**COMSATS University Islamabad,   
Abbottabad Campus**

**Project Proposal   
(SCOPE DOCUMENT)**

**for**

**<Saving Bot>**  
Version 1.0

***By***

**BASIT IQBAL CIIT/FA21-BSE-050/ATD**

**FATIMA AFTAB CIIT/FA21-BSE-080/ATD**

**WALEED RASHID CIIT/FA21-BSE-162/ATD**

***Supervisor*Syed Shahab Zarin**

***Bachelor of Science in Computer Science (2021-2024)***

**SCOPE DOCUMENT REVSION HISTORY**

|  |  |  |
| --- | --- | --- |
| **No.** | **Comment** | **Action** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Supervisor Signature**

**Date:**

Table of Contents

[**1.** **Introduction** 8](#_Toc170682873)

[**2.** **Problem Statement** 9](#_Toc170682874)

[**3.** **Problem Solution for Proposed System** 9](#_Toc170682875)

[**3.1.** **Object Detection** 9](#_Toc170682876)

[**3.2.** **Web Scrapping for Price Information** 9](#_Toc170682877)

[**3.3.** **Seamless User Interface** 9](#_Toc170682878)

[**4.** **Related System Analysis/Literature Review** 10](#_Toc170682879)

[**4.1.** **Existing Similar Systems** 10](#_Toc170682880)

[**4.1.1.** **General Price Comparison Websites:** 10](#_Toc170682881)

[**4.1.2.** **Browser Extensions:** 10](#_Toc170682882)

[**4.1.3.** **Barcode Scanners***:* 10](#_Toc170682883)

[**4.2.** **Existing Similar Systems in Tabular Form** 11](#_Toc170682884)

[**5.** **Advantages and Benefits of Proposed System** 12](#_Toc170682885)

[**5.1.** **Enhanced User Convenience** 12](#_Toc170682886)

[**5.2.** **Accurate Item Identification** 12](#_Toc170682887)

[**5.3.** **Real-time Price Comparison** 12](#_Toc170682888)

[**5.4.** **Comprehensive and Up-to-date Data** 12](#_Toc170682889)

[**5.5.** **Cost Savings** 12](#_Toc170682890)

[**5.6.** **Seamless User Interface** 12](#_Toc170682891)

[**5.7.** **Personalized Shopping Experience** 12](#_Toc170682892)

[**5.8.** **Wide Range of Retailers** 12](#_Toc170682893)

[**5.9.** **Increased Efficiency** 13](#_Toc170682894)

[**5.10.** **Enhanced Decision -Making** 13](#_Toc170682895)

[**6.** **Scope** 14](#_Toc170682896)

[**6.1.** **Functionality** 14](#_Toc170682897)

[**6.1.1.** **Item Identification** 14](#_Toc170682898)

[**6.1.2.** **Price Comparison** 14](#_Toc170682899)

[**6.1.3.** **Additional Features (Optional for future development)** 14](#_Toc170682900)

[**6.2.** **Technical Considerations** 14](#_Toc170682901)

[**6.2.1.** **Image Recognition Model** 14](#_Toc170682902)

[**6.2.2.** **Data Scrapping** 14](#_Toc170682903)

[**6.3.** **Exclusions** 14](#_Toc170682904)

[**6.4.** **Platform** 15](#_Toc170682905)

[**6.5.** **Future Development** 15](#_Toc170682906)

[**7.** **System Limitations/Constraints** 16](#_Toc170682907)

[**7.1.** **Limitations** 16](#_Toc170682908)

[**7.1.1.** **Image Recognition Accuracy** 16](#_Toc170682909)

[**7.1.2.** **Data Scrapping Challenge** 16](#_Toc170682910)

[**7.1.3.** **Limited Scope (Initial Version)** 16](#_Toc170682911)

[**7.2.** **Constraints** 16](#_Toc170682912)

[**7.2.1.** **Computational Resources** 16](#_Toc170682913)

[**7.2.2.** **Real-Time Price Updates** 16](#_Toc170682914)

[**7.2.3.** **Ethical Consideration** 16](#_Toc170682915)

[**8.** **Software Process Methodology** 17](#_Toc170682916)

[**8.1.** **Flexibility** 17](#_Toc170682917)

[**8.2.** **Reduced Risk** 17](#_Toc170682918)

[**8.3.** **Prioritization** 17](#_Toc170682919)

[**8.4.** **Better Quality Control** 17](#_Toc170682920)

