# PROJECT MANAGEMENT PLAN

# FOR THE

# FASHION STORE WEBSITE

**20/10/2016**

BUE
The British University in Egypt
الجامعة البريطانية في مصر

Team Number : 12
Team Member 1: Salma Mohamed Yassin - 133722
Team Member 2: Nouran Moataz Mosallam - 129128
Team Member 3: Sara Gamil Fahmy - 131991
Team Member 4: Amira Adly Mohamed - 131880

## TABLE OF CONTENTS

**Section** **Page**

# SECTION 1. OVERVIEW

## 1.1 PROJECT SUMMARY

### 1.1.1 Purpose, Scope, and Objectives

The purpose of this document is to serve as a guide for development of the project and making sure that all requirements are met and the produced system functions per the client's requirements. The website will be implemented in order to make the customer easily connected to the store , removing any border barriers and turning the store into an international one.

The scope of this system is a web based software serving two main users; the customer and the admin. Through its different functionalities,

The customer:

- Uses the website to view available products through different categories
- Order products online
- Track his purchases through his payment reference number
- Contact the store for customer support

The admin:

- Uses the website to add products of the store to the site
- Uses the website to remove the products of the store to the site
- Update information about the products
- Tracks the sold products to insure their availability in the inventory

**Objectives:**

- The project will increase the sales of the fashion house
- Open a new channel of connection between the fashion house and the customer
- Helps making more marketing and publicity for the brand
- Collecting data and feedback about the products and their customers (market review)
- Make the brand easy to be reached for the customers anywhere around the country
- A way of building customer trust by publishing annual achievements

### 1.1.2 Assumptions and Constraints

**Assumptions:**

- The website will need an internet connection to be reached
- The Website works on most browsers
- The user's account needs to be activated via mail or phone so that the customer would be able to purchase products

- The user's account should have a credit card number for the billing system
- Loads for a few seconds only to find searched products for the user
- The cart of products will have a maximum of 50 products per purchase
- The customer will have to confirm the purchase before billing
- The purchase will take from 10-15 days to be shipped to the customer
- The customer will receive a reference number to his bill to be able to track his purchase
- If anything went wrong the customer can always call customer service
- For the admin, the inventory levels should always be taken care of
- Editing or removing products will have to be confirmed
- Only Microsoft Services will be used

**Constraints:**

- Budget
  - The website will not exceed the estimated budget which is $55,000
- Time
  - Once the URL is purchased it only takes (5 months) for the website to be fully built
  - Images for the products on the website loads in a good amount of time for the customer not to get board
- Staff
  - A group of employees will be responsible for answering customer complaints on the website
  - Another group of staff will be responsible for maintaining the website
- Maintenance
  - The website will have to be designed such as the maintenance expenses do not cost the company a fortune

### 1.1.3 Project Deliverables

- Software program and library binaries
- Publicize plan documentation
- Source code of the implemented web application
- Source code documentation
- Fully implemented web application
- The project should be delivered within 3 months

**Software documentation:**

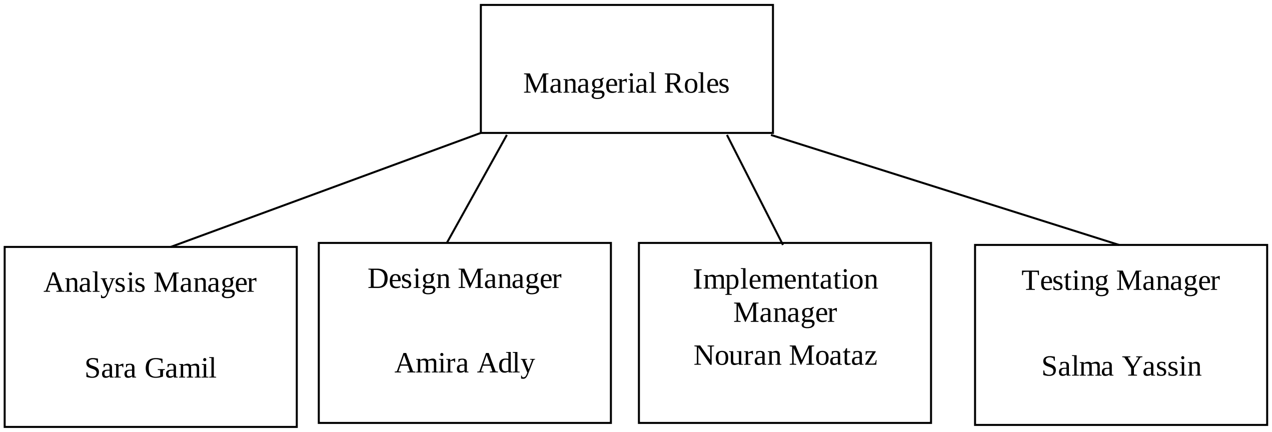
- Installation documentation
- End-user documentation
- Software maintenance documentation

**Project documentation:**

- Software Requirements Specification (SRS) documentation
- Software Design Specification (SDS) documentation
- Software Project Management Plan (SPMP) documentation
- Software Test Plan (STP) documentation
- Software Quality Assurance Plan (SQAP) documentation
- Software Configuration Management Plan (SCMP) documentation
- Software Verification and Validation Plan (SVVP) documentation
- System description documentation
- Project team organization documentation
- Business need documentation
- Cost-benefit analysis documentation

# SECTION 2. PROJECT ORGANIZATION

## 2.1 PROJECT ROLES AND RESPONSIBILITIES



**Responsibilities:**

**Analysis Manager:** His role is to make the organization aware of the challenges along the way before actually facing them in order to set a clear view of the required software project. Once the Analysis phase begins, the Analysis manager starts dividing roles on the members to make sure to gather all functional requirements and identify objectives and required deliverables

**Design Manager:** Design Managers are responsible for managing the process of producing initial low fidelity and high fidelity designs for the GUI (Graphical User Interface) and the initial database design using ERD. Then, get both designs professionally done, ready for implementing.

**Implementation Manager:** Responsible of distributing small pieces of the program on the developers, to make sure that the program is fully implemented so that he can collect all the pieces again to form a strong full developed web application meeting all the required functionalities.

**Testing Manager:** Responsible for fully testing the functionalities of the system and insure that all the expected results are being produced by the system. He also tests all the designs of the system to make sure that they are interactive user friendly designs.

# SECTION 3. MANAGEMENT PROCESS

## 3.1 START-UP .

### 3.1.1 Estimation

**Functions:**

-Track purchases through payment reference number.

-Add products of the store to the website.

**Size Estimation:**

**Track purchases through payment reference number:**

The customer will enter the purchase reference number which is in the purchases table in the website's database. A report including the item's name, number and location (if it's in the warehouse, the delivery company or on its way to the customer) will be viewed to the customer. Consequently, there will be access to two external databases, the warehouse and the delivery company. In the warehouse one table will be checked which is the dispatched items. In the delivery company's table two tables will be checked which are the items received and items dispatched. A query will be used in order to connect between them and check at which stage the item is this query will have four outputs if it is dispatched from the warehouse or not, if it is received to the delivery company or not and if it is dispatched from the delivery company or not and therefore locates the place of the item.

**External input types(EI):** 1 input

**Record Types:** Purchases table

**Data Types:** Purchase reference number

**Complexity:** Low =3

**External output types(EO):** 1 report

**Record Types:** Purchases table, warehouse's dispatched table, delivery company's received table, delivery company's dispatched table

**Data Types:** Item name, item number, location

**Complexity:** Average=5

**External inquiry types(EQ):** 1 query

**Record Types:** purchases (to get item number), dispatched(warehouse), dispatched (delivery company), received (delivery company)

**Data Types:** dispatched(warehouse), dispatched (delivery company), received (delivery company), location

**Complexity:** Average=4

**Logical internal filetypes(ILF):**1 internal database

**Record Types:** Purchases

**Data Types:** Item name, Item number

**Complexity:** Low=7

**External Interface Files(UI):** 2 databases (warehouse, delivery company)

**Record Types:** dispatched(warehouse), dispatched (delivery company), received (delivery company)

**Data Types:** location

**Complexity:** Low=5

**Total size=**(1x3) + (1x5) +(1x4) +(1x7) +(2x5) =29 FP x 60 =1740 LOC =1.74 KLOC

(MVC .NET http://www.qsm.com/resources/function-point-languages-table )

**Add products of the store to the website:**

The admin will receive a report from the warehouse of the available products getting their reference number, name and amounts, he will then add the items with their details to the to the internal database of the website as he will input the item name, amount and description.

