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

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 a 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.

1. 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.

2. 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.

3. PROBLEM SOLUTION FOR PROPOSED SYSTEM

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

3.1. 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.

3.2. 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.

3.3. 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.

4. RELATED SYSTEM ANALYSIS/LITERATURE REVIEW

4.1. Existing Similar Systems

4.1.1. 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.

4.1.2. 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.

4.1.3. 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.

4.2. 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

5. ADVANTAGES AND BENEFITS OF PROPOSED SYSTEM

5.1. 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.

5.2. 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.

5.3. 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.

5.4. 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.

5.5. 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.

5.6. 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.

5.7. 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.

5.8. 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.

5.9. 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.

5.10. 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.

6. SCOPE

6.1. Functionality

6.1.1. 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.

6.1.2. 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.

6.1.3. 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.

6.2. Technical Considerations

6.2.1. 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).

6.2.2. 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.

6.3. 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.

6.4. Platform

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

6.5. 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.

7. SYSTEM LIMITATIONS/CONSTRAINTS

7.1. Limitations

7.1.1. 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.

7.1.2. 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.

7.1.3. 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.

7.2. Constraints

7.2.1. 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.

7.2.2. 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.

7.2.3. Ethical Consideration

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

8. SOFTWARE PROCESS METHODOLOGY

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

8.1. Flexibility

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

8.2. Reduced Risk

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

8.3. Prioritization

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

8.4. Better Quality Control

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

8.5. Adaptability to Change

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