[**8.5.** **Adaptability to Change** 17](#_Toc170682921)

[**9.** **Tools and Technologies** 18](#_Toc170682922)

[**10.** **Project Stakeholders and Roles** 18](#_Toc170682923)

[**11.** **Data Gathering Approach** 19](#_Toc170682924)

[**11.1.** **Using Kaggle data set** 19](#_Toc170682925)

[**11.2.** **Doing web scraping** 19](#_Toc170682926)

[**12.** **Concepts** 19](#_Toc170682927)

[**13.** **Gantt Chart** 20](#_Toc170682928)

[**13.1.** **Gantt Chart using MS Project** 20](#_Toc170682929)

[**13.2.** **Gantt Chart using MS Excel** 21](#_Toc170682930)

[**14.** **Network Diagram** 22](#_Toc170682931)

[**14.1.** **Activities Details** 22](#_Toc170682932)

[**14.2.** **Activity on Node** 22](#_Toc170682933)

[**14.2.1.** **Notations** 22](#_Toc170682934)

[**14.2.2.** **Diagram** 22](#_Toc170682935)

[**14.3.** **Activity on Arrow** 23](#_Toc170682936)

[**15.** **Work Breakdown Structure (WBS)** 24](#_Toc170682937)

[**16.** **Cost Estimation of Project using COCOMO-II Model** 25](#_Toc170682938)

[**16.1.** **Basic Model** 25](#_Toc170682939)

[**16.1.1.** **Effort:** 25](#_Toc170682940)

[**16.1.2.** **Development Time** 25](#_Toc170682941)

[**16.1.3.** **Staff Size** 26](#_Toc170682942)

[**16.1.4.** **Productivity** 26](#_Toc170682943)

[**16.2.** **Intermediate Model** 27](#_Toc170682944)

[**16.2.1.** **EAF according to Project** 28](#_Toc170682945)

[**16.2.2.** **Effort Adjustment Factor** 29](#_Toc170682946)

[**16.2.3.** **Effort:** 29](#_Toc170682947)

[**16.2.4.** **Development Time** 29](#_Toc170682948)

[**16.2.5.** **Staff Size** 29](#_Toc170682949)

[**16.2.6.** **Productivity** 29](#_Toc170682950)

[**16.3.** **Detailed Model** 30](#_Toc170682951)

[**16.3.1.** **Major Phases** 30](#_Toc170682952)

[**16.3.2.** **Distribution** **of** **Effort** 30](#_Toc170682953)

[**16.3.3.** **Effort** **for** **each** **Phase** 30](#_Toc170682954)

[**16.3.4.** **Development** **Time** **for** **each** **Phase** 31](#_Toc170682955)

[**17.1.** **Risk Identification** 32](#_Toc170682956)

[**17.2.** **Risk Analysis using SWOT** 33](#_Toc170682957)

[**18.** **References** 33](#_Toc170682958)

[**19.** **Plagiarism Report** 33](#_Toc170682959)

**Project Category: (**Select all the major domains of proposed project**)**

* **A-**Mobile Application/Information System **B-**Web Application/Mobile Application **C-** Problem Solving and Artificial Intelligence ** D-**Simulation and Modelling ** E-** Smartphone Application ** F-** Smartphone Game ** G-** Networks ** H-** Image Processing****Other (specify category) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

The endeavor at hand is the development of an AI-powered Shopping Optimizing APP, (i.e. Saving Bot) designed to revolutionize the shopping experience. Targeting the individuals seeking the best deals prices on fashion items. This application will use advanced image recognition algorithms to identify the items uploaded by users. The algorithm that we will use is VGG-16 which is an image recognition model that is used to identify the objects. Hence this algorithm will swiftly identify the attributes of items like color, patterns, and style facilitating precise searches across multiple online stores.

Upon user submission, the application systematically searches various online platforms extracting information from variety of retailers. Using its skills in computer programming and data analysis, the app collects a lot of price options. Users then see a list of stores with prices and links to buy things easily.

The project combines advanced AI-Technology with user-friendly interface to streamline shopping experience for our target users. By giving users instant access to the best deals, the app changes how easy and valuable shopping experience can be.

# **Introduction**

In the current digital era, technology continues to transform the way we interact with the world, including how we shop. One of the most important advancements in this domain is the integration of artificial intelligence (AI) and machine learning (ML) into everyday applications. AI technologies, particularly in the fields of image recognition and data analysis, are revolutionizing various industries by enabling more intelligent and efficient processes.