**External input types(EI):** 3 inputs

**Record Types:** available products (website)

**Data Types:** name, amount, description

**Complexity:** Low=3

**External output types(EO):** 1 report

**Record Types:** available products(warehouse)

**Data Types:** reference number, name, amount

**Complexity:** Low=4

**External inquiry types(EQ):** none

**Record Types:** none

**Data Types:** none

**Complexity:** none

**Logical internal filetypes(ILF):** 1 internal database

**Record Types:** available products(website)

**Data Types:** name, amount, description

**Complexity:** Low=7

**External Interface Files(UI):** 1 external database(warehouse)

**Record Types:** available products(warehouse)

**Data Types:** reference number, item name, amount

**Complexity:** Low=5

**Total size=**(3x3) + (1x4)+(1x7)+(1x5)=25 FP x 60 =1500 LOC =1.5 KLOC

(MVC .NET http://www.qsm.com/resources/function-point-languages-table )

**Effort:**

**Track purchases through payment reference number:**

**Exponent Multipliers:**

**Product**

**RELY Required software reliability 1.00:** Nominal , because even though all databases should be regularly updated , there is a chance that the product is ahead with one step but the database was not yet updated by the warehouse or delivery company.

**DATA Database size 1.14:** High, information is being taken from more than one database external and internal.

**CPLX Product complexity 1.34:** Very High, this function is very complex as it will access three databases and a query in order to retrieve the final report to the customer.

**REUSE Required reusability 0.95:** Low, the function is less likely to be reused in other systems the same way because it will need a lot of modifications.

**Computer**

**TIME Execution time constraint 1.11:** High, a bit extra time might be needed in order to access all databases and execute the query in order to produce the final report.

**STOR Main storage constraint 1.00 :** Nominal, the internal database is relatively small as it will only contain the available products and purchases made.

**PVOL Platform volatility 1.30:** Very High , the database changes constantly.

**Personnel**

**ACAP Analyst capabilities 0.71:** Very High, the analyst is very efficient and cooperative.

**AEXP Application experience 0.81:** Very High, the team have worked on this type of projects many times.

**PCAP Programmer capabilities 0.76:** Very High, the team is very highly cooperative and have been working together on lots of projects before.

**PEXP Platform experience 0.91:** High,The developers have a high level of experience in the platforms they are using.

**LEXP Programming lang. experience 0.84:** Very High, programmers are specialized in MVC and have worked of many projects using it.

**PCON Personnel continuity 0.81:** Very High, the annual turnover is very low, almost no changes happen in employees

**Project**

**TOOL Use of software tools 0.78:** Very High, the latest Microsoft tools are being used which are highly integrated

**SITE Multisite development 0.93:** High, all the team members are located in the same site, but the developer might need to contact other developers who are responsible for the warehouse or delivery company .

**SCED Schedule pressure 1.14:** Low, the function is very complex and the schedule is quite compressed.

**Exponent driver ratings:**

**PREC 3.72:** Nominal, the function is a bit similar to other previously designed functions but it has some differences.

**FLEX 1.01**: Very high, the function is only developed in one way.

**RESL 1.41:** Very High, the function is known but many changes may occur due to external databases.

**TEAM 5.48:** very low, the team works together in the same room.

**PMAT 1.56:** Very High, the function is complex and many changes can occur in its development.

A=2.94

B=0.91

Size=**1.74 KLOC**

Sum of exponent driver ratings=3.72+1.01+1.41+5.48+1.56=**13.18**

Product of exponent multipliers= 1.00*1.14*1.34*0.95*1.11*1.00*1.30*0.71*0.81*0.76*0.91*0.84* 0.81*0.78*0.93*1.14 = **0.469**

Sf=B+0.01*∑ (exponent driver ratings) =0.91+0.01*13.18=**1.0418**

Effort (PM) = A (Size)^sf x (Product of exponent multipliers) = 2.94(1.74) ^1.0418*(0.469) = **2.455**

**Cost Estimation:**

Each developer takes 2000$ per project.

Computers cost 700$ per computer.

Projector price 1000$

Internet connection 100$ per month

100$ *5 months =500$

Visual Studio’s price 499$

SQL server is free for developers

Total Cost for this function=
(2000*3)+(700*3)+(1000/2)+(499/2)+(500/2)=9099.5$

**Add products of the store to the website:**

**Exponent Multipliers:**

**Product**

**RELY Required software reliability 1.00:** Nominal, because even though all databases should be regularly updated, there is a small chance that some products are available in the warehouse but their database isn’t yet updated.

**DATA Database size 1.14:** High, lots of new items are added to the database.

**CPLX Product complexity 1.00:** Nominal, this function is not very complex as it will access only two databases one to directly retrieve information from and the other is to directly add information in.

**REUSE Required reusability 1.07:** High, the function can be reused, with slight modification needed in the database connections.

**Computer**

**TIME Execution time constraint 1.00:** Nominal, the report produce only needs to be retrieved form one database and the information is directly added to the internal database so the execution time is normal.

**STOR Main storage constraint 1.17:** Very High, the internal database for the add function is very high as new products needs to be constantly added to the database.

**PVOL Platform volatility 1.30:** Very High, the database changes constantly.

**Personnel**

**ACAP Analyst capabilities 0.71:** Very High, the analyst is very efficient and cooperative.

**AEXP Application experience 0.81:** Very High, the team have worked on this type of projects many times.

**PCAP Programmer capabilities 0.76:** Very High, the team is very highly cooperative and have been working together on lots of projects before.

**PEXP Platform experience 0.91:** High, the developers have a high level of experience in the platforms they are using.

**LEXP Programming lang. experience 0.84:** Very High, programmers are specialized in MVC and have worked of many projects using it.

**PCON Personnel continuity 0.81:** Very High, the annual turnover is very low, almost no changes happen in employees.

**Project**

**TOOL Use of software tools 0.78:** Very High, the latest Microsoft tools are being used which are highly integrated

**SITE Multisite development 1.00:** Nominal, all the team members are located in the same site so few non- face to face communication is needed just some phones or e-mails but extra communication might be needed to contact developers responsible for the warehouse.

**SCED Schedule pressure 1.00:** Nominal, the function is not very complex and the schedule is not pressured.

**Exponent driver ratings:**

**PREC** 4.96: Low, the function is very similar to other previously designed functions with very slight differences.

**FLEX** 1.01: very high, the function is only developed in one way.

**RESL** 5.65: Low, the function is known and really slight changes can be made to it.

**TEAM** 5.48: very low, the team works together in the same room.

**PMAT** 6.24:  Low, the function is pretty organized with a clear path to be developed upon.

A=2.94

B=0.91

Size= 1.5 KLOC

Sum of exponent driver ratings= 4.96+1.01+5.65+5.48+6.24=**23.34**

Product of exponent multipliers= 1.00*1.14*1.00*1.07*1.00*1.17*1.30*0.71*0.81*0.76*0.91*0.84*0.81*0.78*1.00 *1.00= **0.392**

Sf=B+0.01*∑ (exponent driver ratings) = 0.91+0.01*23.34= **1.143**

Effort (PM) = A (Size)^sf x (Product of exponent multipliers) = 2.94(1.5) ^1.143*(0.392) =**1.832**

**Cost Estimation:**

Each developer takes 2000$ per project.

Computers cost 700$ per computer.

Projector price 1000$

Internet connection 100$ per month

100$ *5 months =500$

Visual Studio's price 499$

SQL server is free for developers

Total Cost for this function=

(2000*2)+(700*2)+(1000/2)+(499/2)+(500/2)=6399.5 $

**Contact customer support:**

We want to create a fashion website, so we want to help the user to contact the customer serves anytime. So, the user will open the website. To do that the user should enter his email: the password and the user name then press log in. the main page of the website will appear. The user from browse can choose "customer support". As he presses on customer support an automatic file called "user entries" will be opened to contain everything that the user will enter. When he presses on customer support, the system will connect to a file called "problems file" to extract problems, questions and answers from it. (The problems file contains all the problems that may be occur with the user, the questions and their answers). Then a helping page will appear displaying titles of problems. When the user choose his problem, the system will generate questions related to the user's problem from problems file. The client should choose the question that he wants to ask. After choosing the question, all the product that he bought will appear to allow him to mark the product that he have problem with. Then there will be a message appear trying to give him an answer to his question. If this message does not replay his question he can press leave a message. Then he can enter his message and press send. After entering his message, another message will appear telling the user one of our customer support will replay to your mail within 72 hours and displays what is his problem about?

External input 11:

11 data types, 1 record type (low complexity)

11 data types are: the ones written in read

1 record type is: user entries that will record everything the "user enters".

External output 2:

2 data type from 1 record type (low complexity)

2 data type: a message answers the user question and another message telling the user that one of

the customer support will reply within 72 hours.

1 record type: problem file that contains answers of the questions.

External query 1:

3 data type, 1 record (low complexity)

3 data types are: the problem, the question and the answer.

