Table of Contents

Contents

[Analysis 1](#_Toc169560276)

[Summary of Project 1](#_Toc169560277)

[Proposed user 2](#_Toc169560278)

[Existing System 2](#_Toc169560279)

[Description of Existing System 2](#_Toc169560280)

[DFD of Existing System 3](#_Toc169560281)

[ER Model of existing system 4](#_Toc169560282)

[Walkthrough of existing system 5](#_Toc169560283)

[Evaluation of Current system 6](#_Toc169560284)

[Advantages of Current System 7](#_Toc169560285)

[Disadvantages of current system 8](#_Toc169560286)

[Formal Analysis 9](#_Toc169560287)

[Existing documentation 9](#_Toc169560288)

[Interview transcript 10](#_Toc169560289)

[Interview summary 13](#_Toc169560290)

[Alternative systems 14](#_Toc169560291)

[Proposed System 15](#_Toc169560292)

[Description of proposed System 15](#_Toc169560293)

[DFD of proposed System 16](#_Toc169560294)

[ER Model of proposed system 17](#_Toc169560295)

[Walkthrough of proposed system 18](#_Toc169560296)

[Advantages of Proposed System 19](#_Toc169560297)

[Objectives 20](#_Toc169560298)

[Project Management 20](#_Toc169560299)

[Documented Design 22](#_Toc169560300)

[System Design 22](#_Toc169560301)

[DFD of proposed system 22](#_Toc169560302)

[Hierarchy charts to show structure of system 22](#_Toc169560303)

[Data structures 22](#_Toc169560304)

[Detailed E-R diagrams & detailed data analysis (including normalisation to 3NF) 22](#_Toc169560305)

[Object model and class definitions 22](#_Toc169560306)

[UML 22](#_Toc169560307)

[User defined structures 23](#_Toc169560308)

[Interface 23](#_Toc169560309)

[HCI (Form and report designs) 23](#_Toc169560310)

[Objective X 23](#_Toc169560311)

[Research 23](#_Toc169560312)

[Objective Design 23](#_Toc169560313)

[UML / ER diagram / Object Diagrams 23](#_Toc169560314)

[Pseudo code 23](#_Toc169560315)

[Prototyping & Development 23](#_Toc169560316)

[Code listing to meet objective 23](#_Toc169560317)

[Unit Test plan for Algorithm 23](#_Toc169560318)

[Evidence of testing 26](#_Toc169560319)

[Technical Solution 28](#_Toc169560320)

[System Testing 29](#_Toc169560321)

[Testing objective 1 29](#_Toc169560322)

[Paste the objectives in here 29](#_Toc169560323)

[Test Plan for objective 1 29](#_Toc169560324)

[Explain how you will test this – including normal, boundary and erroneous test data. How do you know if the test is successful 29](#_Toc169560325)

[Evidence of achieving Objective 1 29](#_Toc169560326)

[Evaluation 30](#_Toc169560327)

[Evaluation against Objectives 30](#_Toc169560328)

[User Feedback 30](#_Toc169560329)

[Analysis of user feedback 30](#_Toc169560330)

[During Implementation 30](#_Toc169560331)

[From User Testing 30](#_Toc169560332)

[Improvements 30](#_Toc169560333)

[Bibliography 31](#_Toc169560334)

[Appendix A 31](#_Toc169560335)

[Appendix B 31](#_Toc169560336)

# Analysis

## Summary of Project

With my project (a digital wardrobe), I intend to achieve to solve two main problems. I have decided to choose these goals as I believe that I can design an efficient system in which the user will benefit. The two problems that my project aims to solve consists of: the ability to find clothes from images and the cheapest prices, and an online organisation of a wardrobe.

Many people may find that they see an image of clothing that they like but may not know the brand/name. This leads them to the problem of having to hunt on the internet to attempt to find a product which matches the desired product. Furthermore, the user will also have to search for the cheapest option, adding more issues to the problem. My project aims to automate this system by enabling the user to enter images of clothes, then it will scrape the internet to find the cheapest product that matches the image.

Secondly, my project aims to increase the ease of organising the user's wardrobe. It aims to do this by instead of physically organising it, creating an online version where the user can submit photos of their clothes and create a library where they can be sectioned off into the different categories of clothing. This means that the user can decide what to wear without having to search through other clothes and creating a mess. Furthermore, this can then be used to aid the user further by offering services including creating outfits, telling the user how much they wear certain items etc.

## Proposed user

For my project, my proposed user is the public, as I intend to design the program to solve day to day problems. To develop my software appropriately for this demographic, I will create a system that hides all the complexity by using a simple interface with functions clearly stated. This will enable to software to be simple and easy to use, which is a necessity as the public will have a large variety of technological skills.

While carrying out my analysis I will be receiving feedback from a group of people interested in the fashion industry, one of which will be including Tom Yu. He has an average level of technological skills, however, would still need the simple interface.

## Existing System

### Description of Existing System

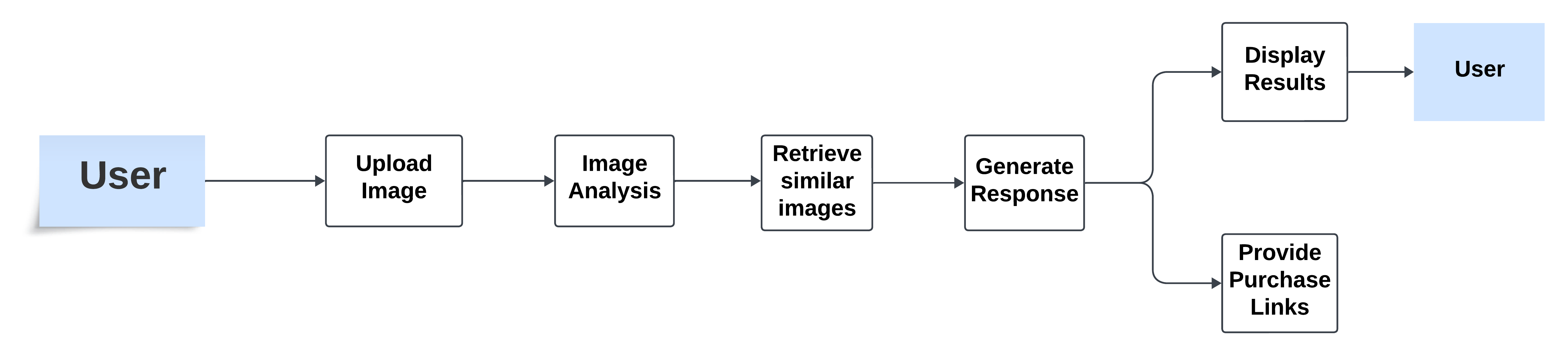
What is the existing system? Currently, users will have to input their images into software like google search by image, or Pinterest. These systems allow the user to input the image and then google search by image will display images like the inputted image, whereas Pinterest will display clothes of a similar style. Sometimes, attached to the images will be a link to where to buy the clothes, however it is not guaranteed that your search will result in a location to purchase.