**Image recognition** is a branch of AI (artificial intelligence) that encompasses the training algorithms to identify and classify objects within images. These algorithms, such as the widely used VGG-16 model, can analyses visual data to detect specific features like color, shape, and patterns. This technology has many applications, from facial recognition systems to automated quality control in manufacturing and industry.

Another crucial and vital technology is **web scraping**, a technique used to extract data from websites. By automating the process of gathering information, web scraping allows for the efficient and convenient collection of large datasets from multiple sources. This data can then be analyzed and used to provide valuable insights, such as comparing prices across different online retailers, which can also be used to make predictions.

Combining these advanced technologies creates powerful tools that can significantly enhance user experiences in various domains, including online shopping. Online shopping has become a staple of modern life, offering convenience and a vast array of choices. However, finding the best deals can be time-consuming and frustrating due to the overwhelming number of options and the difficulty in comparing prices across different websites.

To address these challenges, we propose the development of the **Saving Bot**, an AI-powered shopping optimization app designed to streamline the online shopping experience. The Saving Bot leverages the capabilities of image recognition and web scraping to offer a unique and efficient solution for finding the best deals on fashion items.

Users can simply upload a picture of the clothing item they are interested in, and the app's sophisticated image recognition algorithm, based on the VGG-16 model, accurately identifies the item's attributes. The app then utilizes advanced web scraping techniques to gather real-time pricing information from numerous online retailers. By compiling and comparing these prices, the Saving Bot presents users with a comprehensive list of options, ensuring they can easily find the best deals available.

This project not only aims to make online shopping more accessible and affordable but also promises to enhance user satisfaction by providing a seamless and intuitive interface. By integrating cutting-edge AI technology with practical application, the Saving Bot stands to revolutionize the way users shop for fashion items online, offering a smarter and more efficient alternative to traditional methods.

# **Problem Statement**

Online Shopping is a Hellish experience! When you find the perfect outfit, but only in size they haven’t invented it yet. Shoppers are frustrated with the unlimited number of stores, the hassle of finding the best price, and have difficulty comparing prices and outfits and sizes across different products. Now in this continuously evolving world what we need is a user-friendly app that uses curing edge technology to automate this process and do the work of comparison for us.

# **Problem Solution for Proposed System**

The Saving Bot tackles the challenges of online shopping with a multi-functional approach, featuring these key elements:

## **Object Detection**

The app uses VGG-16 for the detection of clothes from the images uploaded by the user to accurately identify the style, color, and pattern of the clothes, enabling precise searches and comparisons.

## **Web Scrapping for Price Information**

Advanced web scraping techniques are used to gather real time pricing information from numerous online retailers, ensuring that the user will have access to the most up-to-date data.

## **Seamless User Interface**

A user-friendly interface is designed to provide an intuitive and hassle-free shopping experience, allowing users to navigate the app effortlessly and make informed choices.

# **Related System Analysis/Literature Review**

## **Existing Similar Systems**

### **General Price Comparison Websites:**

Websites like Google shopping and Shopzilla allow users to search for a specific item and compare prices across various online stores. However, these websites require users to know the item's name or brand beforehand. They lack the image recognition capability of your proposed application.

### **Browser Extensions:**

Browser extensions like Honey or Invisible Hand automatically search for coupons and price comparisons while users shop online. However, they are limited to the store the user is currently browsing and cannot search for an item across multiple stores based on an image.

### **Barcode Scanners**:

Mobile apps like Shop Savvy or Amazon app allow users to scan barcodes of physical products to find online prices. They require the product to have a barcode and cannot identify items based solely on an image.

## **Existing Similar Systems in Tabular Form**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | Similar Systems | Similar Functionality | Limitation | Proposed Project Solution |
| 1 | Google shopping | General Price Comparison Websites | Can’t do image recognition. | Uses image recognition to identify items from an image uploaded. |
| 2 | Honey | Browser Extensions | Limited to Current Online Store. | Searches for Price Comparison on various stores based on image uploaded. |
| 3 | Shop Savvy | Barcode Scanners | Require physical products with barcode. | Identify item from an image eliminating the need for a barcode. |
| 4 | Invisible Hand | Browser Extensions | Limited to Current Online Store. | Searches for Price Comparison on various stores based on image uploaded. |
| 5 | Shopzilla | General Price Comparison Websites | Can’t do image recognition. | Uses image recognition to identify items from an image uploaded. |
| 6 | Amazon | Barcode Scanners | Require physical products with barcode. | Identify item from an image eliminating the need for a barcode. |

Table 1 : Related System Analysis with proposed project solution

# **Advantages and Benefits of Proposed System**

## **Enhanced User Convenience**

Users can easily upload pictures of desired fashion items, eliminating the need to manually search through numerous websites. This streamlines the shopping process, saving valuable time and effort.

