. Web Frameworks for REST**

Representational State Transfer also referred to as REST, is an application program interface. Using HTTP requests, an application is able to use GET to retrieve data, PUT to change or update data, DELETE to remove it and POST to create data. Developers are able to provide interoperability between devices connected to the internet using HTTP methodologies. [(25)](https://www.zotero.org/google-docs/?eliTRb)

The REST model is particularly handy for cloud web service. It is as straightforward as altering the provided URL to connect through an API. If an API has the following features, it is considered as RESTful:

**Client** - Server: The client handles the front end where the user interacts with the application and the backend processing is handled by the server. Both can be replaced independently of each other.

**Stateless** - Data that are sent from the client are not stored on the server between requests.

**Cacheable** - To improve performance, clients is able to cache responses.

[(26)](https://www.zotero.org/google-docs/?zwhRvm)

**6.1. Flask**

Flask is a web framework. Using Flask developers are able to build a web application using the tools, libraries and technologies that Flask provides. Because Flask is a micro framework, it has little or no dependencies to external libraries. Although this means that the framework is light and has little dependency to update, developers may have to add plugins for some dependencies. [(27)](https://www.zotero.org/google-docs/?gn63w6)

Flasks dependencies are the following:

- Werkzeug: a WSGI utility library

- jinja2: a template engine

**6.2. Django**

Django is also a web application framework. Its free and open source. The framework is written in Python. A framework is a collection of modules that allow the development process to be easier. Instead of implementing websites from scratch, developers can create applications from an existing source. Django allows users to develop websites faster and easier.[(28)](https://www.zotero.org/google-docs/?NNI1sU)



**Flask Vs Django**

The main difference between Flask and Django is that Flask provides *simplicity, flexibility* and *fine-grained control.* Not only that but Flask allows you to implement how you want to. Whereas Django provides developers with an admin panel, database interfaces, an ORM and also directory structure for the application.[(29)](https://www.zotero.org/google-docs/?Io96J2)

Developers should choose flask if:

- They are focused on the experience and learning opportunities.

- They wish to have more control on which components to use.

Developers should choose Django if:

- They are more focused on the final product.

- The application is straightforward application ie. a blog or a news site.

[(29)](https://www.zotero.org/google-docs/?3swSS4)

Despite both frameworks supporting User Authentication, Rate Limiting and Relational Database Mapping for Rest API design, each framework contains different advantages.

**Development Speed**

Experienced developers are able to create smaller applications in short time frames due to Flask’s simplicity. Django provide developers with all the tools that need to implement. They are able to develop scalable, reliable and maintainable web applications at a reasonably short timeframe.[(30)](https://www.zotero.org/google-docs/?mzdxSY)

**Learning Curve**

Django has a steeper learning curve than Flask, therefore, Django allows developers to change their team halfway through the project or scale a new team if needed. Architectures and conventions are easier to understand with Django than with Flask. [(30)](https://www.zotero.org/google-docs/?Txk3Vt)

**Flexibility and Control**

One of Flask’s greatest advantage is its simplicity. Because there are no restrictions, developers are able to implement freely however they like. which makes it extensible and flexible. [(30)](https://www.zotero.org/google-docs/?XJTnqK)

**Maturity**

Django was released in 2005, which makes it a very mature framework. It has assembled many extensions and plugins. Flask however was introduced in 2010 therefore in comparison has a lesser range of options available to it. [(30)](https://www.zotero.org/google-docs/?kfNWtI)

**ChartJs**

ChartJs is a JavaScript library that allow developers to display a graph. The developer input the data and a chart can be generated. ChartJs uses HTML canvas element to draw different type of charts. Developers can import the specific chart they want to implement and ChartJs would use the information to plot the graph. Developers are able to change the way the chart looks and design it how they like.

The ChartJs library is very small in size as it has no dependencies or weight. It is only 11kb when minified, concatenated and served gzipped.

The main advantage with ChartJs is that it is responsive. ChartJs can adjust based on the space availability. Not only that but ChartJs provides extensive and clear documentation which makes implementation easier.

A screenshot of a cell phone

Description automatically generated

<https://www.sitepoint.com/creating-beautiful-charts-chart-js/>

**ExpressJs**

ExpressJs gives JavaScript backend functionality, this allow developers to implement software with JavaScript on the server side. Developers are able to use NodeJs to implement a server side application then publish the NodeJs app as website using ExpressJs.

Due to the fact that NodeJs was not intended to build websites, the Express framework can layer in built-in structure and functions needed to build a site. Express is a lightweight framework used to give developers extra, built-in web application features and the Express API without overriding the NodeJs platform.

Express can be used to build single-page, multi-page and hybrid mobile and web apps. It is also used for common backend functions for web applications and API.

Jade and EJS are the two templating engines that Express come with. It facilitates the flow of data into a website structure. Model View Controller (MVC) is supported by Express which is a helpful way to build a website in a model driven format.

<https://www.upwork.com/hiring/development/express-js-a-server-side-javascript-framework/>

**PuppeteerJs**

*“Puppeteer is a Node library which provides a high-level API to control headless Chrome or Chromium over the DevTools Protocol. Puppeteer runs headless by default buy can be configured to use full (non-headless) Chrome or Chromium.”*

Most things that people can do manually in the browser can also be done using Puppeteer. Developers are able to use Puppeteer to generate screenshots and PDFs of pages, also automate form submission, UI testing and keyboard input. Furthermore, puppeteer can be used to test Chrome Extensions.

<https://developers.google.com/web/tools/puppeteer/>

<https://github.com/GoogleChrome/puppeteer>

<https://www.npmjs.com/package/puppeteer>

**CronJs**

Developers use CronJs when a certain process needs to be executed on a schedule. It can be done using the Cron syntax. When a schedule job triggers, a function is executed. An external job to JavaScript can be executed using the *child\_process* Furhtermore, the CronJs library also allow developers to supply a Date object. The callback is triggered using this Data object.

Cron Job help developers focus on the implementation of the software rather than trying to remember to run a specific function at a specific time.

<https://www.npmjs.com/package/cron>

**VGG19**

VGG-19 is a convolutional neural network that is trained on more than a million images from the ImageNet database. Over a thousand images can be classified such as, shoes, keyboard, pencil and many animals using VGG19. The VGG19 neural network is 19 layers deep and as a result the network has mastered rich feature representations for a wide variety of images. The image input size is of 224-by224.