9. TOOLS AND TECHNOLOGIES

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

	Tools	Version	Rationale
Tools And Technologies	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

10. 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

11. DATA GATHERING APPROACH

The data gathering approach for this project will include:

11.1. Using Kaggle data set

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

11.2. Doing web scraping

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

12. CONCEPTS

Concepts to be learned during this project include:

- Object Detection techniques.
- Using Kaggle Data set to Train Model (AI).
- Web Scrapping
- User Authentication and Authorization
- Database design and management
- Responsive Mobile App design

13. GANTT CHART

13.1. Gantt Chart using MS Project

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessor
1		**1 Saving Bot**	**154 days**	**Mon 3/4/24**	**Sun 8/4/24**	
2		**1.1 Project Initiation**	**11 days**	**Mon 3/4/24**	**Thu 3/14/24**	
3		1.1.1 Feasibility Report	5 days	Mon 3/4/24	Fri 3/8/24	
4		1.1.2 Define Scope	2 days	Fri 3/8/24	Sat 3/9/24	
5		1.1.3 Define initial Requirements	4 days	Sun 3/10/24	Wed 3/13/24	
6		1.1.4 Proposal	3 days	Tue 3/12/24	Thu 3/14/24	
7		**1.2 Project Planning**	**17 days**	**Fri 3/15/24**	**Sun 3/31/24**	2
8		1.2.1 Define Budget & Resources	2 days	Fri 3/15/24	Sat 3/16/24	
9		1.2.2 Define Roles & Responsibilitie	1 day	Sat 3/16/24	Sat 3/16/24	
10		1.2.3 WBS	1 day	Sun 3/17/24	Sun 3/17/24	
11		1.2.4 Create Project Plan	4 days	Mon 3/18/24	Thu 3/21/24	
12		1.2.5 Define System Requirements	2 days	Fri 3/22/24	Sat 3/23/24	
13		1.2.6 Create SRS Doc	6 days	Sun 3/24/24	Fri 3/29/24	
14		1.2.7 Identify Project Risks	2 days	Sat 3/30/24	Sun 3/31/24	
15		**1.3 Project Design**	**15 days**	**Mon 4/1/24**	**Mon 4/15/24**	7
16		1.3.1 UI Prototype	6 days	Mon 4/1/24	Sat 4/6/24	
17		1.3.2 Design Doc	9 days	Sun 4/7/24	Mon 4/15/24	
18		**1.4 Development**	**79 days**	**Tue 4/16/24**	**Wed 7/3/24**	15
19		**1.4.1 Increment 01**	**35 days**	**Tue 4/16/24**	**Mon 5/20/24**	
20		1.4.1.1 Imange Recognition Moc	14 days	Tue 4/16/24	Mon 4/29/24	
21		1.4.1.2 Basic UI	10 days	Tue 4/16/24	Thu 4/25/24	
22		1.4.1.3 Web Scrapping For Price	19 days	Tue 4/30/24	Sat 5/18/24	
23		1.4.1.4 Backend Dev	6 days	Wed 5/15/24	Mon 5/20/24	
24		1.4.1.5 Integratioin	7 days	Tue 5/14/24	Mon 5/20/24	
25		1.4.1.6 Testing & Validation	5 days	Thu 5/16/24	Mon 5/20/24	
26		**1.4.2 Increment 02**	**35 days**	**Thu 5/30/24**	**Wed 7/3/24**	
27		1.4.2.1 Enhanced UI	15 days	Thu 5/30/24	Thu 6/13/24	
28		1.4.2.2 Extended Price Comparis	18 days	Fri 6/14/24	Mon 7/1/24	
29		1.4.2.3 Testing & Validition	7 days	Thu 6/27/24	Wed 7/3/24	
30		**1.5 Testing**	**15 days**	**Thu 7/4/24**	**Thu 7/18/24**	18
31		1.5.1 System Testing	10 days	Thu 7/4/24	Sat 7/13/24	
32		1.5.2 Acceptance Testing	5 days	Sun 7/14/24	Thu 7/18/24	
33		1.6 Deployment	10 days	Fri 7/19/24	Sun 7/28/24	30
34		1.7 Delivered	7 days	Mon 7/29/24	Sun 8/4/24	33
35		1.8 Project Over	0 days	Sun 8/4/24	Sun 8/4/24	34