## **Accurate Item Identification**

The sophisticated image recognition algorithm, based on the VGG-16 model, ensures precise identification of clothing items. This accuracy helps in providing relevant search results and enhances the overall user experience.

## **Real-time Price Comparison**

By leveraging advanced web scraping techniques, the Saving Bot gathers real-time pricing information from multiple online retailers. This allows users to compare prices quickly and efficiently, ensuring they find the best deals available.

## **Comprehensive and Up-to-date Data**

The system continuously updates its database with the latest prices and product availability, providing users with the most current information. This helps in making informed purchasing decisions.

## **Cost Savings**

The primary goal of the Saving Bot is to help users find the best deals on fashion items. By comparing prices across different retailers, users can identify significant cost savings, making their shopping experience more affordable.

## **Seamless User Interface**

The app is designed with a user-friendly and intuitive interface, making it accessible to a wide range of users, regardless of their technical expertise. This ease of use enhances overall user satisfaction.

## **Personalized Shopping Experience**

The AI algorithms can learn from user preferences and past behaviours, offering personalized recommendations and improving the relevance of search results over time. This personalization makes the shopping experience more tailored to individual needs.

## **Wide Range of Retailers**

The web scraping capabilities allow the Saving Bot to gather data from a vast array of online retailers, ensuring a comprehensive selection of options. This variety helps users find not only the best prices but also the best products.

## **Increased Efficiency**

Automating the process of price comparison and product identification significantly reduces the time and effort required for online shopping. Users can focus on other tasks while the Saving Bot does the heavy lifting.

## **Enhanced Decision -Making**

With access to detailed information and price comparisons, users can make more informed purchasing decisions. This transparency helps in avoiding buyer’s remorse and ensures satisfaction with the purchased items.

# **Scope**

## **Functionality**

### **Item Identification**

* + **Scope:** The application will focus on identifying clothing and fashion items in user-uploaded images.
  + **Accuracy:** The application will strive for high accuracy, aiming to identify similar items with slight variations in color or pattern.

### **Price Comparison**

* + **Scope:** The application will focus on identifying clothing and fashion items in user-uploaded images.
  + **Accuracy:** The application will strive for high accuracy, aiming to identify similar items with slight variations in color or pattern.

### **Additional Features (Optional for future development)**

* + Users can filter search results by brand, size, or color.
  + The application can suggest similar items based on the uploaded image and user preferences.
  + Integration with user reviews from trusted sources can be considered.

## **Technical Considerations**

### **Image Recognition Model**

The application will leverage a pre-trained image recognition model like VGG-16, fine-tuned on a relevant fashion image dataset (e.g., Kaggle Fashion Dataset).

### **Data Scrapping**

* + The application will ethically scrape price data from online stores, respecting their terms of service.
  + The system will be designed to handle data from multiple stores efficiently.

## **Exclusions**

* The application will not include features like barcode scanning or in-app purchasing capabilities (users will be directed to retailer websites for purchases).
* Initially, the application will focus on English language support.

## **Platform**

The application will be developed as a cross-platform mobile app (iOS and Android) for wider user accessibility.

## **Future Development**

Based on user feedback and market demands, the scope can be expanded to include additional product categories, a wider range of stores, and more advanced features.

# **System Limitations/Constraints**

## **Limitations**

### **Image Recognition Accuracy**

VGG-16, while powerful, might not perfectly capture subtle variations in clothing items (like slight differences in fabric texture or embellishments). This could lead to inaccurate price comparisons for similar but not identical items.

### **Data Scrapping Challenge**

Websites may change their layout or data structure frequently, requiring the app to adapt its scraping techniques to maintain functionality.

Retailers might implement anti-scraping measures, making it difficult for the app to access price data reliably.

### **Limited Scope (Initial Version)**

Focusing on clothing initially limits the app's usability for other product categories.

Including only a few stores restricts users' access to the broadest range of deals.

## **Constraints**

### **Computational Resources**

Running image recognition models and scraping data requires processing power. This may limit the app's scalability to a large user base or restrict the number of stores it can search simultaneously.