<http://www.image-net.org/>

<https://www.mathworks.com/help/deeplearning/ref/vgg19.html#bvmdok9.mw_6dc28e13-2f10-44a4-9632-9b8d43b376fe>

**MD5**

Message Digest algorithm 5, also known as the MD5, was invented by Ronald Rivest. The MD5 hashing algorithm is a one way cryptographic function. The way it works is it takes a message of any length, then returns the message as a fix length digest value to be used for authenticating the initial message. The input message could be of any length but the output returned is always of fixed length.

One of MD5’s main use is for cryptographic hashing in order to verify the contents of a message or file after transfer. To do this, MD5 produces a checksum on both sets then comparing it make sure they are the same.

The MD5 algorithm provides a 32 hexadecimal number and it is built to be non-reversible, this means that you cannot examine the checksum and identify the initial inputted message.

<https://searchsecurity.techtarget.com/definition/MD5>

<https://www.makeuseof.com/tag/md5-hash-stuff-means-technology-explained/>

<https://www.lifewire.com/what-is-md5-2625937>

**Web Hosting**

Web hosting is a service that allows developers to publish and deploy their web application onto the Internet. A web host service provider , such as AWS and Heroku, is a business that supply the services and technology necessary for the webpage to display on the Internet. Websites are hosted on a server which is a special type of computer made to store and host these websites.

Users are able to access the websites by entering the website address of the domain into their browser. This connects the server to their computer and the website is delivered to the users through the browser.

Both the React Native application and recommender system need to be hosted. The React Native application can be accessed by any user on any browser once the application is hosted. The front-end of the application is displayed to the user and API calls are made to the backend once a function is called upon by the user.

The recommender system is also hosted on the cloud and communicated through RESTful calls. For future development, such as development on a device, can be done without any further development. Hosting the recommender system on the cloud also allows for scaling.

By doing this, we are able to expand the system to different platforms, calling the same webservice. This stops us from having to implement a localized version of the recommender system.

<https://www.website.com/beginnerguide/webhosting/6/1/what-is-web-hosting?.ws>

**AWS**

Amazon Web Services, or AWS, is a cloud computing platform provided by Amazon. Not only can developers, businesses and non-profits use AWS to deliver their websites and web applications but they also have access to virtual clusters of computers made available through the internet.

AWS holds many advantages such as:

**Broad platform support:** AWS provides and supports SDKs for many popular platforms such as PHP, NodeJS, .Net, Ruby and Java. Developers are also able to use any CMS they prefer, including Drupal, WordPress and Joomla.

**Datacentres worldwide:** With AWS, developers can have a datacentre or CDN hosting their website in any geography they wish.

**Scalable from day one:** It is clear that web traffic constantly fluctuate. By using AWS, developers can worry less about the scalability as AWS can adjust to it,

**Flexible pricing models:** There are no up-front costs or long-term contracts with AWS. You are charged for the resources you use. Users are able to choose a fixed monthly or a pay-as-you-go pricing.

This project can make use of their Amazon Elastic Compute Cloud (Amazon EC2). Amazon EC2 is a web service that supplies secure, resizable compute capacity in the cloud. This was designed to help make web-scaled cloud computing easier for developers. The recommender system can be hosted on Amazon EC2 where the code is stored then using Amazon’s API Gateway console to allow for access using RESTful APIs.

<https://aws.amazon.com/websites/>

<https://aws.amazon.com/ec2/>

**Heroku**

Heroku is a cloud platform service that allow developers and businesses to deploy, manage and scale applications. It is flexible and easy to use, offering developers a straightforward path to getting their application onto the Internet.

Heroku is fully managed, allowing developers to focus more on the product in absent of the distraction of maintaining servers, hardware or infrastructure. Services, tools, workflows and polyglot are all provided by Heroku.

Heroku is dispersed and remote, therefore making it faster, more secure, and much more easily scalable than traditional private servers can be. Heroku allow developers to work in any of the eight languages including Ruby, Python, NodeJs, Java, PHP, Go, Scala and Clojure.

Other advantages of Heroku include:

**Productivity:** Every detail of Heroku is designed to maximise efficiency and save developers time. It is easy to use, powerful as well as intuitive.

**Support:** The Heroku Operational Experience (Opex) give great support, including troubleshooting tips and quick diagnostic tools to help developers get to the root of the problem quick.

**Ecosystem:** Developers are able to extend their applications using a wide range of add-ons from 3rd party cloud services.

**Cost:** Companies can save a lot of money with Heroku’s platform as it removes the need for expensive infrastructure.

Using Heroku would mean a Dyno will be created to hold the recommender system. Then using Flask or Django, RESTful APIs can be created and used.

<https://www.heroku.com/about>

<https://courses.telegraph.co.uk/article-details/192/what-is-heroku/>

**Digital Ocean**

Digital Ocean is the last potential web hosting service researched. Digital Ocean is a cloud computing vendor that offers an Infrastructure as a Service platform. It gives developers a reliable and easy to use virtual servers and object storage. It is especially popular with open source developers. Its main competitors are Amazon Web Services and Google Compute Engine. The two main classes of products that Digital Ocean offer are Droplets and Storage.

Droplets is a scalable computer service that provides developers with not only add-on storage and security but also monitoring capabilities that can run production applications with ease. Droplets is available in 8 different locations around the world and also resizable at any time.

Digital Ocean also offer spaces which are object storage that can be used to store and deliver any amount of data to end users and applications. This is a simple and cost effective path to take, it also uses an easy drag-and-drop UI.

<https://miloszkrasinski.com/the-benefits-of-digitalocean/>

1. Other Relevant Research Done

**Testing Tools**

Throughout the course of this project, there is no doubt that a lot of testing have to be done. Because the project has a lot of different functions, it is hard to implement it all using one technology or IDE. It is often difficult to test whether different components work well together or if the current state of the project is working properly. Using various testing tools developers are able to do so.

1. **Postman**

API are a very important for developers as they allow users to not only execute actions without the use of an application’s GUI but also allow users to transfer and share data between two separate applications. APIs is a way to test and execute certain functionalities of their application without great efforts.

Sometimes it is difficult to use APIs regularly as developers may need hundreds of APIs to test. It is hard to test for security and exception handling with an overwhelming number of APIs, as it can be tough to remember their correct headers, authorisation credentials and address.