Who uses this system? In the existing system, users are usually younger generations, aged between 13-30. Furthermore, it is more popular to be used by the fashion industry and social media influencers which specialise in fashion.

How do they use the system? This system can be accessed from any device, and only requires for an image to be taken and uploaded.

What is the system used for? This system is used to enable the user to be able to more easily find clothes that they like. For example, Pinterest’s function is to generate responses to images saved t the user and it will keep generating responses with the same style. In addition, google search by image function is to return images which look like the inputted image.

### DFD of Existing System

Level 0 DFD

Level 1 DFD

A screen shot of a computer

Description automatically generated

### ER Model of existing system

A screenshot of a computer

Description automatically generated

### Walkthrough of existing system

1. The user imports a picture of the piece of clothing that they desire to find into the current system software. This can either be by taking a photo of the piece of clothing or downloading an image from the internet. There are no forms at this stage and no inputs.
2. The image is then uploaded to the chosen platform (such as Pinterest or google search by image). There is an upload form allowing the user to browse and select the image file to input it.
3. The platform will then process the uploaded image and then display any results. Google search by image displays images which appear to look similar, and Pinterest will display clothing items and styles that are like the uploaded image. There are no forms or inputs at this stage. Search results with images and links are outputted.
4. The user will scroll through the resulting outputs to find the desired product and may click on images to view more details or visit linked web pages for more information. User can input filters on the platform so they can refine their search results to more easily find what they are looking for.
5. Once the user has found their desired product, they can look to see if the result has an attached purchase link, however, not all results have purchase links and so users will need to identify ones that do. No inputs, outputs or forms at this stage.
6. At this stage the user can click on the link so that they are redirected to the website in which they can purchase the product. They might have to visit several links to compare prices and options across different stores. At this point there might be website forms that the user must complete for purchase details, like size, colour, quality etc. Product website pages are outputted.
7. The user can then complete a purchase on their chosen sight by adding the product to their cart, proceed to checkout and then provide payment and shipping information. This means they will have to fill out a checkout form and an order confirmation page and email confirmation of the purchase will be outputted.

EXAMPLE FORMS

**Checkout Form**

Shipping Information

Name:

Address:

City:

County:

Postcode:

Country:

Payment Information:

Card Number:

Expiry Date:

CVV:

Image Upload Form:

Upload an Image

[Choose File]

[Upload]

**Purchase Checkout Form:**

### Evaluation of Current system

The primary purpose of the existing system (including platforms like Pinterest and Google search by image) is to aid users to find clothing items like ones in images which are uploaded. This includes clothing which is a similar style, or the exact clothes which are uploaded. There is an additional goal of sometimes providing links to purchase these items.

Evaluation Criteria:

1. User Experience
2. Accuracy of results
3. Availability of purchase links
4. Ease of use
5. Target Audience satisfaction

User Experience:

The user experience is generally positive, due to the simplicity of uploading images and viewing results being a significant advantage. However, there could be improvements in filtering options to streamline the search process.

Accuracy of Results:

The accuracy of the results is good but not perfect. While the system usually performs well, it can be improved, such as refining the algorithm to reduce irrelevant results.

Availability of purchase links:

The system mostly achieves its purpose of facilitating purchases. Some results will include purchase links; however, this inconsistency could cause frustration towards the program for users looking to buy specific items.

Ease of Use:

The system is mostly easy to use, with the main tasks of uploading images and browsing results being simple and intuitive. However, a multi-step process to complete a purchase can be streamlined.

Target Audience Satisfaction:

The target audience is likely satisfied with the system, especially given its integration with popular platforms, however increased integration with social media and direct purchase options could enhance satisfaction further.

Does the system achieve its purpose overall?

I believe the current system partially achieves its overall purpose as it succeeds in providing users with visually similar images and stylistically similar clothing items. It is user friendly and caters well to its target audience. However, the inconsistency in providing purchase links and the occasional irrelevant results indicate that there is room for improvement.

### Advantages of Current System

User Experience:

* The system is easily accessible via popular platforms.
* Users can quickly upload images and receive results without needing to sign up or provide personal information.
* The platforms are well-designed, providing an intuitive and visually appealing interface.

Accuracy of Results:

* Google search by image provides highly accurate visually similar images due to its advanced image recognition algorithms.
* Pinterest algorithm effectively identifies similar styles, catering well to fashion-focused users.

Availability of Purchase Links:

* Some search results include direct links to purchase the items, which is convenient for users.

Ease of Use:

* The process of taking a picture, uploading it, and receiving results is straightforward and quick.
* Both Google and Pinterest have user-friendly interfaces that require minimal effort to navigate.

Target Audience Satisfaction:

* The system caters well to younger generation and fashion enthusiast, offering relevant and visually appealing content.
* Popular among socially media influencers and fashion industry professionals who use it to find and share styles.

### Disadvantages of current system

User Experience:

* Users may sometimes find the navigation and filtering options overwhelming as there are so many options. This can be improved by enhancing filtering options to allow users to more easily refine their searches based on specific simple criteria. This can also be added to preferences for each user to filters are automatically applied.

Accuracy of Results:

* Results may occasionally include unrelated or irrelevant images. This could be improved by improving image recognition and style matching algorithms to increase the accuracy of results and reduce the number of irrelevant images.
* The accuracy of finding exact clothing matches can be inconsistent. This can be improved by implementing a feedback function where users can mark results as relevant or irrelevant, allowing the system to learn and improve over time.

Availability of Purchase Links:

* It is not guaranteed that every search will provide purchase links. This can be improved by allowing the user to filter between items having links and not, so that they can filter out any results which do not have a link.
* Users may have to search through several results to find a purchasable item. This can be improved by clearly highlighting which results have a purchasable links to save users time in finding items they can buy.

Ease of Use:

* Users may need to switch between platforms to complete a purchase, adding extra steps. This can be improved by integrating the purchasing process within the platform to minimise the need for switching between different websites.

Target Audience Satisfaction:

* Younger users might expect more seamless integration with social media and e-commerce platforms. This can be improved by allowing users to integrate social media by sharing recent purchases or including direct links to share items on social media platforms.