### **Real-Time Price Updates**

Constantly checking prices across multiple stores can be resource intensive. Striking a balance between update frequency and efficient data usage is crucial.

### **Ethical Consideration**

Data scraping must comply with website terms of service and avoid overloading their servers.

# **Software Process Methodology**

Agile development is ideal for the Saving Bot Mobile Application for several reasons:

* 1. **Flexibility**

As user needs or senior technology adoption evolves, agile development allows for incorporating new features or functionalities efficiently.

* 1. **Reduced Risk**

By delivering features in small increments, agile development minimizes the risk of investing heavily in features that seniors may not find useful.

* 1. **Prioritization**

Agile prioritizes the most impactful features first, ensuring the core functionalities that enhance well-being are delivered quickly.

## **Better Quality Control**

Regular testing and continuous feedback in Agile development ensure a high-quality, reliable application.

## **Adaptability to Change**

Agile's focus on responsiveness allows the application to quickly adapt to new requirements, user feedback, and technological advancements.

# **Tools and Technologies**

We will be using the following tools and technologies for building this project.

|  |  |  |  |
| --- | --- | --- | --- |
| Tools  And  Technologies | Tools | Version | Rationale |
| MS Visual Studio | 1.82 | IDE |
| Mongo DB | 2015 | DBMS |
| MS Project | CSC 6 | Management |
| MS Word | 2015 | Documentation |
| MS Power Point | 2015 | Presentation |
| Star UML | 2.0.5 | Mock-ups Creation |
| **Technology** | **Version** | **Rationale** |
| React Native | 0.72 | Framework |
| Express JS | 4.19.1 | Framework |
| Node JS | 21.0.0 | Run time environment |
| Java Script | ES 13 | Scripting Language |

Table 2: Tools and Technologies for Proposed System

# **Project Stakeholders and Roles**

Write down the project stakeholders and their roles.

|  |  |
| --- | --- |
| **Project Sponsor** | COMSATS University, Islamabad Committee |
| **Stakeholder** | Mention your stakeholders with their roles and responsibilities.  Default option will be:   * Group Members: BASIT IQBAL, Fatima Aftab, Waleed Rashid * Project Supervisor Name: Mr. Syed Shahab Zarin * Final Year Project Committee: Evaluation of project |

Table 3: Project Stakeholders and their Roles

# **Data Gathering Approach**

The data gathering approach for this project will include:

## **Using Kaggle data set**

Leveraging pre-existing, high-quality datasets from Kaggle to obtain relevant information efficiently.

## **Doing web scraping**

Extracting real-time data from websites to ensure the application has up-to-date and comprehensive information.

# **Concepts**

Concepts to be learned during this project include:

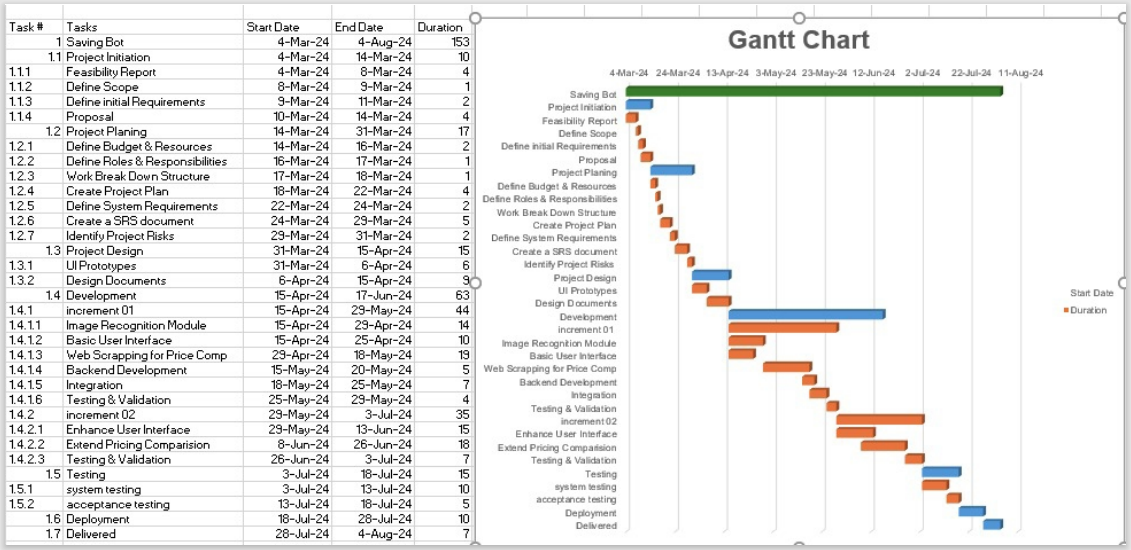
1. Object Detection techniques.
2. Using Kaggle Data set to Train Model (AI).
3. Web Scrapping
4. User Authentication and Authorization
5. Database design and management
6. Responsive Mobile App design