Postman is a powerful HTTP client for testing web services. APIs can be tested and developed using Postman by allowing developers to put together both simple and complex HTTP requests. Postman is available as both a Google Chrome Packaged App and a Google Chrome in-browser app. At publication time, both versions of Postman had a high number of more than 348,000 unique users, making the app one of the highest-rated productivity applications in the Chrome Web Store. [(31)](https://www.zotero.org/google-docs/?gYUkvJ)

Postman’s advantages include:

**Easy Interface** - Postman’s interface is clean and intuitive. Most features are attainable within one click.

**History** - Postman keeps a record of the API calls allowing developers to access the response viewer at a later time.

**Easy Organisation** - Developers who are dealing with large number of APIs are able to organise them into folders called “collections” and further into sun-collections if needed. This method of organisation allows developers to find and access these API requests quickly.

**Response Viewer** - One of Postman’s most crucial feature is the response viewer. Developers are able to view the API response body and header in tabs. Adjacent to the tab is the status and the time code. The response viewer allows developers to view the body of the response in three different formats: Pretty, Raw and Preview.

[(31)](https://www.zotero.org/google-docs/?Z4XZQ3)

A screenshot of a social media post

Description automatically generated

1. **Robo 3T**

This project will be using MongoDB as part of the database, therefore Robo3T can be used to test the database. Robo3T is an open source GUI for MongoDB users. Robo3T is available in most platforms hence there is a low compatibility risk.[(32)](https://www.zotero.org/google-docs/?HMIybX)

Robo3T embeds the same engine and environment as the MongoDB shell. Robo3T is able to examine the semantic of the code and also executes it in an internal JavaScript VM. Additionally, Robo3T is fully asynchronous. Because the operations that are performed on MongoDB are asynchronous, the main application thread is not blocked. [(32)](https://www.zotero.org/google-docs/?a5MG0E)

A close up of a logo

Description automatically generated

1. Resultant Findings and Requirements

After researching all the possible technologies, for this project the following technologies will be used:

* Application
  + NodeJs
  + ReactJs
  + HTML / CSS
* Python
  + VGG19
  + KNN
  + Flask
* Database
  + MongoDB

The front end of this application is implemented in HTML and CSS due to its popularity and simplicity. Because HTML is widely used, there are many documentation and examples of how to implement it. Not only that but HTML is supported by every browser therefore there is no risk with compatibility. ReactJs will also be used for the front end. Because the components are reusable, the code will be easier to implement and maintain. Also due to its popularity, there are many useful tutorials that are easy to learn from. Puppeteer is used to scrape information on the website and using those scraped data, it will be displayed on the application.

The back end of the project will consist of NodeJs. If a user has an account, their details will be stored using MongoDB. The price of each item will also be stored using MongoDB, in order to load and display them later on. To get the item details, the application will scrape the data using Puppeteer from the NodeJs Library.

For the Recommender System, Python was the language chosen due to its numerous mathematical package and libraries such as Numpy, Pandas and SciPy. These libraries are fundamental for the Machine Learning in the Recommender System. Not only that but other libraries such as Keras and TensorFlow are available for the implementation of the system. **- \*\*\*\* EDIT \*\*\*\***

**\*\*\*\* Heroku was chosen … etc \*\*\*\*\***

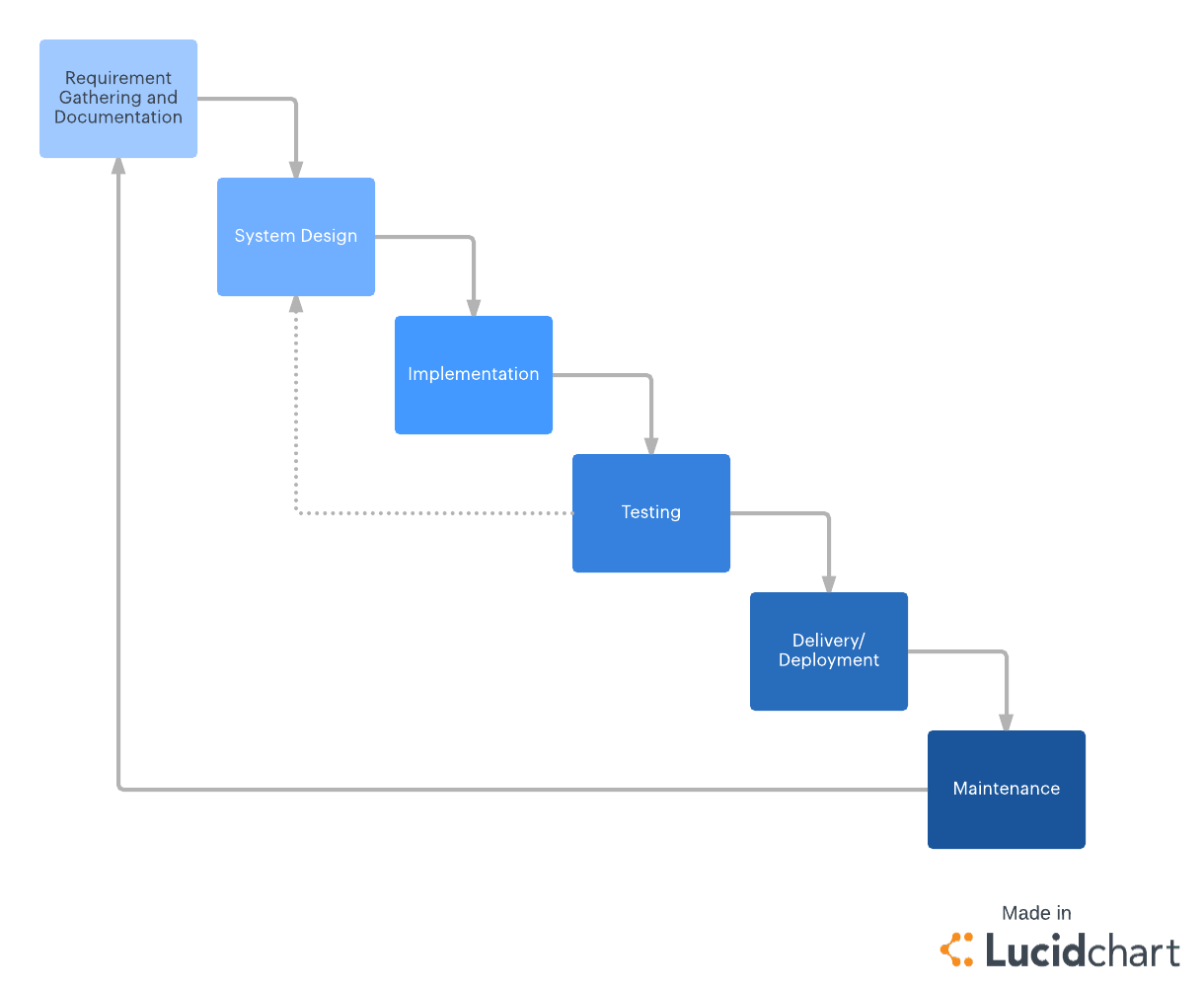
**Design**

1. Approach and Methodology

There are many different design methodologies to use when building an application. Choosing the right approach and project plan is essential when implementing a large-scale project, especially when the application is being built by one person. In this section the following methodologies will be discussed:

* + Waterfall Model
  + Agile

**Waterfall Model**

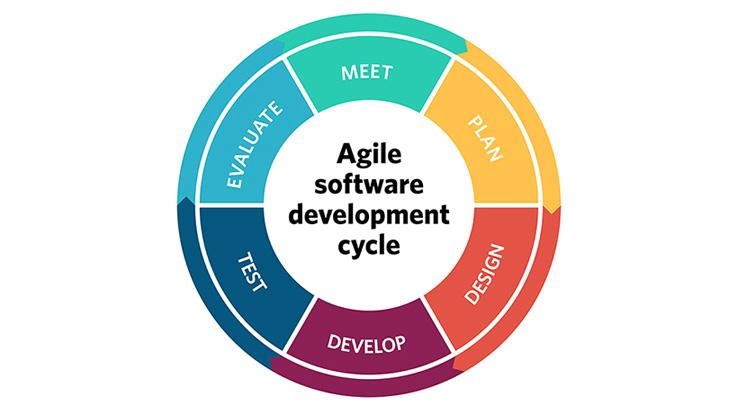


*Figure 3.1 – Waterfall Model*

The Waterfall model was first introduced in a paper published in 1970 by Dr Winston W. Royce. The Waterfall model is a sequential, linear process of project management. The Waterfall model separates the project into distinct phases. The project moves onto the next phase once the current phase has been delivered. No phase begins until the previous phase is complete, and each phase’s completion is terminal therefore it is not possible to return to the phase before. The Waterfall model only allow revisiting previous phase by starting over at phase one. [(33)](https://www.zotero.org/google-docs/?BRdHMw)

The phases usually consist of Requirements Analysis, System Design, Implementation, Testing, Delivery and finally Maintenance. Although the Waterfall model is capable of helping developers build a well implemented project, it lacks flexibility. Some functionalities flaws can only be discovered during testing therefore it is difficult to predict all needs in advanced. For that reason, the Waterfall model should only be used when it is unlikely to change or adapt or when time isn’t a massive constraint. [(33)](https://www.zotero.org/google-docs/?qsi0TY)

**Agile Development**



A more modern approach is Agile Development. Agile is an iterative approach to development. Usually Agile is team based but it can be adapted for one person. The Agile approach focuses on the rapid delivery of a fully functional application. Instead of making tasks, Agile divides the project into phases called ‘sprints’. Each sprint has a specific time duration and a list of deliverables, planned the beginning of the sprint. Agile allow the requirements and solutions to change throughout the development process. [(34)](https://www.zotero.org/google-docs/?AdqxZO)

*Figure 3.2 – Agile Development Cycle*

The iterations included in Agile are Meet, Plan, Design, Development, Testing and Evaluation. This method allows each project to be handled differently and tailored to best suit the requirements consequently optimising the implemented code. [(34)](https://www.zotero.org/google-docs/?vjhRsU)

Advantages include:

* + Users and customers are able to test and check out functionalities at an early stage, allowing them to make decisions and changes throughout the development process.
  + Customers gain a strong feel of ownership as they are able to input decisions. Customers can work directly with the development team.
  + Recommendations and decisions are likely to come from customers therefore development is more user focused.
  + Developers are able to have a clear understanding of the importance of the client’s business and can deliver features that give most value to the business.

[(34)](https://www.zotero.org/google-docs/?tiwjH9)

A subset of Agile is Scrum. Scrum is a lightweight process framework for Agile development. During the implementation of this project, a variation of Scrum, modified to suit a one-man team was used. [(35)](https://www.zotero.org/google-docs/?Vtb7ep)

Developers using Scrum generally create a product backlog which is a prioritised list. At the beginning of the sprint, a meeting is set up by the team and during this they discussed which backlog items will be worked on. A time frame is set to get their designated work done. Each day throughout the sprint, the team members will meet, discuss and assess their progress. At the end of the sprint, each member should have their designated work mostly done and reviewed. This is then repeated until the project is completely developed.

For projects being developed by a single person, the backlog is created, and prioritised items are chosen by one person. This project’s sprint are 2 weeks long, with a goal of having a demonstrable feature available at the end of each sprint. This approach allows testing to be done to each feature as they are developed resulting in a more success efficient code.

1. Other Design Documents

**ER Diagram**

A close up of text on a white background

Description automatically generated

*Figure 4.2 - ER Diagram*

The above ER diagram is a representation fit the MongoDB database. The tables are defined as follows:

**User:**

The User table stores all the information on the users. A new user is inserted when a new user registered. This user information is used during the login process. The User table contains:

* + Username: The name of the user
  + Email: The user’s email address
  + Password: The user’s password

**UserItem:**

This table holds all the item information that a user is pending. The link will be deleted either when it expires or when the user receives a notification that the item is back in stock. The UserItem table consists of:

* + Name: Name of Item
  + Link: Link of item used to scrape the item details
  + Size: The size that the user selected to be notified
  + Price: Price of item
  + Image: URL of item image

**Item:**

The Item schema stores the details of an item for a non-registered user. Because it is not necessary for the user to register, the item must be stored separately to the user’s items. The Item table consists of:

* + Email: The user’s email
  + Username: Name of the user
  + Link: The link of the item the user inserted
  + Name: Name of item, retrieved by the link
  + Size: The size that the user selected
  + Price: Price of item

**Links:**

This table stores the links. The application scrapes every single shoe link from the ASOS and saves it to the Links schema. This table consists of the following:

* + Link: The link of an item
  + WebsiteName: In this case the website would be ASOS. Later when the system support more websites, it will be easier to keep track of which website the link belongs to.
  + Price: \*\*\*\*EDIT\*\*\*\*