1 record: the "problems file" that contains all the Problem, all the questions and all the answers.

Logical internal file 1:

3 data type, 1 record type (low complexity)

3 data type: the main problem, the question and the message that the user has entered.

1 record type: "user entries" file that will record what the user enters.

External interface file: 0

| **Component category** | **Complexity of component** | | |
|---|---|---|---|
| | Low | Average | High |
| **External Input** | 3 | 4 | 6 |
| **External Output** | 4 | 5 | 7 |
| **External Queries** | 3 | 4 | 6 |
| **Internal Logical Files** | 7 | 10 | 15 |
| **User Interface** | 5 | 7 | 10 |

So we get 3*11=33, 2*4=8, 3*1=3, and 7*1=7
Size=33+8+3+7=51 FP
I assume the language will be java
So, 60*51=3060 LOC

**Make an order**

To help the user to order things easily, he should open the website. Then he press login (after entering his username and password". Then the main page will appear showing all the product and its price. The user can press on whatever product he wants, the system should extract information about this especial product from a file called "products detailed file". This file showing the price of each product, the available colors, size and the shipment. If the user want to buy any product, he can press buy then the system will allow him to choose the product information (enter the color and the size that he want). At this time there will be a file called "order file" record everything about the product that the user want. After entering the color and the size of the product, the system wants you to enter your personal information (the country, city, house number, zip code, the credit number and mobile number). Then he presses next. A file called "personal information file" will record the personal information of the user. Then the system will display a message contains the size, the color, the shipment and all your personal information). Then the user should press confirm. Finally, a message will appear telling the user "thanks for shopping with us".

External input 16:

16 data types, 2 record type (high complexity)

16 data types: all the red statements.

2 record type: "order file" that will record things about the order like size and color. And "personal information file" will record the personal information of the user.

External output 2: 2 data type, 2 record (low complexity)

2 data type: a message about your order telling you all the information about your order. And a thanking message.

2 record: "order file" and "personal information file" will record the personal information of the user.

External query 1:

4 data type, 1 record (low complexity)

4 data type: size, colors, shipment and price.

1 record: "products detailed file" that contains all information about all product and when the user

choose one product, the system displays information about this particular product.

Logical internal file 2:

10 data type, 2 record type (low complexity)

10 data type: size, colors, shipment, price, country, city, house number, zip code, credit number and mobile number

2 record type: "order file" and "personal information file" will record the personal information of the user.

External interface file: 0

| Component category | Complexity of component | | |
|---|---|---|---|
| | Low | Average | High |
| **External Input** | 3 | 4 | 6 |
| **External Output** | 4 | 5 | 7 |
| **External Queries** | 3 | 4 | 6 |
| **Internal Logical Files** | 7 | 10 | 15 |
| **User Interface** | 5 | 7 | 10 |

So we get

16*6=96, 2*4=8, 3*1=3, 7*2=14

Size= 96+8+3+14= 121 FP

I assume the language will be java

So, 60*121=7260 LOC

**Effort estimation:**

**Exponent drive rating for function one (contact customer support):**

Our website is a fashion website that help the user to express his problem. Precedentedness (PREC) is consider low as this function is somehow similar to a function in another website. The flexibility is rating high as the way to do a function is a standard way the user can get it by experience. He may find it a bit difficult at the beginning. The risk reduction is low because almost requirement are obvious. The team cohesion is very low since all the developers are seating together most of the time. Process maturity is very high as the project is highly organized and structured.

So

Summation of Exponent drive rating =

4.96+2.03+5.56+5.48+1.56=19.59

So, sf= B+0.01* Summation of Exponent drive rating

Sf=0.91+0.01*19.59=1.1059

**Exponent multiply**:

**Contact customer support:**

In our fashion website we need to build a system that enable the user to contact the customer support easily and detail his problem as he can. This system should be a very high degree of reliability to build the trust of the user. The system should be a high degree of reusability to be able to reprocess the system again in another usage. The difficulty of the system should be low to help the user to have no problem with doing all what he want on the website. In this website the Personal capability will be very low. Personal experience should high, the more the user use the website to contact the customer support, the easier to use. Facilities available and schedule pressure are both nominal.

So, product = 1.91*1.07*0.87*1.62*0.87*1*1=2.506

Effort= A (size) ^ sf *Product of exponent multipliers

= 2.94*(((51*60) ^ (1.1059))*2.506) = 51603.2188 PM

**Exponent drive rating for function two (making order):**

Our website is a fashion website that help the user to make an order easily. Precedentedness (PREC) is consider very low as this function is similar to a function in another website. The flexibility is rating high as there is only way to make the order. The risk reduction is low because almost requirement are obvious. The team cohesion is very low since all the developers are seating together most of the time. Process maturity is very high as the project is highly prepared and planed.

So

Summation of Exponent drive rating =

6.20+2.03+5.56+5.48+1.56=20.83

So, sf= B+0.01* Summation of Exponent drive rating

Sf=0.91+0.01*20.83=1.1183

**Make an order:**

In our fashion website that has "average" novelty. We need to build a system that enable the user to make the order easily. This system should be a very high degree of reliability to protect the personal information of the user. The system should be a high degree of reusability to be able to reprocess the system again in another website. The difficulty of the system should be low to help the user to have no problem with buying all what he want from the website. In this website the Personal capability will be very low because the disabled users will have a great difficulty to use this system as they cannot even see the product. Personal experience should very high, the more the user buy things, the more he saves time and efforts. Facilities available will be high as the user have many facilities to serve the website. Schedule pressure is nominal.

So, product = 1.91*1.07*0.87*1.62*0.74*0.87*1=1.85439

Effort= A (size) ^ sf *Product of exponent multipliers

=2.94*(((7260) ^ (1.1183))* 1.85439) = 113301.0227 PM

***Cost of the first function:***

Cost: we need about 4000 pound to install java and give analysts their salaries.

Days: 3 days

***Cost of the first function:***

Cost: we need about 5000 pound to install java and give developers their salaries.

Days: 3 days

Functions:

1- Use the website to remove a product from the store
 In order the adman could make such an important costly decision he need a function that generates a report about the amount of seals for a specific product in a specific period to make the decision of removing or keeping it. the adman need a high amount of reliability, flexibility and a will specified requirement illustration due to the importance of the decision in addition to that he is not familiar with the platform as it is a new system for the fashion house

Size estimation:

**SOLUTION:**

1. Count # of components per each category “EI, EO, EQ, ILF, EIF”

External input types ------ duration, product name, 2 records

External output types ---- the report, that is 3 record

a. Product information
b. Number of soled items
c. Calculation of the price

External inquiry types ----- profit, I data type, 2 records

Logical internal file types --------- none

External interface file types------------ none

2. Identify the complexity of each of the components

External input types ------ Product name and duration, which is 2 data type and

2 record types (Low complexity)

External output types ------ the report, which is 1 data type and is retrieved from 3 record types (Low complexity)

External inquiry types ------ profit, which is 1 data type and it is calculated from 2 records (product, soled items) (Low complexity)

| **Component category** | **Complexity of component** | | |
|---|---|---|---|
| | Low | Average | High |
| **External Input** | 3 | 4 | 6 |
| **External Output** | 4 | 5 | 7 |
| **External Queries** | 3 | 4 | 6 |
| **Internal Logical Files** | 7 | 10 | 15 |
| **User Interface** | 5 | 7 | 10 |

So we get: 3*2 + 4*3+ 3*1+ = 33 FP

So the number of code lines is =60*33=1980=1.98 KLOC

Effort estimation:

| Driver | Very low | Low | Nominal | High | Very High | Extra High |
|---|---|---|---|---|---|---|
| **PREC** | | | | | 1.24 | |
| **FLEX** | | | | 2.03 | | |
| **RESL** | 7.07 | | | | | |
| **TEAM** | | | | | 1.10 | |
| **PMAT** | | | | | 1.56 | |

**PREC:** it is totally different there wasn't any previous systems for the fashion house

**FLEX:** the function is highly fixable the result of removing can be reached from different ways the report can be made manually

**RESL**: the requirement are very clear and straight forward

**TEAM:** team member are in the same country

**PMAT:** the team is very organized and controlled by a strict roles

Scale factor: 1.24+2.03+7.07+1.10+1.56= 0.91+0.01*16.28=1.04

| Effort Modifier | Rating | Multiplier |
|---|---|---|
| **Product reliability & complexity** | V. High | 1.91 |
| **Required reusability** | V. High | 1.15 |
| **Platform difficulty** | Low | - |
| **Personnel capability** | V. High | 1.62 |
| **Personnel experience** | Nominal | 1 |
| **Facilities available** | High | .87 |
| **Schedule pressure** | Nominal | 1 |

Product= 1.91*1.15*1.62*1*.87*1= 3.0957

Effort (PM) = A (Size)$^{sf}$ x (Product of exponent multipliers)

= 2.94(1.98) ^ 1.04 *(3.0957) = 18.5198

## Function 2:

2- The admen are required to deliver a report about the customer feedback and satisfaction about some specific products to take the decision of refactoring them or not also to make a market study about the rating age that visit the website.
He needs a reliable function to produce a report about the visitors age rate of the website (accounts database file) and their feedback in addition to the amount of seals for a specific products.
The system is a new system that applied for the first time in the fashion house

## Size estimation:

**SOLUTION:**

3. Count # of components per each category “EI, EO, EQ, ILF, EIF”:

External input types: product name, product code, 1 record.