# **Gantt Chart**

## **Gantt Chart using MS Project**

Figure 1: Gantt Chart using MS Project

## **Gantt Chart using MS Excel**

****F*igure 2: Gantt Chart using MS Excel*

# **Network Diagram**

## **Activities Details**

|  |  |  |
| --- | --- | --- |
| Activity Name | Duration | Predecessor |
| Project Initiation | 11 | - |
| Project Plan | 17 | Project Initiation |
| Project Design | 15 | Project Plan |
| Development | 79 | Project Design |
| Testing | 15 | Development |
| Deployed | 10 | Testing |
| Delivered | 7 | Delivered |

**Table 4: Activities Details**

## **Activity on Node**

### **Notations**

Figure 3: Notation

### **Diagram**

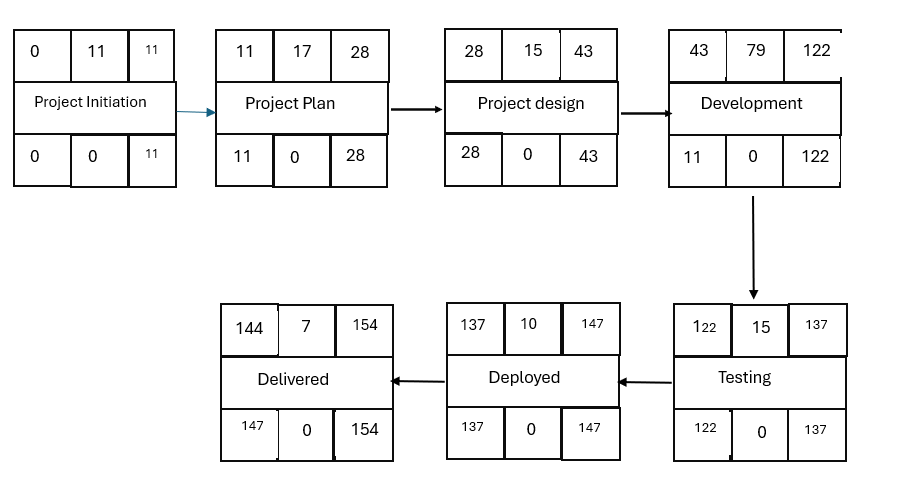
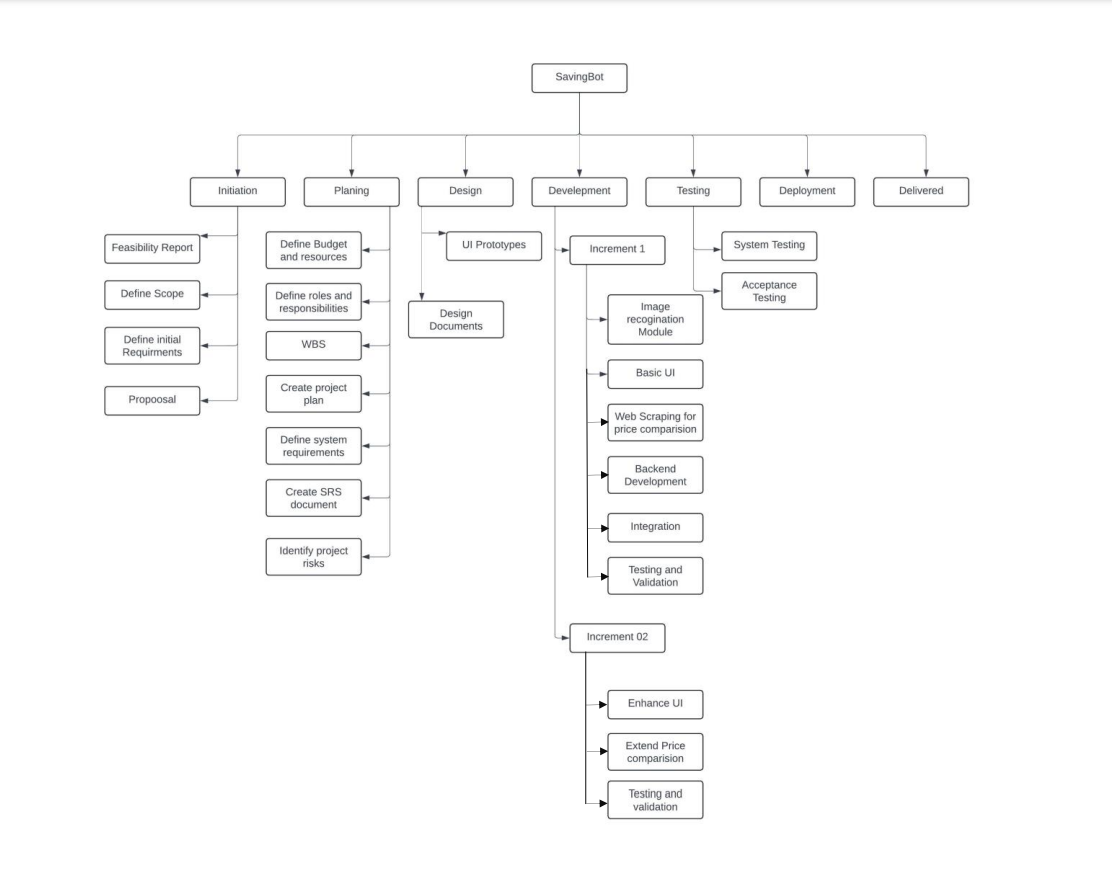