## Formal Analysis

### Existing documentation

You should include any existing documentation the use of counter uses, such as application forms or screenshots of data entry screens.

Google Search by Image:

A screenshot of a computer

Description automatically generated

Pinterest

A screenshot of a chat

Description automatically generated

### Interview transcript

Create a set of interview questions and include the transcript of interviewing ideally 2 proposed users. You may wish to put this in an appendix and only refer to it when summarising the key points from the interviews.

Personal Background:

1. Give me a brief description of you and your background?
2. How often would you say you use online platforms to search for clothing?

Usage of The Current System

Platform Choice:

1. Which platform do you primarily use for searching clothing items? Why?
2. How frequently do you use these platforms?

Image Upload Process:

1. Can you walk me through your process of uploading an image to search for clothing items?
2. Do you find the image upload process straightforward? If not, what issues do you encounter?

Search Results

1. How accurate do you find the search results in terms of matching the style of appearance of the clothing in your uploaded image?
2. How often do you find results that are completely irrelevant to your search?

Filtering and Navigation

1. How do you typically navigate and filter through the search results?
2. Do you find the filtering options sufficient for narrowing down your search? If not, what additional filtering options would you like to see?

Purchase Links

1. How often do you find purchase links in the search results?
2. When purchase links are available, how clear and useful do you find them?
3. Have you ever encountered issues with broken or misleading purchase links?

Ease of Use

1. Overall, how easy do you find it to use the platform for searching and purchasing clothing items?
2. Are there any specific features or steps you find particularly difficult or time-consuming?

User Experience

User interface

1. How do you find the overall user interface of the platform you use most frequently?
2. Are there any aspects of the interface that you find confusing or difficult to use?

Comparison of Platforms

1. Have you used multiple platforms for this purpose? If so, how do they compare in terms of user experience and effectiveness.

Satisfaction and Improvements

1. How satisfied are you with the current system’s ability to help you find and purchase clothing items?
2. What features or improvements would make the system more useful and enjoyable for you?

Device Usage

1. What device do you primarily use to access these platforms?
2. Have you encountered any device specific issues?

Overall Feedback

1. What do you like most and least about the current system?
2. What new features or functionalities would you want added to the platform?
3. Is there any extra information you would like to share about your experience?

Transcripts:

Interviewee: Tom Yu

1. My name is Tom, and I am a 17-year-old student from Kent. I am very interested in clothing and find myself researching it a lot.
2. I find myself using various apps on my phone and laptop to search for clothes several times a day, including retails apps and social media.
3. I primarily use TikTok to find clothing inspiration or ideas and apps like StockX, vinted, END, supreme and size? Are where I typically purchase my clothes from.
4. I check the apps several times a day and make purchases multiple times each month.
5. I don’t usually upload photos of clothing I want to purchase as there is not an effective and useful tool/application for it.
6. The image upload software I have used before do not give me reliable results and never hone in on specifics.
7. Search results tend to be very vague especially when I need branded clothes, most search engines struggle to pinpoint what I am looking for.
8. Very often
9. I usually filter by price, size and colour.
10. Depending on the app I find these filters sometimes very useful and sometimes then never give me useful results especially on apps with huge stock like END.
11. Very often
12. Purchase links are typically very clear but not all websites are enjoyable to navigate.
13. Occasionally fake websites appear at the top of Google
14. Usually, apps or websites themselves also cannot locate the clothing I’m looking for – instead just giving me similar results.
15. Filtering through useless similar results can be very time consuming
16. I find the user interface good but missing lots of filters and comparisons that I could utilise.
17. Not particularly
18. I typically use Pinterest or TikTok – TikTok has no features to purchase clothes
19. Fairly satisfied – 6/10
20. Better filtering systems and if they made it easier to locate sizes and brands
21. Laptop
22. No
23. Most accessibility, least – lack of features
24. Ones I mentioned on question 20
25. No

Interviewee: Teddy Kennard

1. I am a 17-year-old student from Maidstone who studies at MGS sixth form
2. Every day to an extent
3. Depop, Asos, Flannels, binged
4. Couple times a week
5. Find photos of clothes I like, and then upload to google to find my results
6. Yes, it’s easy
7. Somewhat accurate, doesn’t always work
8. Not too often
9. Sizes, price, brand and colour
10. I think they are helpful, maybe a condition of the material option
11. Often
12. They’re useful if I’m interested in the product
13. Sometimes
14. Reasonably easy
15. Not really
16. They’re modern and fast
17. No
18. Second hand resellers are easier to use as you can contact the seller to find out more details about the product
19. Reasonably satisfied
20. For it to be more automated
21. Computer
22. No
23. I like the efficiency but not the fact that I don’t always get a result
24. I would like to be able to share the clothes I’m buying
25. No

### Interview summary

From my interviews, I can summarise that users enjoy a simplistic system in which they are able to filter through results easily and have links provided for them. Neither of my interviewees use a typical system in which I have described such as Pinterest or Google search by image. This is because they find they do not always get their desired results.

Users frequently engage with these platforms’ multiple times a day or week, which is indicating they are well-integrated into their daily routines. They also find image uploading to be a straightforward process which means that a user-friendly interface is popular.

They will commonly use filters to filter for prices, sizes, brands, and colours. When purchasing links are available, users find this very useful especially when they are interested in buying the products.

Both interviewees provided that they use these platforms mainly on their computers as they prefer the bigger screen, in addition, one of my interviewees mentioned that there is no effective tool for uploading photos to search for specific clothing items, suggesting a gap in the functionality offered by current platforms. Both interviewees admitted that image analysis results are often incorrect or not specific enough, especially when it comes to branded clothing, which means they are not getting the results that they want.

Advantages:

* The Existing systems give inspiration to the user and sometimes provide purchase links. It is seamlessly incorporated into users’ daily routines.
* Users find features like purchasing links and filtering to be very useful.

Disadvantages:

* Accuracy and relevancy of image-based search results are not always reliable.
* Users may encounter difficulties with incorrect links, poor navigation, and efficient filtering
* To increase overall satisfaction, there is a need for more advanced features, improved automation, and improved user interfaces.

### Alternative systems

Commercial Packages for Image-Based Clothing Search

Google Lens

Good Features:

* Accurate image recognition.
* Wide range of results.
* Integration with google services.
* User-friendly interface.

Non-suitable features:

* Limited E-commerce integration (does not always prioritise ecommerce sites).
* Results can sometimes include unrelated items.