External output types: name, code, satisfaction rate, age rate, 4 records.

External inquiry types: satisfaction rate, age rate, 2 records

Logical internal file types: feedback, age, 2 records

External interface file types: none

The complexity of each of the components:

1- External input types: name , code which is 2 data type, 1 record (Low complexity)
2- External output types: 1 report which has name, code, satisfaction rate, age rate, 1 datatypes 4 records (average complexity)
3- External inquiry types: satisfaction rate, age rate, 1 datatype, 2 records (Low complexity)
4- Logical internal file types: feedback, age, 2 datatypes, 2 records (Low

complexity)

| Component category | Complexity of component | | |
|---|---|---|---|
| | Low | Average | High |
| **External Input** | 3 | 4 | 6 |
| **External Output** | 4 | 5 | 7 |
| **External Queries** | 3 | 4 | 6 |
| **Internal Logical Files** | 7 | 10 | 15 |
| **User Interface** | 5 | 7 | 10 |

So we get: 2*3+ 5*1 +3*2 +7*2=70 FP

So the number of code lines is =60*70=4200 =4.2KLOC

Effort estimation:

| Driver | Very low | Low | Nominal | High | Very High | Extra High |
|---|---|---|---|---|---|---|
| **PREC** | | | | | 1.24 | |
| **FLEX** | | | 3.04 | | | |
| **RESL** | | | 4.24 | | | |
| **TEAM** | | | | | 1.10 | |
| **PMAT** | | | | | 1.56 | |

**PREC:** it is totally different there wasn't any previous systems for any fashion house

**FLEX:** the function is fixable the can be reached from other way but it would be slow and need effort

**RESL**: the requirement are clear

**TEAM:** team member are in the same country

**PMAT:** the team is very organized and controlled by a strict roles

Scale factor: 1.24+3.04+1.10+4.24+1.56 = 0.91+0.01* 14.64 =1.0218

| Effort Modifier | Rating | Multiplier |
|---|---|---|
| **Product reliability & complexity** | V. High | 1.91 |
| **Required reusability** | V. High | 1.15 |
| **Platform difficulty** | Nominal | 1.00 |
| **Personnel capability** | Nominal | 1.00 |
| **Personnel experience** | Nominal | 1 |
| **Facilities available** | High | .87 |
| **Schedule pressure** | Nominal | 1 |

Product =1.91*1.15*1*1*1*.87*1=1.91

Effort (PM) = A (Size)$^{sf}$ x (Product of exponent multipliers)

= 2.94(4.2) ^ 1.0218*(1.91) = 24.33418

**Functions:**

1. Uses the website to view available products through different categories
2. Tracks the sold products to insure their availability in the inventory

**Size Estimation:**

1. **Uses the website to view available products through different categories:**

In Fashion Store website the customer can enter which category he/she wants to shop or browse from then all of the items will be displayed in front of the customer with their names, descriptions, prices, sizes, colors, details and whether they are in stock or not so that the customer can browse through them freely with everything clearly displayed for him/her.

**External input types (EI):** 1 input (Category)

**Record Types:** 1 Record type – Category table

**Data Types:** 1 Data Types – Category Name

**Complexity:** Low = 3

**External output types (EO):** 1 Report

**Record Types:** 2 Record Type – Products table, Inventory table

**Data Types:** 6 Data Types – Product Name, Product Description, Product Price, Product Color, Product Details, In-Stock

**Complexity:** Average = 5

**External inquiry types (EQ):** None

**Logical internal file types (ILF):** 1 internal database

**Record Types:** 1 Record Type – Products table

**Data Types:** 5 Data Types – Product Name, Product Description, Product Price, Product Color, Product Details

**Complexity:** Low = 7

**External Interface Files (UI):** 1 database (Inventory)

**Record Types:** 1 Record Type – Inventory table

**Data Types:** 1 Data Type – In-Stock

**Complexity:** Low = 5

**Total size =** (3*1) + (5*1) + (7*1) + (5*1) = 20 FP x 60 = 1200 LOC = 1.2 KLOC

(MVC .NET http://www.qsm.com/resources/function-point-languages-table)

### 2. Tracks the sold products to insure their availability in the inventory:

One of the main admin functions of the Fashion website is to track sold products of the site to insure their availability in the inventory, a program needed to extract the number of sold products from the sold items file and the details for each item and the remaining number of that item in the inventory. The program will produce a record showing for each item the number sold and the number which is still in the inventory.

**External input types (EI):** None

**External output types (EO):** 1 report

**Record Types:** 3 Record Types – Products table, Sold Products table, In-stock table

**Data Types:** 4 Data Types – Product Name, Product ID, Number Sold, Number In-Stock

**Complexity:** Low = 4

**External inquiry types (EQ):** None

**Logical internal file types (ILF):** 2 internal databases

**Record Types:** 2 Record Types – Products table, Sold Products table

**Data Types:** 3 Data Types – Product ID, Product Name, Number Sold

**Complexity:** Low = 7

**External Interface Files (UI):** 1 database (Inventory)

**Record Types:** 1 Record Type – Inventory table

**Data Types:** 1 Data Type – In-Stock

**Complexity:** Low = 5

**Total size =** (4*1) + (7*2) + (5*1) = 23 FP x 60 = 1380 LOC = 1.38 KLOC

(MVC .NET http://www.qsm.com/resources/function-point-languages-table)

**Effort Estimation:**

1. **Uses the website to view available products through different categories:**

**Exponent driver rating:**

**Precedentedness (PREC):** Low: 4.96, Most of the websites which sells products have this have the customers view their products so this function is similar to other previously designed ones.

**Development Flexibility (FLEX):** Low: 4.05, it is very flexible there are a lot of ways that the products could be displayed for the user in

**Architecture / Risk reduction (RESL):** Nominal: 4.24, The function is clear but a few changes can be made to the function so that it can look new in the site

**Team cohesion (TEAM):** Very Low: 5.48, The team members are very close to each other, they works together on a lot of projects

**Process maturity (PMAT):** Extra High: 0.00, The project is very organized and structured, nothing can go wrong hopefully

**Exponent Multiplier:**

**Product:**

**RELY:** High: 1.10, The website can always show new version clothes, ns, The database of the products of the Fashion store is constantly changes and there is a small change that a new product will not show in a specified category

**DATA:** Very High: 1.28, The database size for this very big due to the constant added products of the store to the site

**DOCU:** High: 1.11, The function matches the life cycle needs as it shows the customer the products in the categories of the site **

**CPLX:** Very Low: 0.73, The function complexity is very low it is just displays the products for the customer to browse from

**REUSE:** High: 1.07, This function can be reused again in many sites that sells products online because it shows the items of the products the site have to offer to the customer

**Computer:**

**TIME:** Nominal: 1.00, The time is very precise in this project because it takes a lot of time to upload new products to the site

**STOR:** High: 1.05, A big storage is needed so that all the products of the site can be placed on the website

**PVOL:** Nominal: 1.00, it is not very volatile because it contains lots of pictures for the displayed products

**Personnel:**

**ACAP:** Very High: 0.71, The Analysts are very cooperative and efficient

**AEXP:** Very High: 0.81, The team working on the site is very skilled in testing applications similar to this one so they will be very precise when testing this one

**PCAP:** Very High: 0.71, The team working on the site will be interviewed to insure that their capabilities meets the project's requirements

**PEXP:** Very High: 0.85, The team working on the project must be familiar with the platform while testing the function

**LEXP:** Very High: 0.84, The team working on the project will be interviewed to insure that they are very skilled with the computer language needed to test the functions

**PCON:** Nominal: 1.00, The Website needs to always be updated with new the store's new products but can also displays older ones