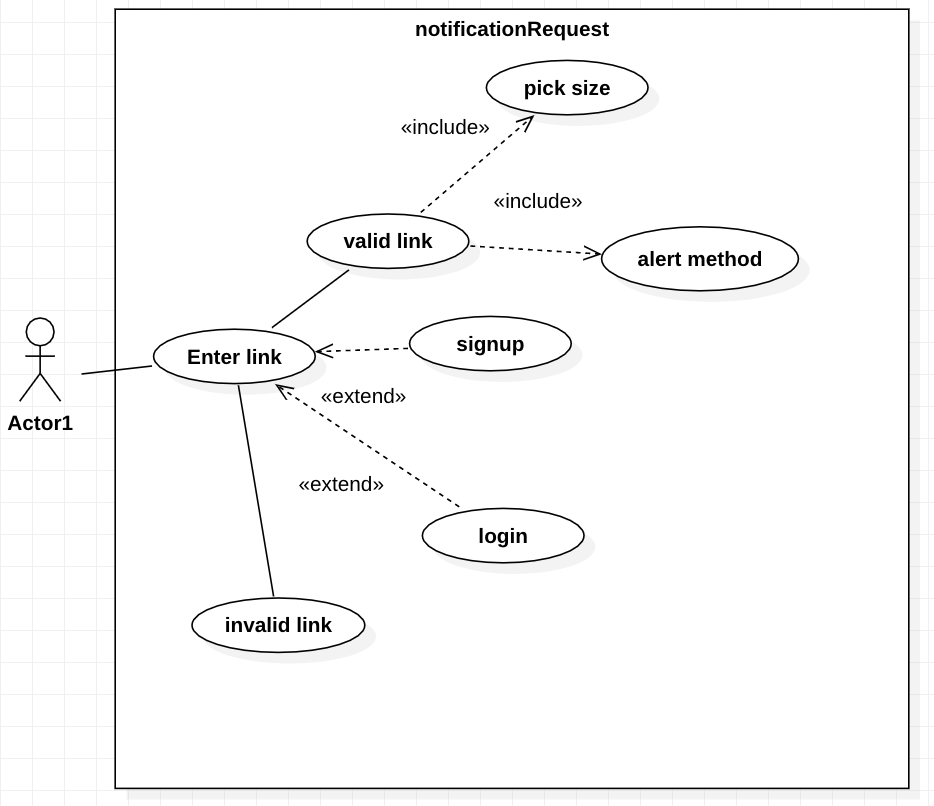
**Prices:**

The Prices table holds the all the prices of an item. When the item is scraped every day, it will check if the link match and if they do then the price of that day will be inserted to the Price table. This table can then be used to display the price history of an item. The Price table consists of:

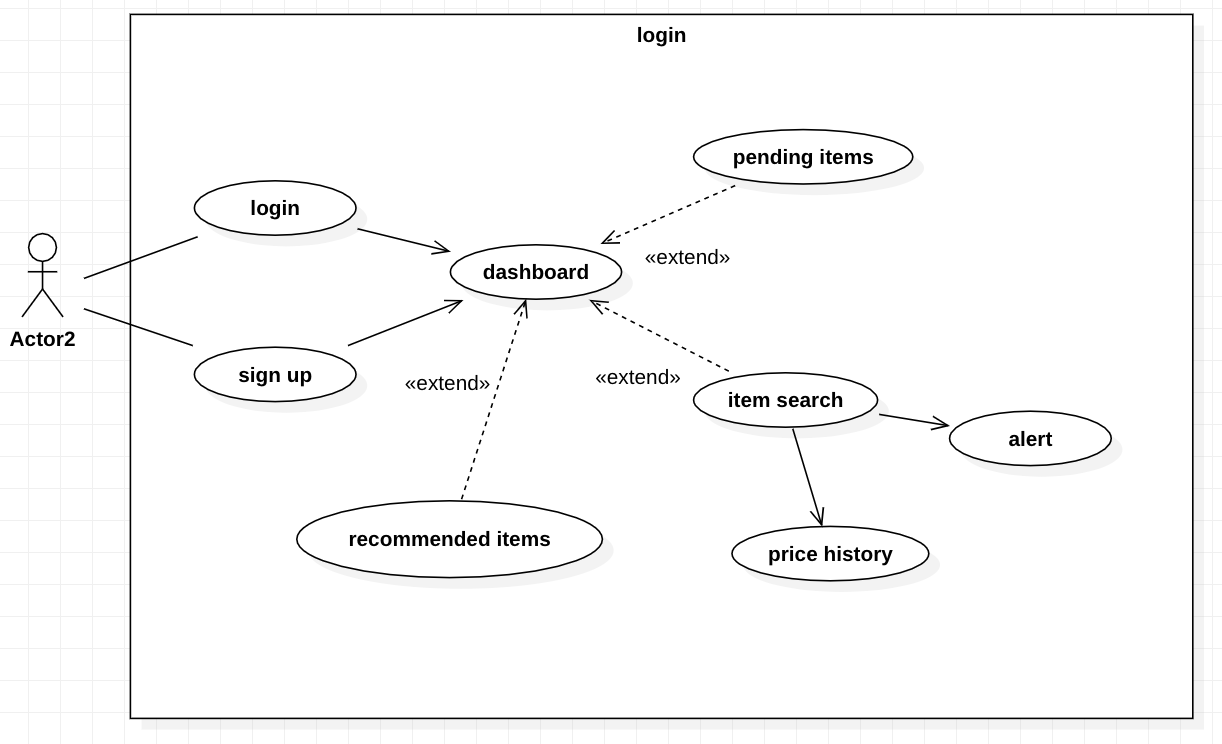
* + Price: Primary key of the Price Table
  + Date: The date when the price was inserted