Pinterest Lens

Good Features:

* Style matching
* Inspirational content
* Shopping integration (has partnerships with retailers to make it easier to find purchasable items)

Non-suitable features:

* Overwhelming interface
* Variable result quality (quality of results may vary)

ASOS style match

Good Features:

* Fashion focused
* High quality matches
* Personalised recommendations

Non-Suitable features:

* Limited to ASOS inventory
* Regional availability

Justification of advantages and disadvantages

* Platforms like Google lens and Amazon StyleSnap use large databases and integrations to provide a vast range of results and to simplify the purchasing process.
* Pinterest lens and ASOS style match are tailored specifically for fashion, in which they offer style inspirations and personalised recommendation.

## Proposed System

### Description of proposed System

My Digital Wardrobe project seeks to address two primary issues that users may encounter. This involves creating an online wardrobe and locating clothing goods from photos and the cheapest cost. I plan to use web scraping and advanced image recognition technology to accomplish this. My system aims to give consumers who want to identify, buy and maintain their wardrobe with a smooth and effective experience.

My system will allow the user to submit an image, analyse it using an API, and then utilise Google to search for the product to help the user select clothing from images and determine the best pricing. Then, it will show links to the online store, and I plan on the program highlighting the cheapest option. The user will also receive URLs via emails.

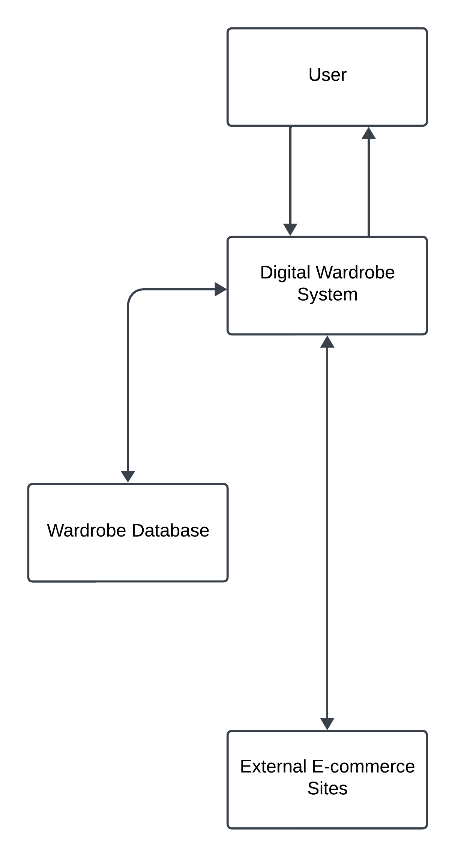
Additionally, my project will let users manage their online wardrobe by allowing them to construct a library of their clothing. The user won’t need to go into their wardrobe to make outfits and to view their clothes. This will enable the programme to offer information on how frequently clothes are worn and other analytics.

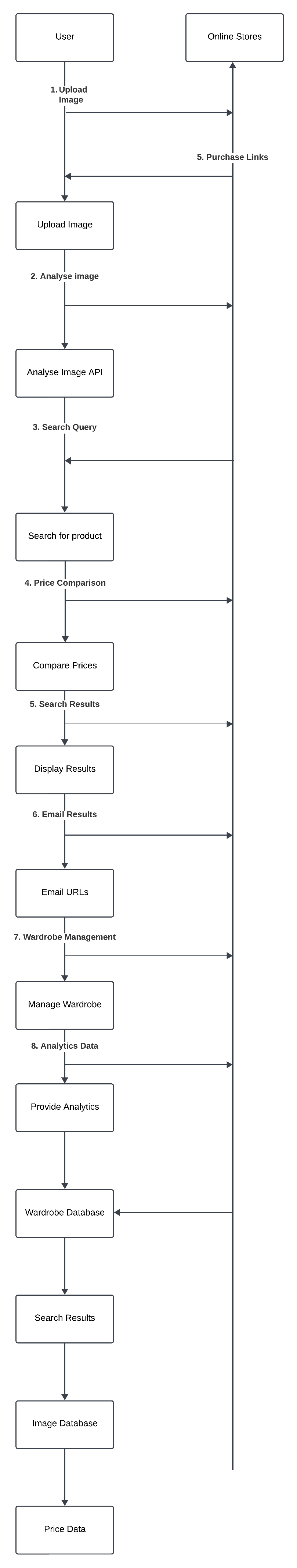
In contrast to the other current system, my project will compare pricing to identify the best offer. It differs from previous systems in that it enables users to digitally manage their wardrobe. To enhance the user experience with my system, I also intend to fix the disadvantages of existing systems.

Disadvantages of my system could be that users will have to put in initial work because they will have to take pictures of the items in their current wardrobe to construct the digital library. Moreover, the precision of the online scraping algorithm determines how effective the price comparison is.

### DFD of proposed System

Level 0: Level 1:





### ER Model of proposed system

A computer screen shot of a computer flow chart

Description automatically generated

### Walkthrough of proposed system

It is in this section that you will model your proposed system by walking through how you anticipate the user using your system. This should include mock-ups of the user interface and mock-ups of the outputs produced.

1. The user will import a photo of the piece of clothing that they want the system to find. They can either use a photo that they have taken or downloading an image from the internet. The user will be able to open an input form for the image to upload it to the program, and then it will display the selected image on the screen for the user to see.

Upload Image

[ Choose File ]

[ Upload ]

1. The program will then process the image the user uploaded using the Cloud Vision API, creating labels based on the image, and then using Google’s Custom Search API it will create a google search using the labels. The program will then output links for the user.
2. The user can then browse through the search results to find their desired product. They can click on images to view more details or visit web pages and filters can be applied to refine search results.

Search Results

[ Image ] Product 1 [Link]

[ Image ] Product 2 [Link]

[ Image ] Product 3 [Link]

Filters: [ All ] [ Colour ] [ Size ] [ Clothing ]

1. The User can purchase the clothing through the program and the system will output their order details.
2. Images that are inputted into the system by the user will be saved to a library in which they can organise their clothes into sections (e.g Tops, trousers, etc.) and will contain tools to allow them to create outfits and provides analytics on the clothes they use.