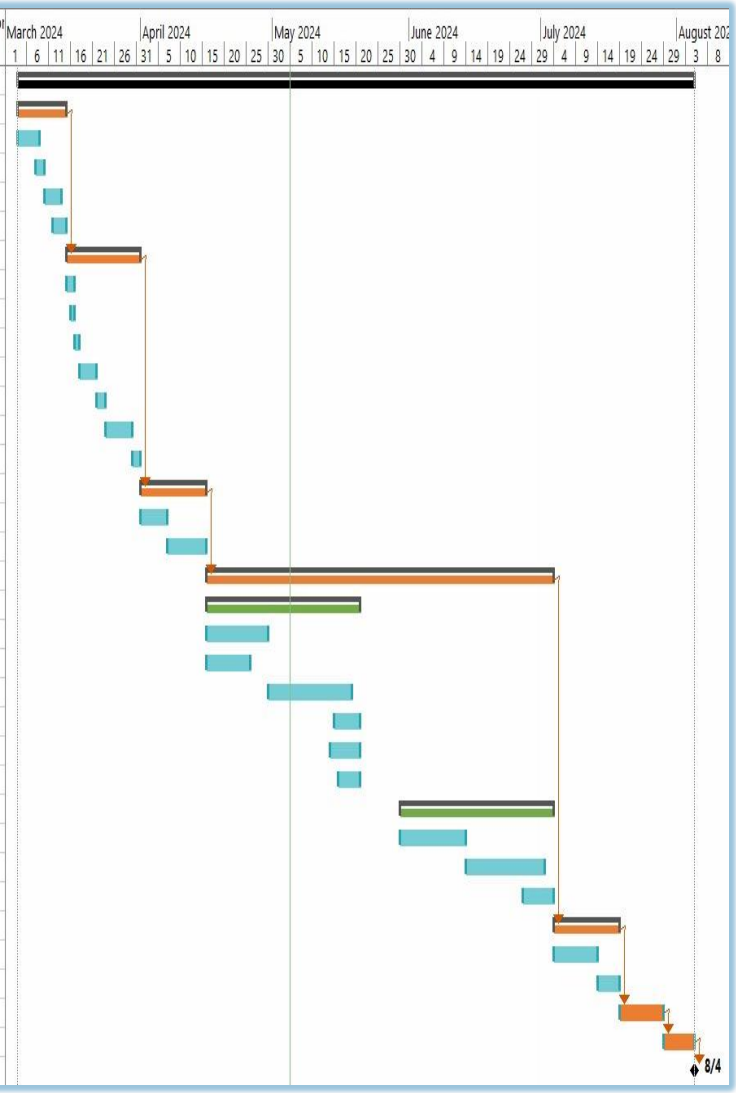

Figure 1:Gantt Chart using MS Project

13.2. Gantt Chart using MS Excel

Task #	Tasks	Start Date	End Date	Duration
1	Saving Bot	4-Mar-24	4-Aug-24	153
1.1	Project Initiation	4-Mar-24	14-Mar-24	10
1.1.1	Feasibility Report	4-Mar-24	8-Mar-24	4
1.1.2	Define Scope	8-Mar-24	9-Mar-24	1
1.1.3	Define initial Requirements	9-Mar-24	11-Mar-24	2
1.1.4	Proposal	10-Mar-24	14-Mar-24	4
1.2	Project Planing	14-Mar-24	31-Mar-24	17
1.2.1	Define Budget & Resources	14-Mar-24	16-Mar-24	2
1.2.2	Define Roles & Responsibilities	16-Mar-24	17-Mar-24	1
1.2.3	Work Break Down Structure	17-Mar-24	18-Mar-24	1
1.2.4	Create Project Plan	18-Mar-24	22-Mar-24	4
1.2.5	Define System Requirements	22-Mar-24	24-Mar-24	2
1.2.6	Create a SRS document	24-Mar-24	29-Mar-24	5
1.2.7	Identify Project Risks	29-Mar-24	31-Mar-24	2
1.3	Project Design	31-Mar-24	15-Apr-24	15
1.3.1	UI Prototypes	31-Mar-24	6-Apr-24	6
1.3.2	Design Documents	6-Apr-24	15-Apr-24	9
1.4	Development	15-Apr-24	17-Jun-24	63
1.4.1	increment 01	15-Apr-24	29-May-24	44
1.4.1.1	Image Recognition Module	15-Apr-24	29-Apr-24	14
1.4.1.2	Basic User Interface	15-Apr-24	25-Apr-24	10
1.4.1.3	Web Scrapping for Price Comp	23-Apr-24	18-May-24	19
1.4.1.4	Backend Development	15-May-24	20-May-24	5
1.4.1.5	Integration	18-May-24	25-May-24	7
1.4.1.6	Testing & Validation	25-May-24	29-May-24	4
1.4.2	increment 02	29-May-24	3-Jul-24	35
1.4.2.1	Enhance User Interface	29-May-24	13-Jun-24	15
1.4.2.2	Extend Pricing Comparision	8-Jun-24	26-Jun-24	18
1.4.2.3	Testing & Validation	26-Jun-24	3-Jul-24	7
1.5	Testing	3-Jul-24	18-Jul-24	15
1.5.1	system testing	3-Jul-24	13-Jul-24	10
1.5.2	acceptance testing	13-Jul-24	18-Jul-24	5
1.6	Deployment	18-Jul-24	28-Jul-24	10
1.7	Delivered	28-Jul-24	4-Aug-24	7