**Use Case Diagram**

*Figure 4.3 - Use Case Diagram 1*

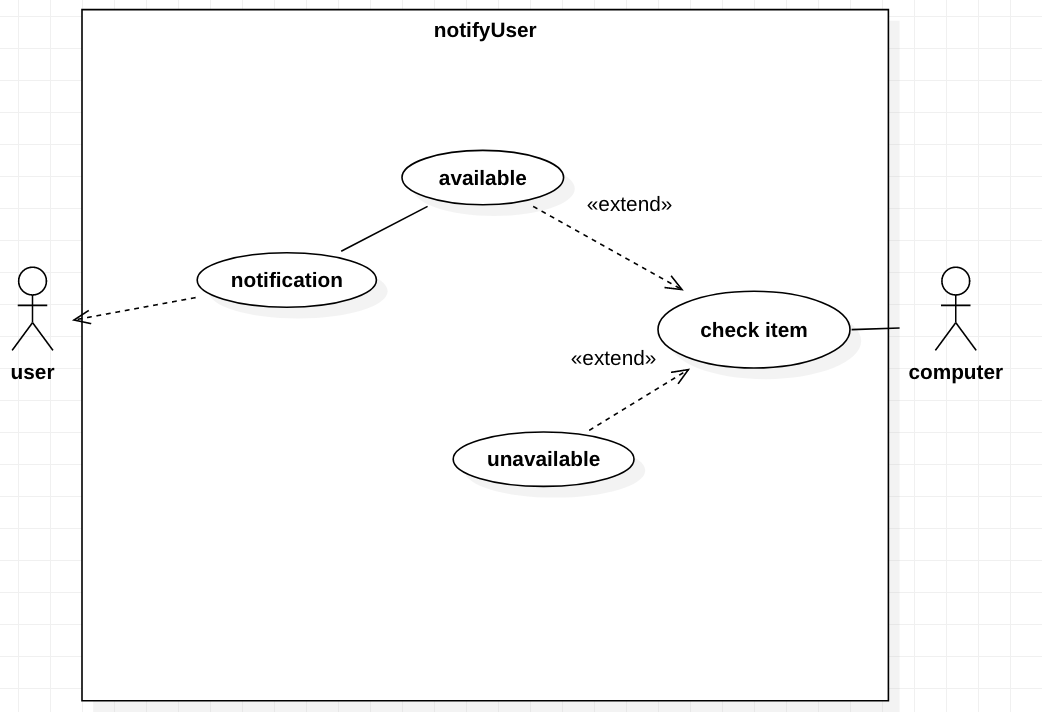


The use case diagram above shows the user entering a link of an out of stock item. If the link is valid, the user then picks a size and an alert method (ie. email or text messages). The user also has the option of login or sign up.



*Figure .4 - Use Case Diagram 2*

In this Use Case diagram, the user can login or sign up. Both leading to the dashboard which displays the pending items that the user has, recommended items made by the system and user can also search for an item to either view price history or request an alert. The above functions are the only features that the user can interact with, the user has no awareness or access to the inner workings of the application and recommender system.



This Use Case diagram demonstrates how the user will get a notification. The system checks if an item is in stock or still out of stock. Once the item is back in stock, the system will notify the user.

*Figure - Use Case Diagram 3*

Original UI Mock-ups:

Home Page:

A screenshot of a cell phone

Description automatically generated

Login Page:

A screenshot of a cell phone

Description automatically generated

Register Page:

A screenshot of a cell phone

Description automatically generated

User Dashboard Page:

A screenshot of a cell phone

Description automatically generated

Recommender Page:

A screenshot of a cell phone

Description automatically generated

Choose Size and Price History Page:

A screenshot of a cell phone

Description automatically generated

Final UI Outcome

Home Page:

A screenshot of a cell phone

Description automatically generated

Login Page:

A screenshot of a cell phone

Description automatically generated

Register Page:

A screenshot of a cell phone

Description automatically generated

User Dashboard Page:

A screenshot of a social media post

Description automatically generated

Recommender Page:

A screenshot of a social media post

Description automatically generated

Choose Size / Price History:

A screenshot of a cell phone

Description automatically generated

Source Code Layout

ReactJS: - Pages

A screen shot of a computer

Description automatically generated

Above is the layout for the React Native code. The code is separated to serve its concern. The pages are divided into directories relating to the Navigation Bar, Signed In pages, Signed Out pages. Other activities are placed in the main project folder, such as the Login and Register page.

**Pages > NavigationBar**

* + **Header**: This controls whether the Signed-Out navigation bar or the Signed-In navigation bar is displayed. Checks whether the JSON web token is present before determining which navigation bar to display. If token is present, it means that the user has logged in successfully and the logged in navigation bar will display. Once the user clicks logout, the JSON web token is deleted and the logged-out navigation bar will display.
  + **History**: This page is called when a new route is needed to be pushed. When the current page changes to a new page, it needs to be redirected, which can be done using history.
  + **Root**: Allows the navigation bar to display in every page along with the content of the page without having to repeat the code in every page. This means the code is dry and there isn’t any repetitive code.
  + **SignedInLinks**: The header file checks whether a JSON web token is present. If present, the navigation bar designed for logged in users is displayed. The logged-in navigation bar allows users to navigate to the dashboard page, the recommended items page and also allow users to log out.
  + **SignedOutLinks**: The header file checks whether a JSON web token is present. If absent, the navigation bar for home page is displayed. The logged-out navigation bar consists of the home button which navigates to the home page, the login and register buttons which navigates to the login and register page.

**Pages > SignedIn**

* + **Recommend**: This page displays the recommendation made for the user logged in. Recommendations are made based on what the user adds to their item list. The recommender system creates the best suited recommendation and the output is displayed onto this page.
  + **User**: This is the user’s dashboard. It displays the items that the user is waiting for to come back in stock. The expired items are also displayed, the user has the option to renew the item.

**Pages > SignedOut**

* + **ChooseSize**: This page is redirected when a link is entered in the home page. The item’s information is scraped and displayed to this page. The user then chooses a size they would like to track. Lastly the user can enter their details and the item along with their details is stored to the database.
  + **Home**: This is the website’s home page. The user can enter a link or upload an image for a recommendation here.
  + **Recommendation**: After the image is uploaded, the recommendation for that image is displayed onto this page.

**Backend**:

The backend directory consists of all the files that make up the main functionality of the application.

**Server**

A close up of a logo

Description automatically generated

* + **.env**: This file is used to set the environment variables and values for quick access.
  + **Server**: The server file consists of all the HTTP requests. These HTTP requests include retrieving data from the database for login and sign up etc. The server file is also where the database is connected. Other HTTP requests include retrieving scraped data from a given link.

**Cronjob & Mail**

A close up of a logo

Description automatically generated

* + **Cronjob**: The crobjob is running in the background and when the timer goes off a function is called to perform some type of action. The cronjob goes off every hour to check for the availability of every item in the database. The cronjob also scrapes the prices for the price history.
  + **Mail**: This file is in charge of emailing users in certain scenarios. The user is notified when an item is back in stock and also when their item has expired.

**Scrape**

A close up of a sign

Description automatically generated

**Scripts > checkItemAvailability**

* + - **checkAvailability\_asos**: This file contains the functions necessary for checking the availability of an item. These functions only work for the ASOS website. For future implementation, other websites can be added using the same concept. This allows less work to be put into future implementations as each file have their own function and will not clash if something is changed.