### Advantages of Proposed System

* Automation and efficiency
  + Automates the process of finding clothing from images and identifying the cheapest prices, saving users time and effort.
* Comprehensive Search
  + Utilises web scraping to cover a wide range of online stores, ensuring users get the best deals
* User-Friendly Organisation
  + Provides a digital wardrobe that is easy to navigate, reducing the hassle of physical organisation
* Enhanced decision making
  + Offers features such as outfit suggestions and usage tracking, helping users make informed decisions about their wardrobe

## Objectives

1. Find clothes from images and identify the cheapest prices
   * This objective involves image recognition combined with price comparison, which are technically challenging and require integration if multiple processing tasks.
   * Processing Requirements
     1. Image Recognition: implementation of reliable image analysis algorithms (potentially using machine learning) to identify clothes from images.
     2. Price Comparison: Integrating with external APIs or implementing web scraping to fetch prices from multiple sources
     3. Efficiency: Ensuring quick response times for image processing and price comparison.
     4. Accuracy: Algorithms must accurately identify items and provide correct price comparison.
2. Allow user to input images
   * User input functionality is essential for a practice application involving image recognition and analysis.
   * Processing Requirements
     1. Interface Design: Developing an intuitive interface for users to upload images.
     2. Data Handling: Efficiently managing and processing uploaded images.
     3. Security: Implementing measures to secure user-uploaded content.
     4. Usability: Ensuring seamless integration with the image recognition system.
3. Compare prices
   * Suitability: Price comparison is a practical feature for a shopping-related application.
   * Processing Requirements for Higher Grades:
     1. Real-Time Comparison: Implementing real-time or near real-time price fetching and comparison.
     2. Integration: Handling data from multiple sources accurately and efficiently.
     3. Scalability: Ensuring the system can handle a large volume of price comparisons.

4. Online Organisation of a wardrobe

* Suitability: Organising wardrobe items digitally is practical and user focused.
* Processing Requirements for Higher Grades:
  1. Database Management: Designing and optimising a database for storing and retrieving wardrobe items.
  2. User Interface: Developing a responsive and intuitive interface for managing wardrobe items.
  3. Personalisation: Allowing users to categorise and tag items according to personal preferences.

5. Create a library of clothes

* Suitability: Creating a digital library complements wardrobe organisation.
* Processing Requirements for Higher Grades:
  1. Metadata Handling: Efficiently storing and managing metadata associated with each clothing item.
  2. Search and Retrieval: Implementing fast and accurate search functionalities within the library.
  3. Visualisation: Providing visual representations of clothing items for user interaction.

6. Data analytics based on user’s choices

* Suitability: Analysing user data can provide insights for personalisation and recommendations.
* Processing Requirements for Higher Grades:
  1. Analytics Algorithms: Developing algorithms to derive meaningful insights from user interactions and choices.
  2. Visualisation: Presenting analytics results in a clear and understandable manner.
  3. Privacy: Ensuring compliance with data protection regulations and maintaining user privacy.

7. Aid user to make outfits

* Suitability: Offering outfit suggestions enhances user experience and engagement.
* Processing Requirements for Higher Grades:
  1. Algorithm Development: Creating algorithms to suggest outfits based on user preferences and available wardrobe items.
  2. User Feedback: Implementing mechanisms for user feedback to improve outfit recommendations.
  3. Personalisation: Customising outfit suggestions according to individual style preferences.

8. Reliable image analysis to produce consistent results

* Suitability: Consistency in image analysis is crucial for accurate results.
* Processing Requirements for Higher Grades:
  1. Algorithm Robustness: Developing robust image analysis algorithms that perform consistently across different image qualities and types.
  2. Error Handling: Implementing mechanisms to handle errors and improve algorithm accuracy over time.
  3. Performance: Optimising processing speed without compromising accuracy.

9. Create more automation for buying clothes

* Suitability: Automating parts of the shopping process enhances user convenience.
* Processing Requirements for Higher Grades:
  1. Integration with E-commerce Platforms: Integrating with APIs of online stores for seamless purchasing.
  2. Transaction Security: Ensuring secure transactions and handling of sensitive user data.
  3. User Experience: Optimising the purchasing process for ease of use and reliability.

Emphasised Objectives:

* Advanced Algorithms: Especially for image analysis and data analytics.
* Efficiency: Quick response times and scalable processing capabilities.
* Integration and Security: Secure and seamless integration with external services.
* User Experience: Intuitive interfaces and personalised features.

## Project Management

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Start | End | Duration (days) |
| Allow user to input images | 03/07/2024 | 04/07/2024 | 1 |
| Reliable Image Analysis to produce consistent results | 03/07/2024 | 10/07/2024 | 7 |
| Find clothes from images and identify the cheapest prices | 10/07/2024 | 17/07/2024 | 7 |
| Compare prices | 17/07/2024 | 03/08/2024 | 17 |
| Online organisation of a wardrobe | 16/08/2024 | 25/08/2024 | 9 |
| Create a library of clothes | 04/08/2024 | 16/08/2024 | 12 |
| Data analytics based on user's choices | 17/08/2024 | 24/08/2024 | 7 |
| Aid users to make outfits | 25/08/2024 | 05/09/2024 | 11 |
| Create more automation for buying clothes | 06/09/2024 | 25/09/2024 | 19 |

A diagram of a project

Description automatically generated with medium confidence

FIX THE GAANT GRAPH – switch online organisation of a wardrobe and create a library of clothes

# Documented Design

Include a description of your proposed system. This could be copied from the analysis and is used to set the scene for the design.

In this section, you're expected to demonstrate that you have the skills to:

* Use formal design methodologies, including pseudo code.
* Research solutions and develop your findings into a prototype and a system that meets your users’ requirements or objectives
* If multiple algorithms are available, you're expected to be able to choose and justify your choice, for example using big O notation time complexity.
* Unit test your code, identifying errors and fixing them.

## System Design

### DFD of proposed system

A screen shot of a black screen

Description automatically generated

### Hierarchy charts to show structure of system

A computer screen shot of a system

Description automatically generated

## Data structures

You should identify the key data structures and data that you require. For simulations this include the data required for the simulation to operate. 4 non data processing systems, you're encouraged to identify local and global variables and any constants you'll be using.