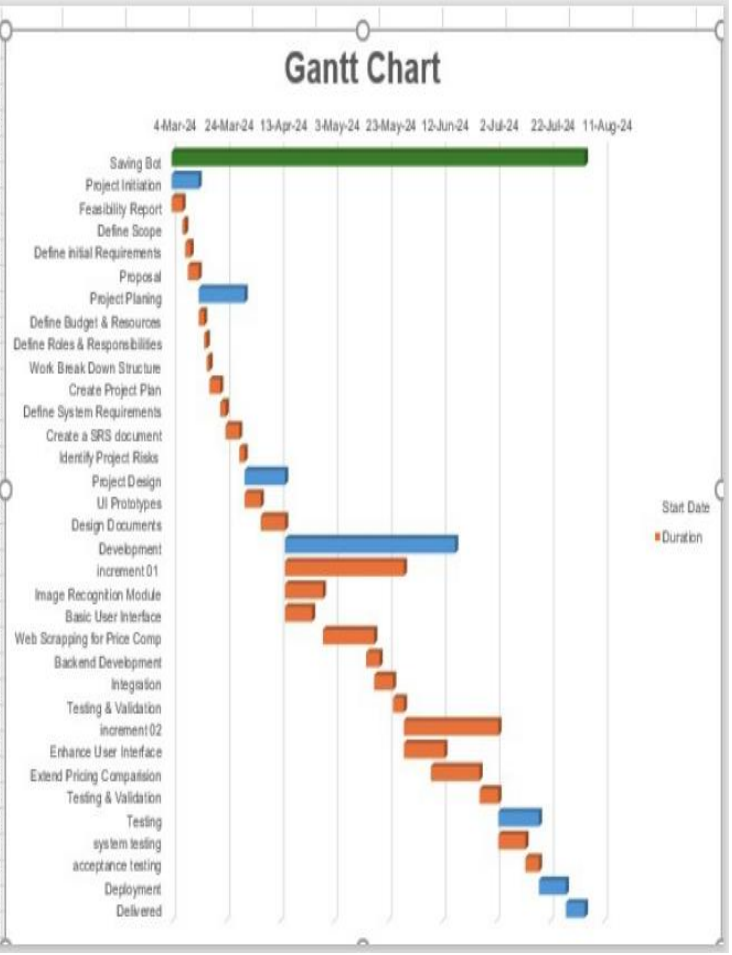

Figure 2:Gantt Chart using MS Excel

14. NETWORK DIAGRAM

14.1. 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

14.2. Activity on Node

14.2.1. Notations

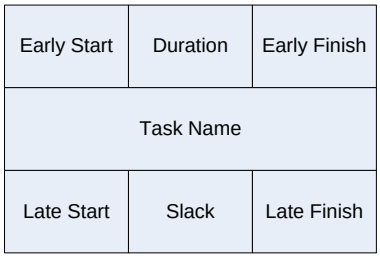

Figure 3:Notation

14.2.2. Diagram

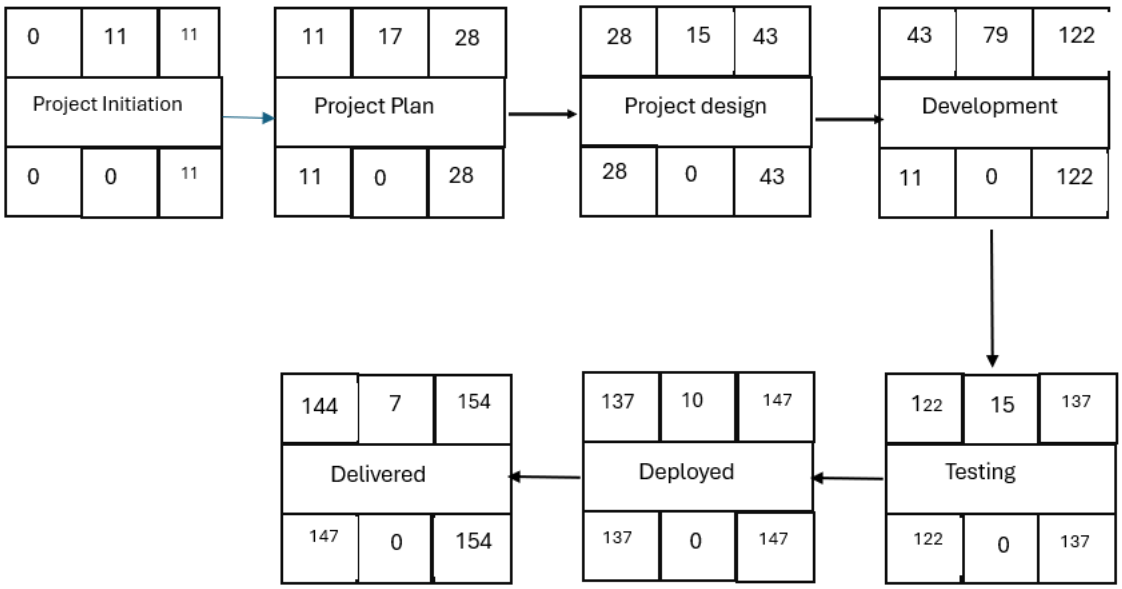

Figure 4:Activity on Node Detailed

14.2.3. Critical Path

Critical Path for the given project is,

Project initiation→Project Plan → Project Design→Development→Testing→Deployed→Delivered