**Scripts > scrape**

* + - **scrapeImage**: Scrapes the images off the website in order to perform a recommendation on the image.
    - **scrapeItems**: Scrapes the item’s details such as name, sizes, price or image URL.
    - **scrapeLinks**: Scrapes each item’s link along with the prices and stores it into the database for price history.

**Helper**: Every file in the Scripts directory need to open the browser in order perform their action. The helper file allows the scripts to call and open the browser instead of having to implement it in every file. This allows the code to be dry and less repetitive code is present.

**MongoDB - Models**

A black sign with white text

Description automatically generated

* + **items**: Item schema used to store all information regarding an item and a non-registered user.
  + **links**: Link schema used to store all the links and prices that are used for the price history
  + **users**: User schema used to store all necessary information on a user and all the items they are waiting for.

**Python – Recommender System**

The following directory holds the code to the recommender system.

A screenshot of a cell phone

Description automatically generated

* + **DB**: This folder contains all the images used to plot a recommendation for the user. The images in this folder were also used to train the classifier to build a model. The images are split up into their classes, (i.e. Men’s Jumpers, Women’s Dresses). Each image is either in the .JPEG format or .JPG format. There are 18 classes and each class contain roughly around 700-1000 images.
  + **Output**: This directory contains all the recommendations that were made after the code was ran. These recommendations are displayed to the users
  + **Imagenet\_utils**: External library
  + **KNN**: Find the top three most similar images using the imputed image from the user.
  + **Plot**: Plots the top three recommendation onto a graph. This graph can be displayed to the user.
  + **Recommend**: Imports all the files in the directory. Using the functions, a recommendation can be created for the user.
  + **Sort**: Finds the first unique k elements, based on lowest distance, of a list.
  + **Vgg19**: External library

\*\*\*\* Expand – 2 pages \*\*\*\*

Demonstration and Feature List

The demonstrable features for Shopaholic include:

* 1. Registering Users:

A user is able to sign up using just their name, email and provided password. This information is stored in the MongoDB database. The password that the user provide is encrypted using the ‘MD5’ hashing method, which cannot be read from the database directly.

Given the password, the hashing returns a 32 hexadecimal number that is non reversible.

A screenshot of a cell phone

Description automatically generated

* 1. Login

Once the user is registered, the user can login to view and add items of their choice. The dashboard provided helps user see which items they are currently tracking. The user also has the option to speculate a list of recommendations made specifically for the user.

* 1. Track and Manage Items

A user is able track an item by entering a URL to an item and selecting a size. The user can always delete the item if they are no longer interested in the item. Each item added has a time span of two weeks. Once the item reaches two weeks, the item will be moved to the expired item section. The user can renew the item which will move it back to the dashboard. If the user doesn’t take action within an additional week, the item will be deleted and removed from the user account.

* 1. View Price History

When a URL of a shoe is entered, a price history graph is displayed. The prices are stored in the Database alongside the date. A HTTP request can be called to retrieve the data and display it in the front end. The graph shows the price of the item at a certain date. The user is able to use this to determine when is the best time to purchase the item. Not only that but the user can also see if the item is really on sale or if the store raised the price before the sale.

* 1. User Notification

The user receives an email once the item that they are tracking is back in stock. The email consists of the item details and the URL in which the user can purchase the item from.

The user also receives an email when an item that they are tracking expires, the email lets the user know that the item is no longer being tracked and a link is available for the user to renew the item. Once the user clicks on the link, the item will be put back to the item list and removed from the expired items list.

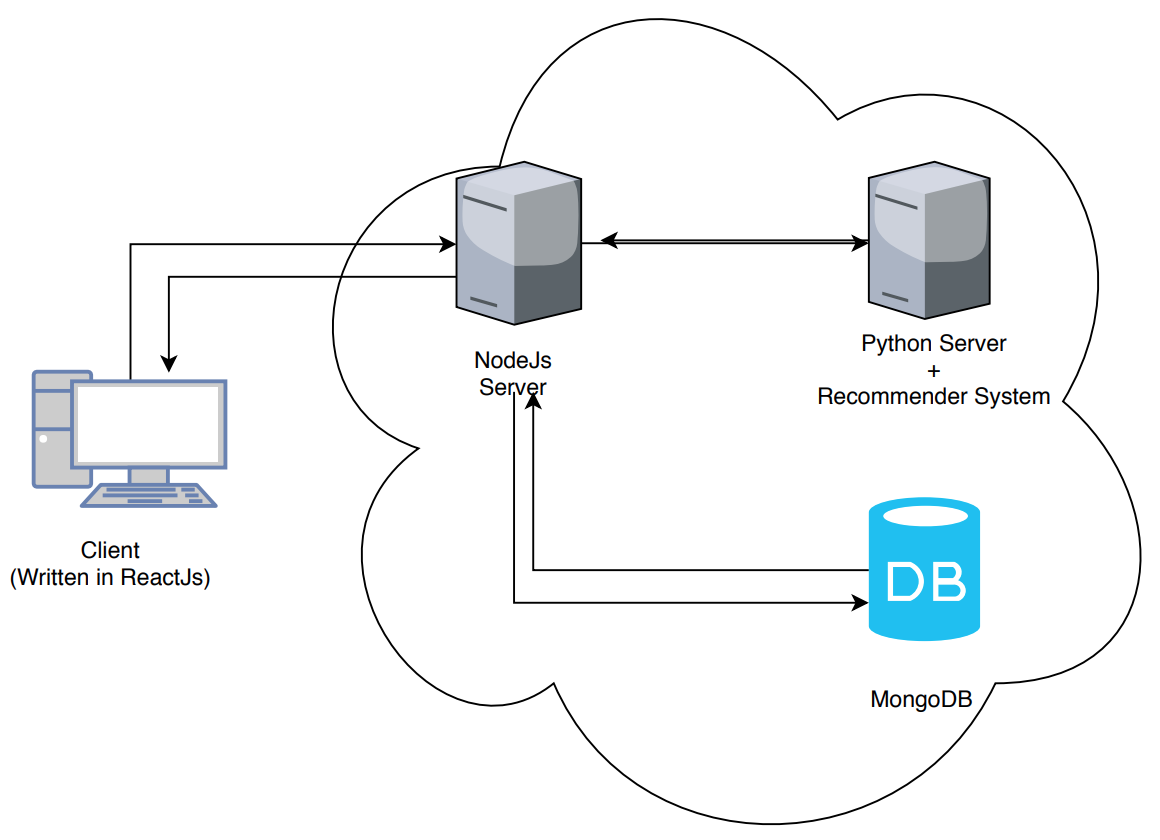
* 1. Recommender System

The last demonstrable feature is the recommender system. The system is able to use the image that the user provides, then plot the similarity of that image to the existing images stored in the database. The user can also enter a link to a product with a clear image displaying. The system scrapes the image from the URL provided and match it to the most similar item then displaying it to the user.