**Project:**

**TOOL:** High: 0.90, The website will use a lot of software so this function may use these software used in the website

**SITE:** Very High: 0.86, The team can handle working on multiple sites at the same time

**SCED:** Very High: 1.00, The team working on the project will divide their time perfectly

A = 2.94

B = 0.91

Size = 1.2 KLOC

Sum of exponent driver rating = 4.96 + 4.05 + 4.24+5.48+0.0 = 18.73

Product of exponent driver rating = 1.10 * 1.28 * 1.11 * 0.73 * 1.07 *1.00 * 1.05 * 1.00 * 0.71 * 0.81 * 0.71 * 0.85 * 0.84 * 1.00 * 0.90 * 0.86 * 1.00 = 0.2892

Sf = B + 0.01* Σ(exponent driver ratings) = 0.91 + (0.01 * 18.73) = 1.0973

Effort (PM) = A(Size)^sf x (Product of exponent multipliers) = 2.94(1.2) ^1.0973*(0.2892) = **1.0385**

2. **Tracks the sold products to insure their availability in the inventory:**

**Exponent driver rating:**

**Precedentedness (PREC):** Low: 4.96, Most of the websites which sells products tracks their sold to insure their availability in their inventory

**Development Flexibility (FLEX):** Very High: 1.01, The function to check the inventory is only developed in one or two ways

**Architecture / Risk reduction (RESL):** Low: 5.65, The function is certain and very few changes can be made to the function

**Team cohesion (TEAM):** Very Low: 5.48, The team members are very close to each other, they works together on a lot of projects

**Process maturity (PMAT):** Extra High: 0.00, The project is very organized and structured, nothing can go wrong hopefully

**Exponent Multiplier:**

**Product:**

**RELY: Very** High: 1.26, The Function is very reliable it doesn't need to change

**DATA:** Very High: 1.28, The Database is big due to the number of products in the inventory, and the number of products sold

**CPLX:** Low: 0.73, The function complexity is low because it only tracks the sold products to insure their availability in the inventory

**REUSE:** Nominal: 1.00, This function is not very reusable in many sites

**Computer:**

**TIME:** Nominal: 1.00, The time is very precise in this project because it takes a lot of time to upload new products to the site

**STOR:** High: 1.05, A big storage is the inventory

**PVOL:** Nominal: 1.00, it is not very volatile because it contains lots of pictures for the displayed products

**Personnel:**

**ACAP:** Very High: 0.71, The Analysts are very cooperative and efficient

**AEXP:** Very High: 0.81, The team working on the site is very skilled in testing applications similar to this one so they will be very precise when testing this one

**PCAP:** Very High: 0.71, The team working on the site will be interviewed to insure that their capabilities meets the project's requirements

**PEXP:** Very High: 0.85, The team working on the project must be familiar with the platform while testing the function

**LEXP:** Very High: 0.84, The team working on the project will be interviewed to insure that they are very skilled with the computer language needed to test the functions

**PCON:** Nominal: 1.00, The Website needs to always be updated with new the store's new products but can also displays older ones

**Project:**

**TOOL:** High: 0.90, The website will use a lot of software so this function may also use these software used in the website

**SITE:** Very High: 0.86, The team can handle working on multiple sites at the same time

**SCED:** Very High: 1.00, The team working on the project will divide their time perfectly

A = 2.94

B = 0.91

Size = 1.2 KLOC

Sum of exponent driver rating = 4.96 + 1.01 +5.65+5.48+0.0 = 17.1

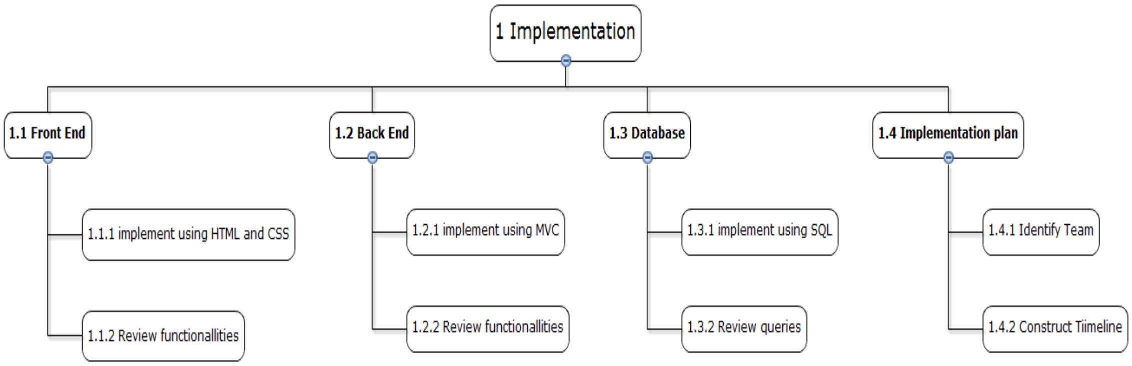
Product of exponent driver rating = 1.10 * 1.28 * 1.11 * 0.73 * 1.07 * 1.05 * 1.00 * 0.71 * 0.81 * 0.71 * 0.85 * 0.84 * 1.00 * 0.90 * 0.86 * 1.00 = 0.2892

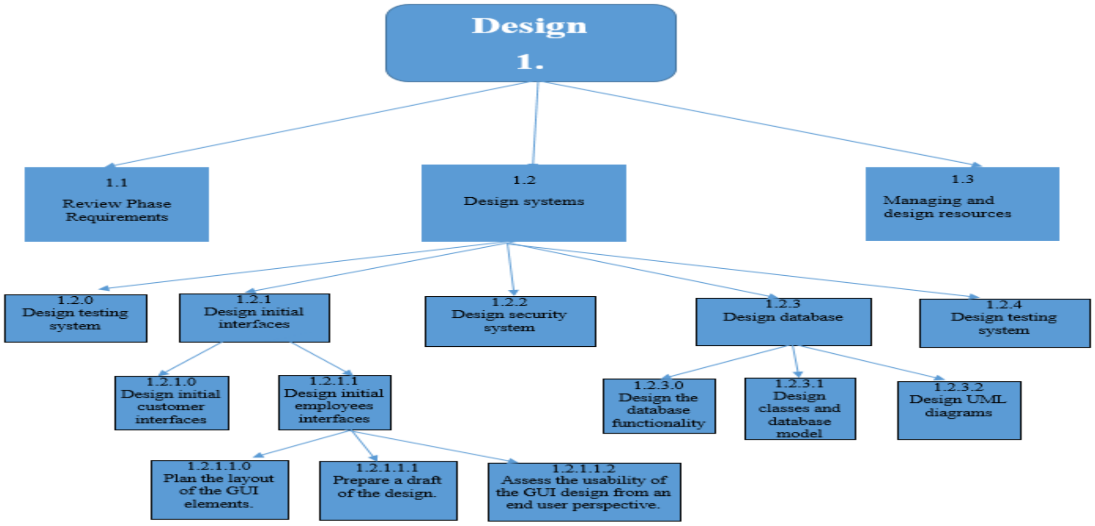
Sf = B + 0.01* Σ(exponent driver ratings) = 0.91 + (0.01 * 17.1) = 1.0973

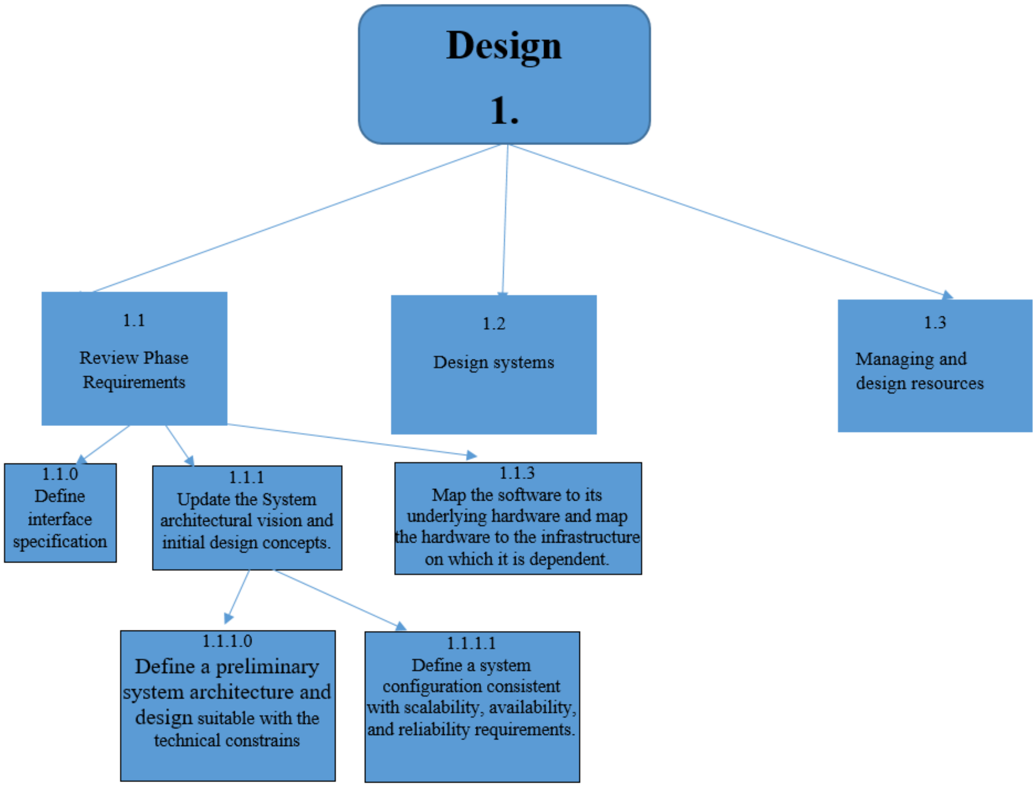
Effort (PM) = A(Size)^sf x (Product of exponent multipliers) = 2.94(1.2) ^1.0973*(0.2892) = **1.0385**