Image Analysis

Data Required: Image File

Local Variable: analysisResults

Data Structure: Dictionary to store labels/features made from image analysis

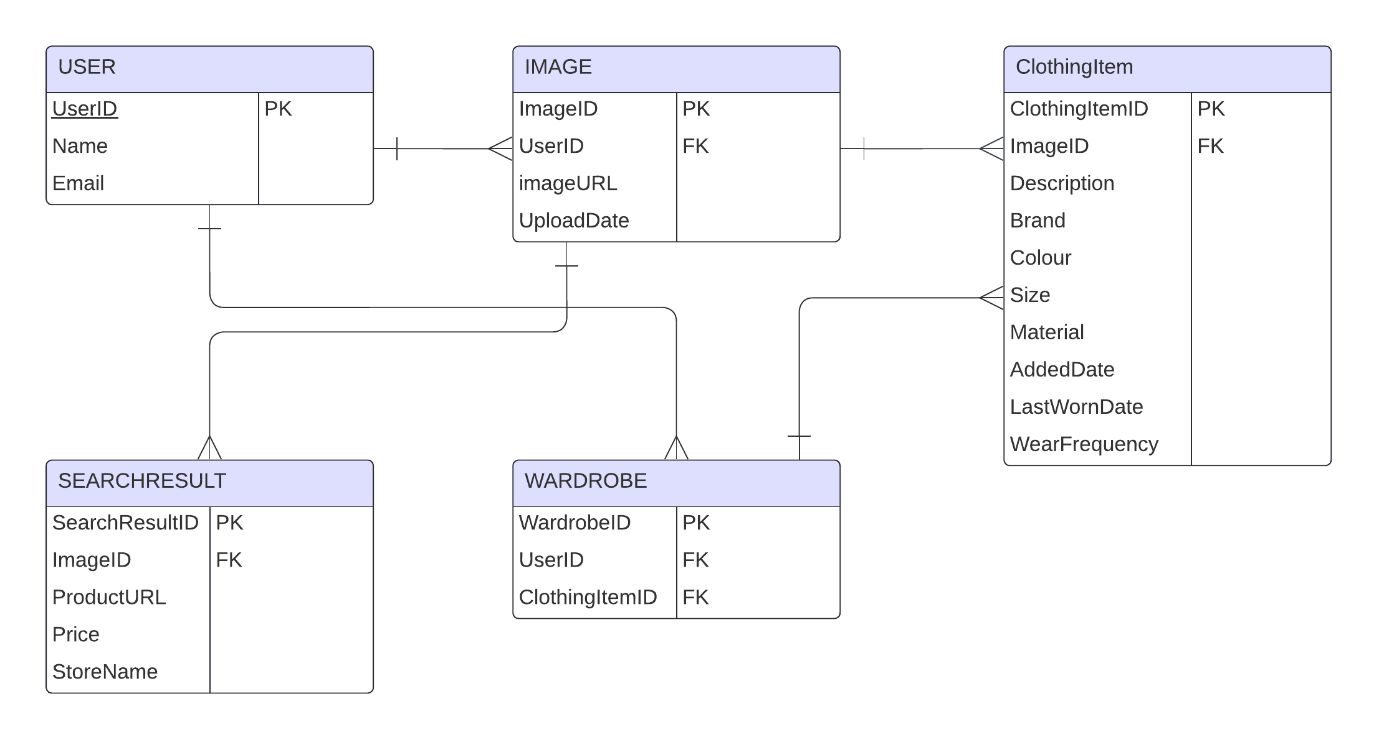
Images

* + Labels (e.g colour, material etc)
  + Metadata (upload date etc)
  + Local variables: imageFile,

Products

* + ProductID
  + Name
  + URL

### Detailed E-R diagrams & detailed data analysis (including normalisation to 3NF)



USER (UserID, Name, Email)

IMAGE (ImageID, UserID, ImageURL, UploadDate)

CLOTHINGITEM (ClothingItemID, ImageID, Description, Brand, Colour, Size, Material, LastWornDate, WearFrequency)

SEARCHRESULT (SearchResultID, ImageID, ProductURL, Price, StoreName)

WARDROBE (WardrobeID, UserID, ClothingItemID)

USER and IMAGE have a one-to-many relationship as a user can upload multiple images, however, an image can be uploaded by one user. IMAGE and CLOTHING have a one-to-many relationship as one image can be associated with many clothing items, however, a clothing item can be associated with one image. USER and WARDROBE has a one-to-one relationship as one user can have one wardrobe and one wardrobe can have one user. CLOTHINGITEM and WARDROBE has a many-to-many relationship as one clothing item can be in multiple wardrobes, and one wardrobe can contain many clothing items. IMAGE and SEARCHRESULT has a one-to-many relationship as one image can have multiple search results, but a search result can only have one image.

### Object model and class definitions

### UML

Include a UML of your object-oriented programme design, including any inheritance and association.

### User defined structures

These could be diagrammatic representations of things like lists, graphs and trees

## Interface

### HCI (Form and report designs)

Include hand drawn plans for the user interface and use these to make mock-ups of the intended user interface. If required these could be prototypes for the user interface.

\*\*\*\*Repeat the following for each objective / key section of your project\*\*\*\*

## Allow Users to Input images

* Create a UI for a main menu
  + Decide a user-friendly layout
  + Consider aesthetics
* Create a button which allows images to be inputted
  + Requires an input dialogue
  + Only allow desired formats to be entered
* Display the image that is entered
* Error handling

### Research

I will have to research how to make a professional appearing UI, as I have only experience in making simple forms.

### Objective Design

Use Case Diagram

### UML / ER diagram / Object Diagrams

### Pseudo code

Image1 = image

IF Select Image Clicked

Create new OpenFileDialogue object

ImageFile = string

Set OpenFileDialogue filter = Image Files

Set OpenFileDialogue title = “Select An Image File”

Display OpenFileDialogue to user

If user selects a file

Set ImageFile to selected file’s path

Open imageAnalysis form

Display ImageFile

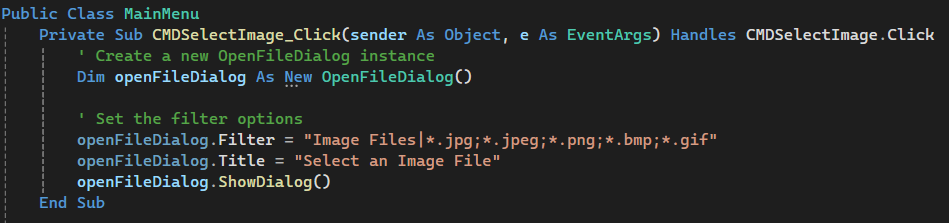
End IF

End IF

### Prototyping & Development

#### Creating a button to allow images to be inputted

The objective which I aimed to complete first was allowing users to enter an image into my program. I created a temporary button on my main menu to use to allow the user to click it. This will open the OpenFileDialogue component I added to the forms so that the user can choose an image file. I also added a filter so that the user can only select an image file.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

#### Creating a UI for the Main Menu

To develop the user interface of my software, I have chosen to use photoshop to create an image that I can use as a button because my client and I do not like the design of the current buttons provided in Visual Studio Forms. From research from my target market, I identified that my software would have more popular use on a mobile device, so I found what the optimal button sizes were for mobile. Instead of using I button, I added a PictureBox component and selected the image I created on photoshop. Then I added code which will occur when the picture box is clicked.

A grey oval with white text

Description automatically generated

When the button is hovered over, it will change to indicate to the user that they will be pressing that button.