Figure 4:Activity on Node Detailed

## A diagram of a project Description automatically generated**Activity on Arrow**

Figure 5: Activity on Arrow

# **Work Breakdown Structure (WBS)**

Figure 6: Work Breakdown Structure of System

# **Cost Estimation of Project using COCOMO-II Model**

**Total Line of Code:**

Total line of Code = 200KLOC

**Type of Project:**

As there are 200KLOC so it falls under the category of “semidetached”.

## **Basic Model**

According to the type of project we will pick the value of a, b, c, d from the following table.

Figure 7: Constant Values According to type of Project.

According to the table a=3.0, b=1.12, c=2.5, d=0.35.

Now we start the calculation.

### **Effort:**

E= 3.0 (

E= 3.0×545.01

E= 1133.12 Person-Months

### **Development Time**

D= 2.5

D=2.5×11.72

D=29.30 Months

### **Staff Size**

SS=

SS= 38.67 People

### **Productivity**

P=

P= 0.1765 KLOC/PM

## **Intermediate Model**

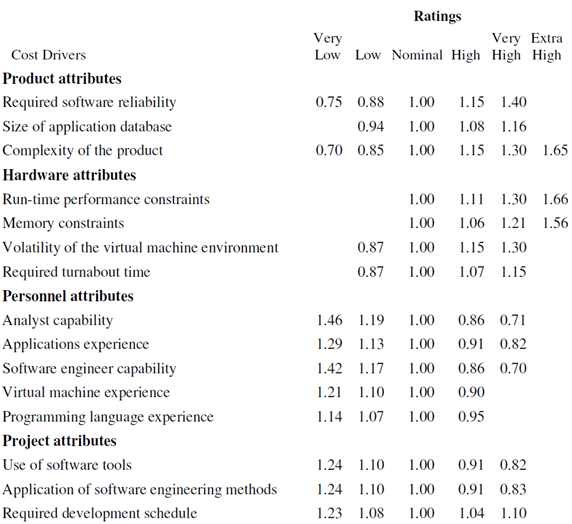
Now we will find the factors form the following table:

Figure 8: Factors of Intermediate Model

### **EAF according to Project**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No | Attribute | Level | Reason of selecting level | Value from table |
| 1 | Required software reliability extent | High | Financial data (price comparisons) is involved, so minimal errors and crashes are crucial. | 1.15 |
| 2 | Size of the application database | Low | As it would store only the user info and some basic info related to the items | 0.94 |
| 3 | The complexity of the product | High | As first it would detect the item and then it will web scrape the items related to its which are available then it will compare the prices and at last it will show the items. | 1.15 |
| 4 | Run-time performance constraints | Very High | As the bot is using third part VGG-16 model to detect the item so it might take some time then it must scrape the websites for better results so it will take time | 1.30 |
| 5 | Memory constraints | Low | As the app wouldn’t store that much info | 1.00 |
| 6 | Required turnabout time | Normal | Instant responses aren't crucial. Users likely expect a few seconds for search results. | 1.00 |
| 7 | Analyst capability | Normal | As the analyst will have to understand the fashion trends, which keep on changing | 1.00 |
| 8 | Programming language experience | Normal | As we would be using VGG-16 which is per-built but we need a good experience in coding to do web-scraping and for front and backend development. | 1.00 |