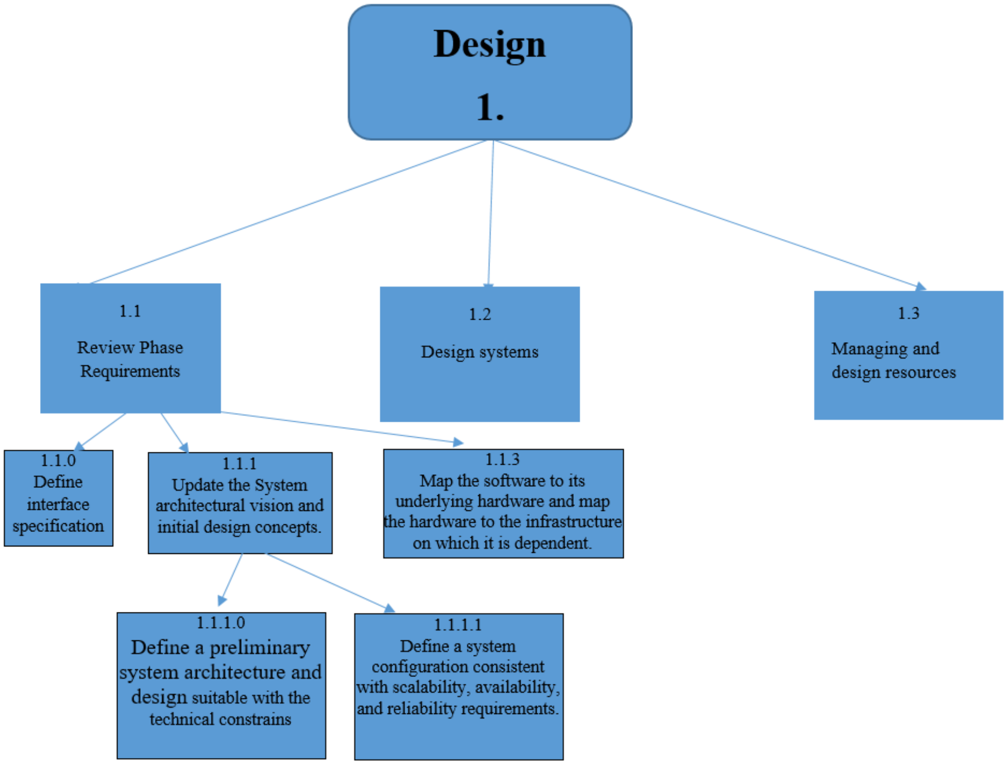
## 3.2 WORK PLANNING

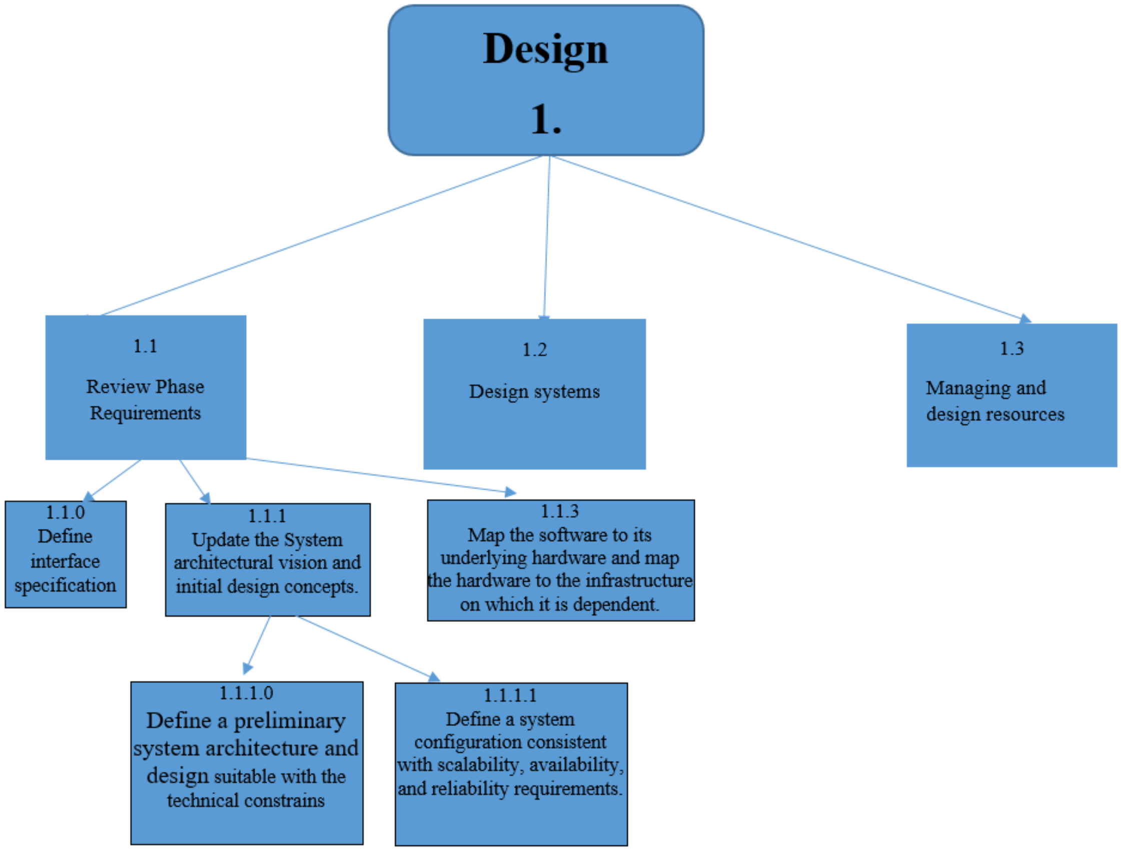
### 3.2.1 Work Activities

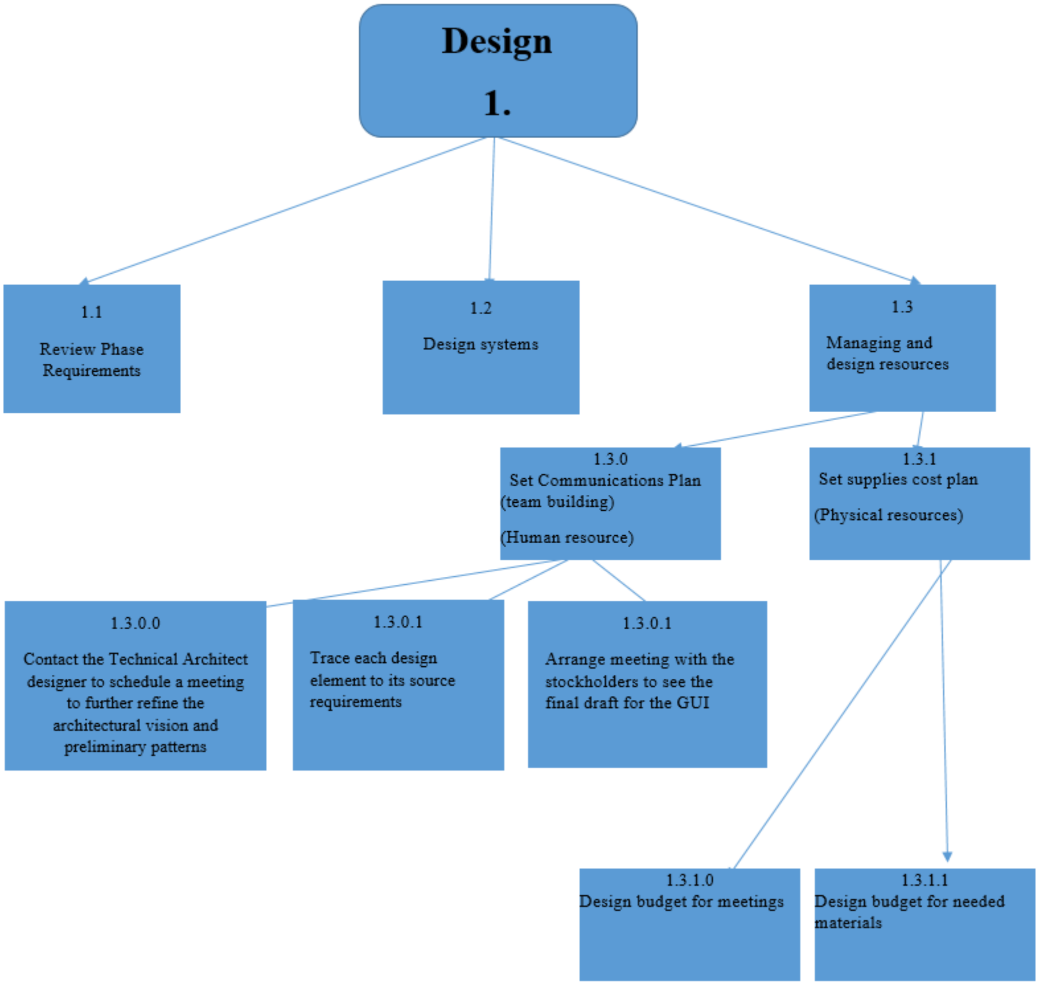

1 Implementation
1.1 Front End
1.1.1 implement using HTML and CSS
1.1.2 Review functionallities
1.2 Back End
1.2.1 implement using MVC
1.2.2 Review functionallities
1.3 Database
1.3.1 implement using SQL
1.3.2 Review queries
1.4 Implementation plan
1.4.1 Identify Team
1.4.2 Construct Tiimeline

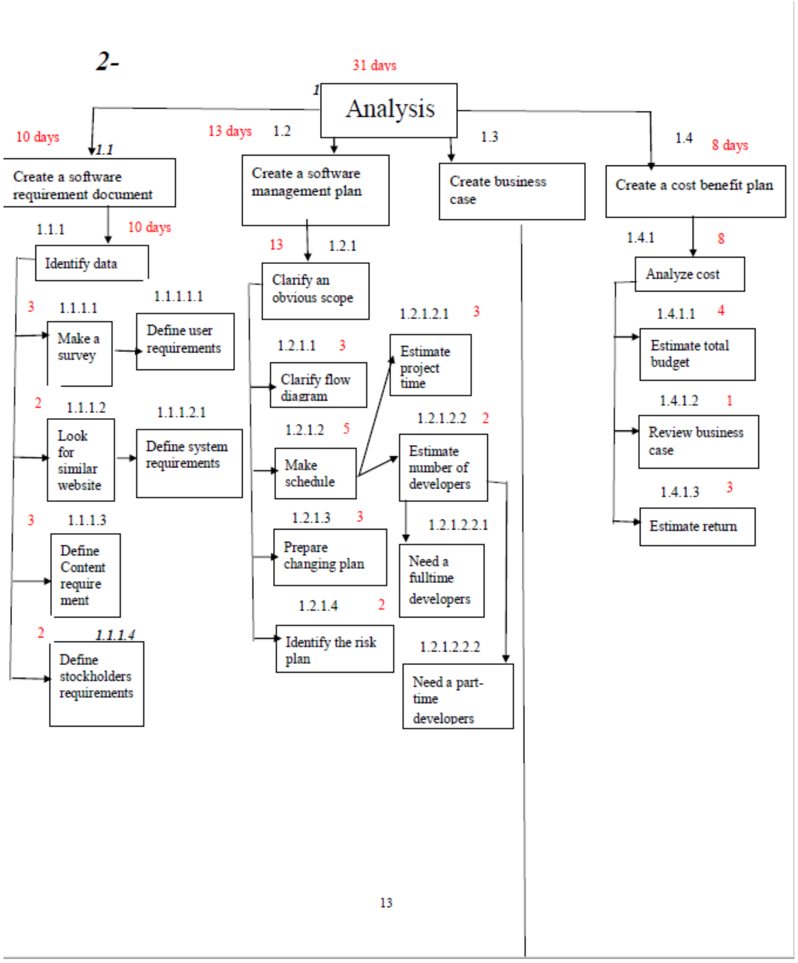

Design
1.
1.1
Review Phase Requirements
1.2
Design systems
1.3
Managing and design resources
1.2.0
Design testing system
1.2.1
Design initial interfaces
1.2.2
Design security system
1.2.3
Design database
1.2.4
Design testing system
1.2.1.0
Design initial customer interfaces
1.2.1.1
Design initial employees interfaces
1.2.3.0