A grey oval with white text

Description automatically generated

My clients and I did not like the UI of this main page, so I decided to update it to make it a more user-friendly UI. I used this video (M, 2021). I used a split panel so that I could create the effect of a menu bar being open and closed as I needed to be efficient with how I use the space as there is not a lot.

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

My client and I didn’t like how this menu bar functioned, so I incorporated a different method using a similar concept as before. I was not able to display different forms as pages inside of the split container, so I created to panels. One for the menu so it can be shown and hidden and one of the pages that I want to load. (crimson, 2015) this was used to display a form in a panel. I also created a new form called imageAnalysis with a picture box inside of it so that when an image is selected, it can be displayed inside of the picture box.

A white screen with a blue border

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

#### Error Handling

If no image was entered and the OpenFileDialogue was closed, the program would crash, so to stop this I used an IF statement.

A screen shot of a computer program

Description automatically generated

This allowed me to load forms as pages in the panel, however, when trying to switch between the two forms, I found that the program would not perform as wished and the buttons started to switch around showing each other’s contents. To fix this I need to add a variable which will keep track if a form is being displayed or not and what form is being displayed. This way if a form was being displayed, it can be removed before then loading the next one.

A screenshot of a computer program

Description automatically generated

After some testing I also realised that if you clicked the button of the page that you are already on, the program crashes. So, to fix this I will using an IF statement that will only run the code to change the menu, if the panel that is trying to be loaded is not visible.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

### Code listing to meet objective

## Create a Library of Clothes

### Research

I didn’t know how to display my images, so I looked through Visual studios toolbox and found a FlowLayoutPanel. The description of this matched the functionality that I needed, and so I decided to use it. For the code I looked through the options I was given and used my own logic to add the picture box to the FlowLayoutPanel.

To store an image in access I used some from stack overflow and adapted it to suit my program. (Mary, 2020)

Used: <https://flatuicolors.com/> to help with colour scheme

To create a library of clothes for my user, I aimed to store the images as a blob in a database.

### Objective Design

### UML / ER diagram / Object Diagram

A screenshot of a computer

Description automatically generated

### Pseudo code

### Prototyping & Development

Things to do:

* Allow users to enter images
* Display the images as organised as they are entered
* Prompt user to enter information about the image
* Create a database
* Save the image and the data entered to a database
* Have images in the database load as the form is opened

First I knew that I needed to allow the user to input images to be added to the library. So, I created a new form called wardrobe which can be accessed from the main menu using a button, then I used the same method as for analysing images to input the images and added a button with an openFiledialog.

A screenshot of a computer

Description automatically generated

I then looked through the properties of the controls I added and found I could enable autoScroll into the FlowLayoutPanel, which enables the scroll bar to appear when the images take up too much space.

The code for the OpenFileDialog was the same as used before so I was able to reuse that code, and then I added the code which allowed a pictureBox to be created for every image entry, and to then resize the picture box and add it to the FlowLayoutPanel.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

I found that the images are quite small and so I will resize the picture boxes so that they are larger and so now it looks like this:

After Updating the UI for my main menu, I re-designed this form as it will now be displayed inside a panel. I moved the button and flowlayoutpanel so it was positioned in the middle when it is loaded into the panel.

I then wanted to be able to save the images to a Database. I will make it so that if an image is entered without it being analysed, the user will be prompted to quickly fill out a small form describing the clothes so it can be saved to the database, however, if an image is analysed and added to the database then it will fill most of this automatically.

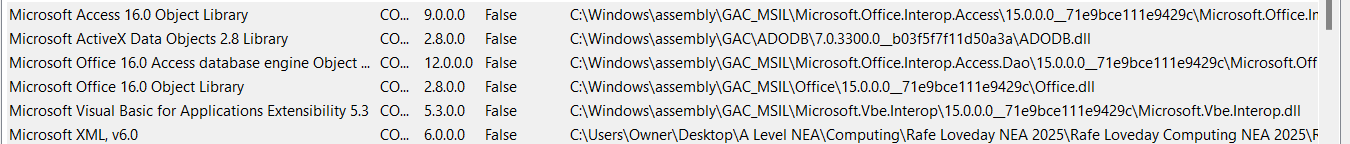
#### Creating a database

Using the ER diagram I made, I created the tables I needed inside of Access.

For the user table, I had to change the “Name” Field Name to “UserName” as “Name” is a reserved word inside of Access.

I created a database using access and then connected it to my program as a new data connection in visual studio. Using the notes in Onenote I can manipulate my database to add data to it and retrieve data.

I had to add these references into my program so that my program and the database can communicate.



### Code listing to meet objective

## Reliable Image Analysis to Product Consistent Results

### Research

* Using Forums on google and YouTube, I tried to discover a method to analyse an image inside of Visual basic, however I could not find anything.
* (Google, n.d.)
* As I already knew that Google provided many APIs, I used the Google API explorer page to find if there would be an API which is suitable to solve my task.
* I determined that the most suitable API would be Googles Cloud Vision API as it enables image labelling.
* This is a proof-of-concept database being used to simulate calling the API to find and retrieve codes. This is because APIs are blocked from school computers. I plan on creating my own database of clothes, and users will be able to enter clothes into the database and will be requested to enter information about the clothes.

### Objective Design

### UML / ER diagram / Object Diagrams

### Pseudo code

### Prototyping & Development

#### First Image Analysis method

The first method of image analysis I attempted was to implement a web browser into my program, which would allow the use of google search by image. I did this by adding a WebBrowser component into the project and adding the google URL into the properties or it.

A screenshot of a computer

Description automatically generated

Since I could not use Google Search by Image through a browser in my program, using my research on Google’s APIs, I decided to use ChatGPT to help me use the Google Cloud Vision API with my project as I was not able to find many resources online which were specific to Visual Basic. (OpenAI, 2024):

#### Connecting Google Cloud Vision API to my project

This Is the instructions that ChatGPT returned

A screenshot of a black screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

This is my Google Cloud Console project set up with the API connected to it that I needed to get the JSON key and so I can manage traffic through the API.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

As there is not a vast number of resources online for linking Google APIs to a Visual Basic project, I had to use code given from ChatGPT to create the link, however, I develop the code given for the label detection.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Here I added the code that was given so I had a base which would product labels. From this I can refine it to make the image analysis more reliable and more relevant to my project.A screenshot of a computer program

Description automatically generatedA computer screen shot of a program

Description automatically generated

After some testing, I realised that some of the labels that were created could be irrelevant. As shown a lot of these labels are not helpful in finding the clothes, so I decided that I would create a list of labels which I want to be detected and I used ChatGPT to create an IF statement which will only display labels from my list when they are created.A screen shot of a shirt