14.3. Activity on Arrow

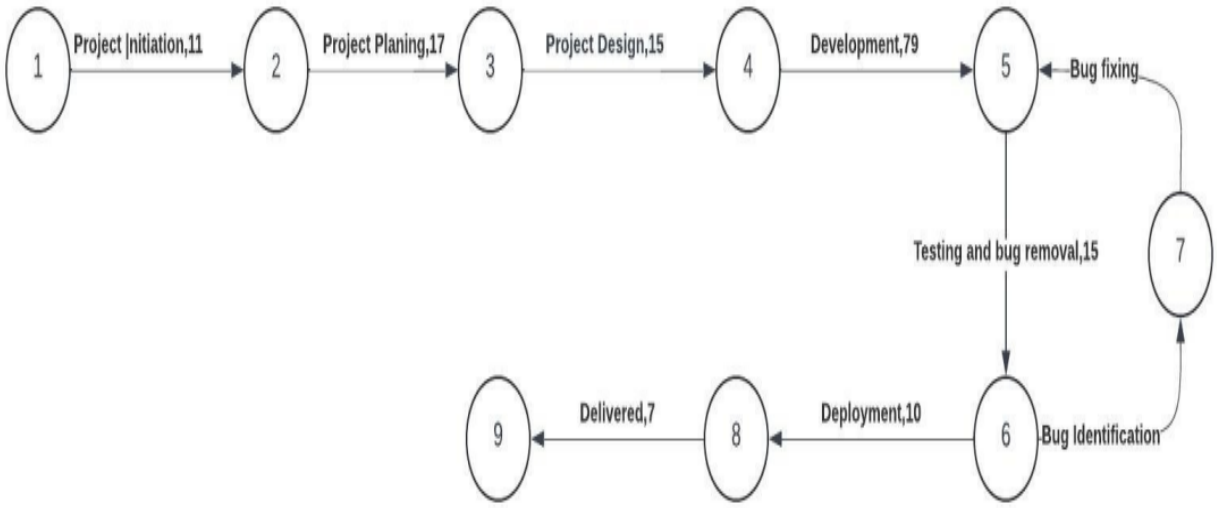

Figure 5:Activity on Arrow

15. WORK BREAKDOWN STRUCTURE (WBS)

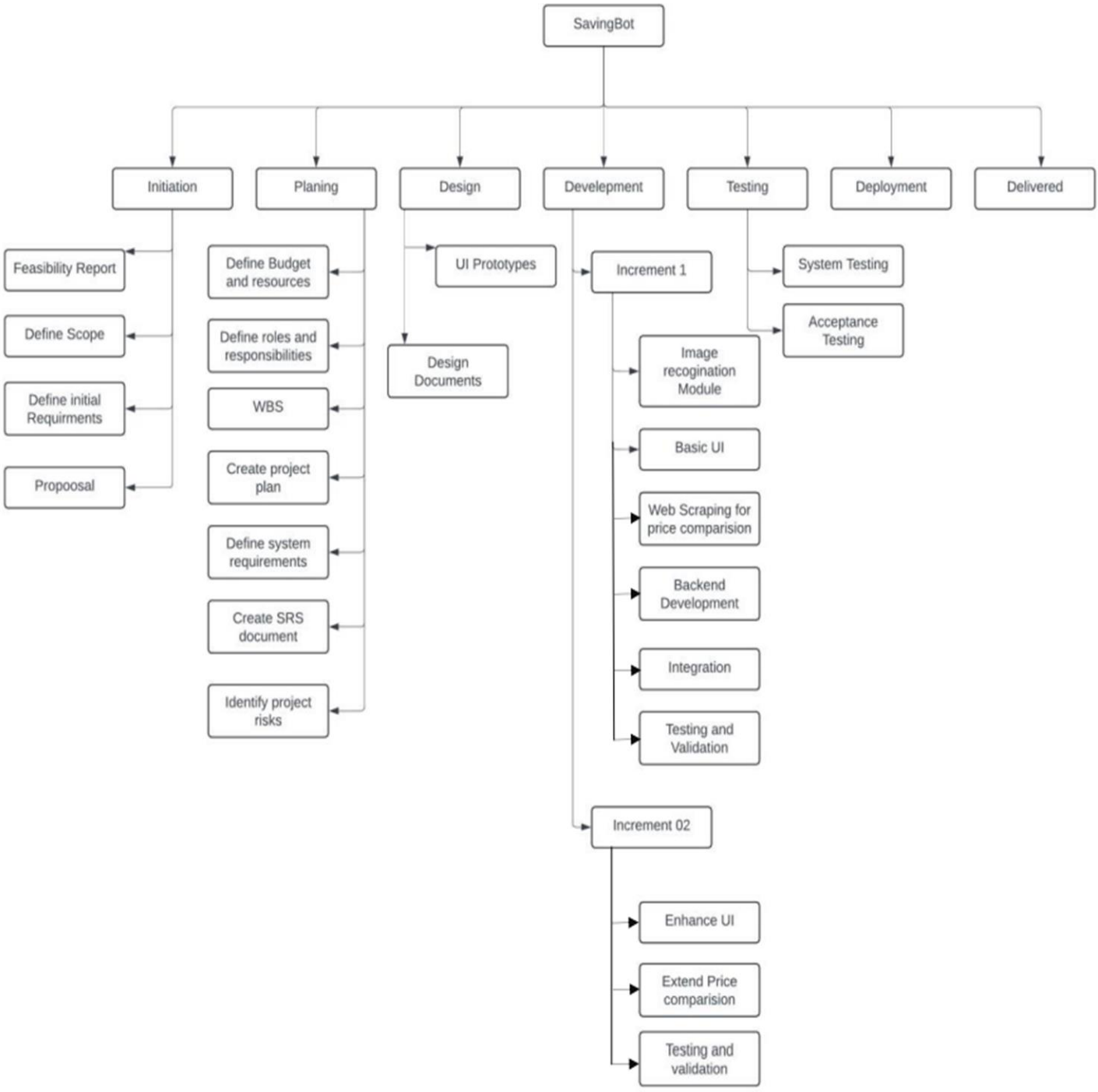

Figure 6:Work Breakdown Structure of System

16. COST ESTIMATION OF PROJECT

Total Line of Code:

Total line of Code = 200KLOC

Type of Project:

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

16.1. Basic Model

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

Software Project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

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.

16.1.1. Effort:

$$E = a_b(KLOC)^{b_b}$$

E= 3.0 $(200)^{1.12}$
E= 3.0×545.01
E= 1133.12 Person-Months

16.1.2. Development Time

$$D = c_b(Effort)^{d_b}$$

D= 2.5 $(1133.12^{0.35})$

D=2.5×11.72

D=29.30 Months

16.1.3. Staff Size

$$SS = \frac{Effort}{Development\ Time}$$

SS= $\frac{1133.12}{29.30}$

SS= 38.67 People

16.1.4. Productivity

$$P = \frac{KLOC}{Effort}$$

P= $\frac{200}{1133.12}$

P= 0.1765 KLOC/PM

16.2. Intermediate Model

Now we will find the factors form the following table:

	Ratings					
Cost Drivers	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Use of software tools	1.24	1.10	1.00	0.91	0.82	
Application of software engineering methods	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Figure 8:Factors of Intermediate Model

16.2.1. 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