Design the database functionality
1.2.3.1
Design classes and database model
1.2.3.2
Design UML diagrams
1.2.1.1.0
Plan the layout of the GUI elements.
1.2.1.1.1
Prepare a draft of the design.
1.2.1.1.2
Assess the usability of the GUI design from an end user perspective.

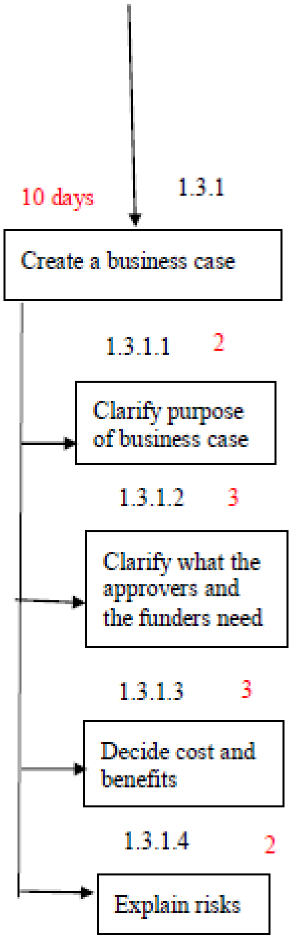

Design
1.
1.1
Review Phase Requirements
1.2
Design systems
1.3
Managing and design resources
1.1.0
Define interface specification
1.1.1
Update the System architectural vision and initial design concepts.
1.1.3
Map the software to its underlying hardware and map the hardware to the infrastructure on which it is dependent.
1.1.1.0
Define a preliminary system architecture and design suitable with the technical constrains
1.1.1.1
Define a system configuration consistent with scalability, availability, and reliability requirements.