Table 5: Project EAF

### **Effort Adjustment Factor**

EAF = ∏ (factors)

EAF = 1.62

### **Effort:**

E= 3.0 ( \* 1.62

E= 3.0×545.01 × 1.62

E= 1835.12 Person-Months

### **Development Time**

D= 2.5

D=2.5×11.72

D= 29.30 Months

D=29.30 Months

### **Staff Size**

SS =

SS= 62.63 Persons

### **Productivity**

P=

P= 0.109 KLOC/Person Months

## **Detailed Model**

To provide a detailed COCOMO estimation for the project with the identified major components, we'll allocate the total effort calculated earlier (1835.12 Person-Months) across these phases. Here’s how we can break it down:

### **Major Phases**

1. Responsive Mobile App design
2. Database design and management
3. Web Scraping
4. User Authentication and Authorization
5. Object Detection by VGG-16

### **Distribution** **of** **Effort**

We'll assume a reasonable distribution of effort across these phases. This distribution will depend on the complexity and relative effort each component typically requires. A possible distribution could be as follows:

1. Responsive Mobile App design: 25%
2. Database design and management: 20%
3. Web Scraping: 15%
4. User Authentication and Authorization: 20%
5. Object Detection by VGG-16: 20%

### **Effort** **for** **each** **Phase**

#### Responsive Mobile App design:

E= 1835.12×0.25

E=458.78Person-Months

#### Database design and management:

E=1835.12×0.20

E=367.02 Person-Months

#### Web Scraping:

E=1835.12×0.15

E=275.27 Person-Months

#### User Authentication and Authorization:

E=1835.12×0.20

E=367.02 Person-Months

#### Object Detection by VGG-16:

E=1835.12×0.20

E=367.02 Person-Months

### **Development** **Time** **for** **each** **Phase**

To calculate the development time for each component, we will use the formula:

Where, 𝑐=2.5 and 𝑑=0.35

Let's calculate the development time for each component.

#### Responsive Mobile App design:

𝐷=2.5×

D=7.85 Months

#### Database design and management:

𝐷=2.5×

D=7.17 Months

#### Web Scraping:

𝐷=2.5×

D=6.26 Months

#### User Authentication and Authorization:

𝐷=2.5×

D=7.17 Months

#### Object Detection by VGG-16:

𝐷=2.5×

D=7.17 Months

1. **Risk Management**

## **Risk Identification**

|  |  |
| --- | --- |
| Risk | Potential Indicators |
| Technology | * Compatibility issues with mobile platforms (iOS, Android) * Slow performance in image processing * Delays in image recognition accuracy improvement * Issues integrating web scraping tools * Security vulnerabilities in web scraping modules * Failure to maintain up-to-date technology stack |
| People | * High turnover rates among team members * Inadequate training for new technology * Low motivation among team members * Interpersonal conflicts * Insufficient expertise in AI and machine learning * Poor collaboration * Low team engagement |
| Organizational | * Conflict between project goals and academic requirements * Lack of support from faculty * Delays in decision making * Internal miscommunications * Limited resource allocation |
| Tools | * Difficulty in setting up and maintaining development environment * Lack of tool support * Resistance to using selected development tools * Frequent tool crashes * Complaints about efficiency * Incompatibility between chosen software tools * Lack of necessary features |
| Requirements | * Ambiguity in project requirements * Conflicting requirements from different stakeholders (professors, team members) * Frequent changes in project scope * Unclear requirements from stakeholders * Late requirement additions * Incomplete requirement specifications * High rate of requirement changes |
| Estimation | * Inaccurate project timelines – * Delays in meeting milestones – * Unexpected defects impacting deadlines – * Underestimation of project complexity – * Unrealistic deadlines imposed by semester schedule |

Table 6: Project Risks

## **Risk Analysis using SWOT**

### **Strength**

### **Weakness**

### **Opportunity**

### **Threats**

# **References**

<https://www.kaggle.com/datasets/zalando-research/fashionmnist>

<https://www.shopzilla.com/>

<https://www.google.com/shopping?hl=en>

# **Plagiarism Report**

Attach the Plagiarism report of your project scope document from library staff of Turnitin tool:

Sir, please add here as we don’t have Turnitin instructor account.