16.2.2. Effort Adjustment Factor

EAF = ∏ (factors)
EAF = 1.62

16.2.3. Effort:

$$E = a_b(KLOC)^{b_b} * EAF$$

E= 3.0 $(200)^{1.12}$ * 1.62
E= 3.0×545.01 × 1.62
E= 1835.12 Person-Months

16.2.4. Development Time

$$D = c_b(Effort)^{d_b}$$

D= 2.5 $(1835.12^{0.35})$

D=2.5×11.72

D= 29.30 Months

D=29.30 Months

16.2.5. Staff Size

$$SS = \frac{Effort}{Development\ Time}$$

SS = $\frac{1835.12}{29.30}$

SS= 62.63 Persons

16.2.6. Productivity

$$P = \frac{KLOC}{Effort}$$

P= $\frac{200}{1835.12}$

P= 0.109 KLOC/Person Months

16.3. 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:

16.3.1. 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

16.3.2. 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%

16.3.3. Effort for each Phase

16.3.3.1. Responsive Mobile App design:

E= 1835.12×0.25

E=458.78Person-Months

16.3.3.2. Database design and management:

E=1835.12×0.20

E=367.02 Person-Months

16.3.3.3. Web Scraping:

E=1835.12×0.15

E=275.27 Person-Months

16.3.3.4. User Authentication and Authorization:

E=1835.12×0.20

E=367.02 Person-Months

16.3.3.5. Object Detection by VGG-16:

E=1835.12×0.20

E=367.02 Person-Months

16.3.4. Development Time for each Phase

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

$$D = c_b(Effort)^{d_b}$$

Where, c=2.5 and d=0.35

Let's calculate the development time for each component.