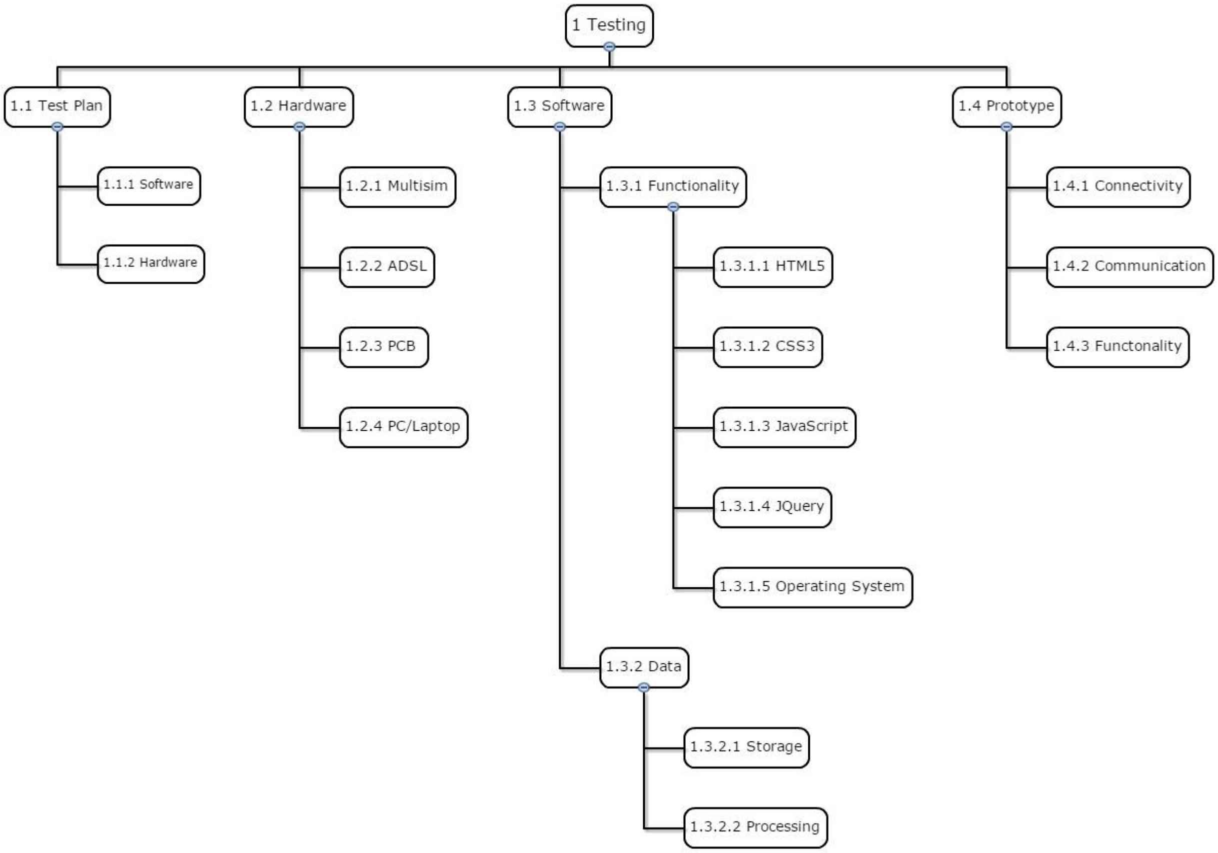

Design
1.
1.1
Review Phase Requirements
1.2
Design systems
1.3
Managing and design resources
1.1.0
Define interface specification
1.1.1
Update the System architectural vision and initial design concepts.
1.1.3
Map the software to its underlying hardware and map the hardware to the infrastructure on which it is dependent.
1.1.1.0
Define a preliminary system architecture and design suitable with the technical constrains
1.1.1.1
Define a system configuration consistent with scalability, availability, and reliability requirements.


Design
1.
1.1
Review Phase Requirements
1.2
Design systems
1.3
Managing and design resources
1.1.0
Define interface specification
1.1.1
Update the System architectural vision and initial design concepts.
1.1.3
Map the software to its underlying hardware and map the hardware to the infrastructure on which it is dependent.
1.1.1.0
Define a preliminary system architecture and design suitable with the technical constrains
1.1.1.1
Define a system configuration consistent with scalability, availability, and reliability requirements.


Design
1.
1.1
Review Phase Requirements
1.2
Design systems
1.3
Managing and design resources
1.3.0
Set Communications Plan (team building)
(Human resource)
1.3.1
Set supplies cost plan
(Physical resources)
1.3.0.0
Contact the Technical Architect designer to schedule a meeting to further refine the architectural vision and preliminary patterns
1.3.0.1
Trace each design element to its source requirements
1.3.0.1
Arrange meeting with the stockholders to see the final draft for the GUI
1.3.1.0
Design budget for meetings
1.3.1.1
Design budget for needed materials


2-
31 days
1
Analysis
10 days
1.1
Create a software requirement document
13 days
1.2
Create a software management plan
1.3
Create business case
1.4
8 days
Create a cost benefit plan
1.1.1
10 days
Identify data
3
1.1.1.1
Make a survey
1.1.1.1.1
Define user requirements
2
1.1.1.2
Look for similar website
1.1.1.2.1
Define system requirements
3
1.1.1.3
Define Content require ment
2
1.1.1.4
Define stockholders requirements
13
1.2.1
Clarify an obvious scope
1.2.1.1
3
Clarify flow diagram
1.2.1.2
5
Make schedule
1.2.1.3
3
Prepare changing plan
1.2.1.4
2
Identify the risk plan
1.2.1.2.1
3
Estimate project time
1.2.1.2.2
2
Estimate number of developers
1.2.1.2.2.1
Need a fulltime developers
1.2.1.2.2.2
Need a part-time developers
1.4.1
8
Analyze cost
1.4.1.1
4
Estimate total budget
1.4.1.2
1
Review business case
1.4.1.3
3
Estimate return
13


10 days
1.3.1
Create a business case
1.3.1.1
2
Clarify purpose of business case
1.3.1.2
3
Clarify what the approvers and the funders need
1.3.1.3
3
Decide cost and benefits
1.3.1.4
2
Explain risks



### 3.2.2 Resource Allocation

- 5 developers are need, two for the Add products of the store to the website function, and three for the Track purchases through payment reference number function as it is more complex.
- 5 computers are needed one for each developer.
- 1 meeting room with a projector is needed and shared between them.
- Visual Studio for both functions
- Microsoft SQL server for both functions
- Internet connection shared between them
- Resources needed for the first function:
- Human Resources: 2 analysts work part-time. They should have experience of doing “contact customer support” function in many websites. They also should have the ability to learn new skills easily.
- Material Resources: we need 2 computers with java installed in them and they should be experts on java.
- Cost: we need about 4000 pound to install java and give developers their salaries.

- Days: 3 days
- Resources needed for the second function:
- Human Resources: 2 analysts work fulltime and they should be experts on java. They should have experience of doing “make order” function in many websites. They also should have the ability to learn new skills easily.
- Material Resources: we need 2 computers with java installed in them.
- Cost: we need about 5000 pound to install java and give developers their salaries.
- Days: 3 days
- Hardware needed: Multisim, ADSL, PCB, and a PC or a Laptop
- Software needed: HTML5, CSS3, JavaScript, JQuery, and an Operating System
- 4 Quality assurance are needed to test out those two main functions of the system; To display the site’s products through different categories of the site and for the admin to track the sold items to insure their availability in the inventory
- A meeting place for the testing team to meet and work together

## Resources:

1- Human resources : to accomplish those 2 functions we need 5 members of the time to design and implement all required databases and files and writing code
2- Materials : 5 laptops the cost of every one is 15.000 $
3- 2 months
4- 8 meeting in the specified duration each meeting cost 100 $ per person

| Type | Number of items | Cost per item | Total cost |
|---|---|---|---|
| Laptop | 5 | 15.000 $ | 75.000$ |
| Meetings | 8 | 100 $ | 800 $ |
| Row materials | | | 400$ |

Total estimated cost =1275.000 $

# SECTION 4. TECHNICAL PROCESS

## 4.1 METHODS, TOOLS AND TECHNIQUES

**Project Management Method (SPM)**:

The Step-Wise approach will be applied throughout the project. First, the project will be selected this step is called step 0 because it is outside the main planning process. Second, the project scope and objectives will be identified in parallel with the identification of the project's infrastructures. Afterwards, we will start analysing the project characteristics, by taking in consideration user requirements concerning implementation and reviewing overall resource estimates. In the next step, the products and activities will be identified by describing the project products including quality criteria; we get back to this step every once in a while for review. Afterwards, efforts will be estimated for every activity to create controllable activities. Then, identifying activity risks for every activity estimated in the previous step to achieve planned risk reduction. Next, we will allocate resources by revising plans to take account of resource constraints. Before reaching the execution plan we will review and publicize the plan in order to review quality aspects of the plan and documenting them. Finally, we reach the execution plan and if any issues were found in the lower level details we will get back to estimating efforts for every activity or back to reviewing them with the products and activities identification.

**Software Development Method (SD):**

The used software development methodology will be the agile model where small incremental releases happen in rapid cycles, as it is the most suitable methodology to our web application due to its rapidness. The web application is small with clear requirements, but needs constant feedback and testing from the users in our case either the admin or the customer , the phases will be tested and updated every once in a while to make sure that the final product is exactly what is required by the customer and that it meets all functional requirements. The project will be generally analysed; gathering most of the functional requirements, the design phase will start. The design phase consists of two levels the system design and project design. The system design phase will start by designing a system including the database and the main basic functionalities for both users, when the system design is complete, the project design phase will be developed creating an initial GUI (Graphic User Interface) for both users to surf through the application, it needs to as accessible as possible so that all users can use the system with ease. The implementation phase, developers will start developing the system designed in the previous phases while working on developing the system a lot of technicalities can be found in the designs previously made therefore the previously made designs will be updated to meet the new specifications and design until we reach the final stage of this incremental cycle. Afterwards, the system would be fully implemented to make sure that all the requirements were met in order to be delivered to the customer.

**Tools and techniques:**

- Enterprise Architect → for Analysis phase.
- Visual Studio → for Design and Implementation phases; as we will be using HTML5, CSS3 for designing the frontend as they are the most recent versions of HTML and CSS with the greatest amount of functionalities. JavaScript and JQuery will be used to make the website more interactive and user friendly. For the backend development we will be using MVC since all the developers have a strong background in it and we will only be using Microsoft services.
- Microsoft SQL Server → for Design and Implementation phases, as SQL will be used for designing and implementing the database layer of the web application.
- Visual Studio Test professional → for testing phase.

## 6.2 PROJECT INFRASTRUCTURE

- The Laptop or the PC the user is using should be connected with a good internet connection for the site to work perfectly
- It is preferred that the internet connection's speed to exceed 1mbps
- The website works with HTML5, CSS3, JavaScript and JQuery which works best with the following internet browsers:
    - Firefox 48.0
    - Google Chrome 52.0
    - Safari 9.1
    - Edge 14