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

Testing having target labels

A screen shot of a computer program

Description automatically generated

I decided to come up with a collection of words which would have meaning for clothes to create an accurate analysis of the inputted photo. Also, copying the response variable, I only wrote Detect in to see what other functions I could use, and in doing so I was able to find the DetectLogos function. So, I copied the same code format that was used for label detection and just removed the IF statement used for the target labels. This code worked on the first attempt and after testing can accurately detect different logos.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

### Code listing to meet objective

## Find Clothes From Images and Identify the Cheapest Prices

### Research

https://developer.ebay.com/Devzone/shopping/docs/CallRef/index.html

### Objective Design

### UML / ER diagram / Object Diagrams

### Pseudo code

### Prototyping & Development

### Code listing to meet objective

## Compare Prices

### Research

### Objective Design

### UML / ER diagram / Object Diagrams

### Pseudo code

### Prototyping & Development

### Code listing to meet objective

## Online Organisation Of a Wardrobe

### Research

### Objective Design

### UML / ER diagram / Object Diagrams

### Pseudo code

### Prototyping & Development

### Code listing to meet objective

## Data Analytics Based on User’s Choices

### Research

### Objective Design

### UML / ER diagram / Object Diagrams

### Pseudo code

### Prototyping & Development

### Code listing to meet objective

## Aid Users to Make Outfits

### Research

### Objective Design

### UML / ER diagram / Object Diagrams

### Pseudo code

### Prototyping & Development

### Code listing to meet objective

## Create More Automation for Buying Clothes

### Research

### Objective Design

### UML / ER diagram / Object Diagrams

### Pseudo code

### Prototyping & Development

### Code listing to meet objective

### Unit Test plan for Algorithm

* Test plan templates are available.
* What part of the algorithm are you testing?
* Test data (you are encouraged to use a range of test data to test each stage including actual, boundary and erroneous).
* Expected results - you should include annotated screenshots of the results of your testing.
* Evidence that your tests have identified errors and give details of how you corrected these-you must show an error correction cycle.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No | What will be tested | Tests date | Expected results | Ink to evidence of actual results |
|  |  |  |  |  |
|  |  |  |  |  |

This this is an example of the detail required:

Test:

Validation routines

* Real/actual data
* Range/extreme data
* Erroneous data

Test database

* Adding records
* Amending records
* Deleting records
* Finding records
* Sorting records
* Complex algorithms
* Sorting
* Searching

Complex data structures (linked lists, trees, 2D arrays)

Calculations

What the system was designed to do!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Test  No | Description of what is being tested | Test data to be used | Expected results | Evidence |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Customer Age validation routine on FRMcustomer | | | |
| Test  No | Description of what is being tested | Test data to be used | Expected results | Evidence |
| 1.1 | Entering a valid customer age field | 26 | The data is accepted and entered in to the tblCustomer table | See page 100 |
| 1.2 | Entering a boundary value into the customer age field | 18 | The data is accepted and entered in to the tblCustomer table | See page 101 |
| 1.3 | Entering an erroneous value in the customer age field | 9999 | The data is rejected and an error message is displayed | See page 102 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | Adding customers to the customer linked list | | | |
| Test  No | Description of what is being tested | Test data to be used | Expected results | Evidence |
| 2.1 | Add a new customer, by selecting the option from the main menu | Enter a new customer:  Forename = Fred  Surname = Bloggs  Age = 25 | Record added to the end of the linked list  End-of\_list variable should be increased by 1 |  |
| 2.2 | Amend the record by  2.2a searching for the customer  2.2b changing their details | 2.2a – select the update button and enter “blogs” in the search field  2.2b change the age from 25 to 26. | 2.2a. Customer found and displayed on screen  2.2b Date changed and saved to the linked list |  |
| 2.3 | Amend the record by searching for the customer that does not exist | select the update button and enter “XXX” in the search field | Customer not found message displayed |  |
| 2.4 | Delete a customer | select the update button and enter “blogs” in the search field | Confirmation window displayed  On NO, nothing happens |  |
| 2.5 | Delete a customer | select the update button and enter “blogs” in the search field | Confirmation window displayed  On YES, node removed  End-of\_list variable should be decreased by 1 |  |
|  |  |  |  |  |

### Evidence of testing

\*\*\* END OF OBJECTIVE DEVELOPMENT

# Technical Solution

In section identify the techniques and skills you have used, making reference to table A and table B in the marking criteria.

# System Testing

The system testing should test your system against the objectives in the analysis. Make sure you include a test that shows that the objective has been met. It needs to be clear that your system meets the objective stated. The unit testing identifies the system works.

A good approach is to come up with several scenarios that demonstrates your user using the system and this could be represented as a walkthrough with lots of screenshots and annotation explaining how your system meets the objectives.

Suggestion section titles (repeated for all objectives)

## Testing objective 1

### Paste the objectives in here

### Test Plan for objective 1

### Explain how you will test this – including normal, boundary and erroneous test data. How do you know if the test is successful

### Evidence of achieving Objective 1

Include annotate screenshots of your solution meeting the objective.

# Evaluation

## Evaluation against Objectives

A reasoned and detailed explanation of how objectives have been met for acomplex problem with SMART objectives

### User Feedback

Clear evidence of genuine feedback authenticated by the assessor

## Analysis of user feedback

### During Implementation

Thorough analysis of user feedback and/or evidence of continual interaction with the user during the development of the system

### From User Testing

### Improvements

Full and realistic suggestions as to how to improvements and extensions are related to user feedback

# Bibliography

crimson. (2015, June 18). *Stackoverflow*. Retrieved from Stackoverflow: https://stackoverflow.com/questions/30904538/load-a-form-into-a-panel

Google. (n.d.). *Google API Explorer*. Retrieved from Google: https://developers.google.com/apis-explorer

M, K. (2021, June 4). *Youtube*. Retrieved from Youtube: https://www.youtube.com/watch?v=pscDVLiYYs0

Mary. (2020, May 31). *StackOverflow*. Retrieved from https://stackoverflow.com/questions/33763773/save-insert-image-in-ms-access-database

Microsoft. (n.d.). *Microsoft*. Retrieved from Microsoft: https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.openfiledialog?view=windowsdesktop-8.0

OpenAI. (2024, July 11). *ChatGPT*. Retrieved from ChatGPT: https://chatgpt.com/c/116dcae2-ca94-425e-85ba-a2dff679eda8

# Appendix A

Include a full program listing if not included in the documented design.

# Appendix B

Any additional information, such as

* full transcripts of any interviews,
* evidence of existing documentation used or
* any other relevant information.