This feature not only allow the user to receive an item that they like but also potentially save them a lot of time from having to look for the similar item.

\*\*\*\* Expand on this (ie. Using HTTP request, the response is in what format \*\*\*\*

5. Architecture and Development

1. System Architecture Overview 

*Figure 4.1 – System Architecture*

All user interactions are through the web application. Anything the user clicks, NodeJs will display it in the front end. When the user is doing any account specific exchanges such as logging in, registering as a new user, checking their list of pending items or viewing the price history of an item, the application uses a NodeJs library called ExpressJs to make API calls to communicate with the database.

When the user wants a recommendation (ie. recommending a similar item), NodeJs will save that image to the database and the Python script will run to make a recommendation to the user. That recommendation will be stored in the database which can be retrieved making API calls to the database.

\*\*\*\* Expand on this \*\*\*\*

\*\*\*\* Summary \*\*\*\*

\*\*\*\* Primary Development \*\*\*\*

20 pages

\*\*\* External API / Library \*\*\*\*

Vgg16/vgg19

Imagenet\_util

Chartjs

PROJECT PLAN

Changes to initial plan

Although image recommendation was the main idea, the initial plan for Shopaholic included other functions such as price comparison and price prediction. Due to the time constraint, it was better to focus on image recommendation, making sure it was fully functional rather than have a system that had loads of half implemented functions.

Having the price comparison would mean the system would have to scrape the price for multiple websites that had matching items. The prices would then be stored into a Database. The Database would increase to an unmanageable size very quick due to storing every item of multiple websites.

The price prediction would’ve been a great additional feature but due to lack of data it would be difficult to achieve an accurate prediction. The prices normally remain the same for a long period of time then occasionally dropping to a discounted price. This isn’t enough information to train an accurate model.

However, the idea for the recommender system remained the same from the beginning. The only change was that the recommended image is not paired with a link. The images that were recommended were retrieved from the Database. Due to the nature of how it was scraped, the images were not stored with the link.

Possible Iteration if repeated

If given another chance to implement the system again from scratch, more time would definitely be allocated to the recommender system. A lot of time was spent on implementing the scraping functions and checking the availability of the items.

I would also spend more time on testing. Because testing was not taught to us, a lot of time was spent investigating the best testing method to use rather than implementing the testing itself.

Add more websites \*\*\*\* Expand \*\*\*\*

\*\*\*\* Conclusion \*\*\*\*

Testing

It is important to test out the system not only to check for errors but also to ensure that the code is optimised. The three testing methods that were looked at were:

* Black Box Testing
* Unit Testing
* e2e Testing

**Black Box Testing**

Black Box Testing, also known as Behavioural Testing, is a software testing technique. Developers use Black Box Testing when they want to test their system without looking into the internal code structure, implementation details and knowledge of internal paths of the system. The testing is based only on the specifications and the requirements. During the test, the main focus is the input and the output of the software system.[(36)](https://www.zotero.org/google-docs/?QD7XuS)

The most well-known types of Black Box Testing are:

**Functional Testing**: the testing of functional requirements of the system

**Non-Functional Testing**: tests non-functional requirements such as performance, scalability and usability

**Regression Testing:** Used when new code is implemented or updated to check whether the new code affected the current working code.

[(36)](https://www.zotero.org/google-docs/?yWEVBS)

**Unit Testing**

Unit Testing is a testing method where individual units or components of software are tested. It is used to confirm that the function works as it is supposed to. In any software, the smallest testable part is a unit. Unit Testing normally has one or very few inputs and outputs. The main goal is to separate each unit of the system to pinpoint, analyse and fix the defects of a project. [(37)](https://www.zotero.org/google-docs/?sE8Rqi)

Advantages include:

* Unit testing increases confidence in updating or maintaining the code. A good unit test code can catch any defects caused by the update.
* Code that are using unit tests are modular, therefore, those codes are more likely to be reusable.
* Unit tests allows debugging to be simpler as developers know that it is the latest change that caused the test to fail. [(37)](https://www.zotero.org/google-docs/?9RZiNi)

**e2e Testing**

End-to-end testing is a testing technique used to see if the flow of a system is performing as designed from start to finish. Developers implement End-to-end tests to identify system dependencies and to verify that the right information is passed between different system components and systems. [(38)](https://www.zotero.org/google-docs/?ztep7G)

End-to-end testing involves that the implemented components of a system function as expected. Functions such as communicating with the database, network, hardware and other applications are tested. [(38)](https://www.zotero.org/google-docs/?2ft4Dp)

Advantages of e2e testing includes:

* Reduces cost and time
* As the system is tested rigorously after every iteration, it reduces future risks
* In e2e testing the application is tested thoroughly, therefore there is no need to go back a test the functionalities again in the future.

[(39)](https://www.zotero.org/google-docs/?7zsOpx)

Testing was an essential part to the application. 3 users were approached, which will be referred to as P1, P2, P3. For one of the users, not only were they not familiar with the application but also unware that a test was being conducted. While the remaining two were ask to perform a set of action in order for the test to be unbiased. Each user were from a different age class, ranging from 18 to 62. The test was recorded and feedback was taken into consideration.

* P1: P1 first looked for an item that they were interested in, then check if the size of the item they wanted was in stock. Upon an out of stock item, P1 copied and pasted the URL of that item into the application without complications. P1 then selected their size and continued to enter their details.

P1 also tried the recommender system by uploading an image and waited for the result.

P1 had no trouble using the application but would’ve liked it if they were able to register and login to avoid entering their details every time.

* P2: P2 first registered for an account. They then added a list of links they prepared prior to the test. P2 reported that 2 out of the 5 links entered came back in stock, resulting in P2 receiving an email. P2 was able to purchase both items.

P2 also tested out the recommender system by allowing the system to place a recommendation based on the 5 links that was entered. P2 felt that the recommendations could’ve