16.3.4.1. Responsive Mobile App design:

D=2.5×$(458.78)^{0.35}$

D=7.85 Months

16.3.4.2. Database design and management:

D=2.5×$(367.02)^{0.35}$

D=7.17 Months

16.3.4.3. Web Scraping:

D=2.5×$(275.27)^{0.35}$

D=6.26 Months

16.3.4.4. User Authentication and Authorization:

D=2.5×$(367.02)^{0.35}$

D=7.17 Months

16.3.4.5. Object Detection by VGG-16:

D=2.5×$(367.02)^{0.35}$

D=7.17 Months

17. RISK MANAGEMENT

17.1. 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's Risks

17.2. Risk Analysis using SWOT

17.2.1. Strength

- **Innovative Use of AI:** Utilizes advanced AI for object detection, offering a distinct advantage over traditional comparison tools.
- **Multi-Platform Availability:** Cross-platform design ensures accessibility on both iOS and Android devices.
- **High User Value***:* Addresses significant user pain points in online shopping, providing a highly valuable service.

17.2.2. Weakness

- **High Development Costs:** Requires substantial investment for development and ongoing maintenance.
- **User Privacy Concerns:** Handling of user-uploaded images may raise significant privacy issues.
- **Technical Limitations:** Potential inaccuracies in image recognition, especially for items with subtle differences.

17.2.3. Opportunity

- **Market Demand:** Increasing consumer demand for convenient and efficient online shopping solutions.
- **Advancements in AI:** Continuous improvements in AI can further enhance the app's performance and features.
- **Global Reach:** Expanding to international markets can significantly boost the user base and revenue potential.

17.2.4. Threats

- **Regulatory Changes:** Evolving data privacy laws could impact how data is collected and used.
- **Dependency on Internet**: Requires stable internet access, which may limit usability in areas with poor connectivity.
- **Negative User Feedback:** Initial bugs or inaccuracies could result in negative reviews, impacting user adoption.

SWOT Category	Details
Strength	- Innovative Use of AI - Multi-Platform Availability - High User Value.
Weaknesses	- High Development Cost - User Privacy Concern - Technical Limitations
Opportunities	- Market Demand - AI & ML Advancement - Global Research
Threats	- Regulatory Changes - Dependency on Internet Connectivity - Negative User Feedback

Table 7:SWOT Analysis

18. QUALITY STANDARDS FOR SAVINGBOT

18.1. Functionality

- **Accuracy:** Ensuring the image recognition algorithm accurately identifies clothing items and the web scraping accurately collects price data.
- **Completeness:** The system should cover all necessary functionalities, such as item identification, price comparison, and user interface efficiency.

18.2. Reliability

- **Availability:** The app should always be available to users, with minimal downtime.
- **Fault Tolerance:** The system should handle errors gracefully and recover quickly from failures.

18.3. Usability

- **User Interface (UI):** The interface should be intuitive and easy to navigate.
- **User Experience (UX):** The overall user experience should be smooth, with quick response times and helpful feedback.

18.4. Performance Efficiency

- **Response Time:** The system should provide quick responses to user actions, such as image uploads and price comparisons.
- **Resource Utilization:** The system should efficiently use computational resources to perform image recognition and web scraping tasks.

18.5. Security

- **Data Protection:** User data should be securely handled, ensuring privacy and compliance with relevant regulations.
- **Secure Data Scraping:** The web scraping component should respect the terms of service of the websites being scraped and avoid triggering anti-scraping measures.

18.6. Maintainability

- **Modularity:** The system should be built in a modular fashion to facilitate easy updates and maintenance.
- **Scalability:** The system should be designed to handle increased load and new features as the user base grows.

18.7. Portability

- **Cross-Platform Compatibility:** The app should be available on multiple platforms (iOS and Android) without significant changes in the codebase.
- **Adaptability:** The system should be easily adaptable to include new features and support additional product categories in the future.

19. REFERENCES

https://www.kaggle.com/datasets/zalando-research/fashionmnist
https://www.shopzilla.com/
https://www.google.com/shopping?hl=en

20. 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.