



## CPIT 456- Project (15 Points)

**Deadline:** Sunday, 24 April 2022

(23 Ramada 2022)

---

# (Hospital Leave System)

---

Student Name	ID
1. Elaf Yousef Aloufi	1911265
2. Manar Mutlaq Altairy	1906775
3. Randa Ibrahim Zhrani	1907096
4. Ghaliah Banounah	1906829
Section: <b>DAR</b>	Group ID: <b>2</b>

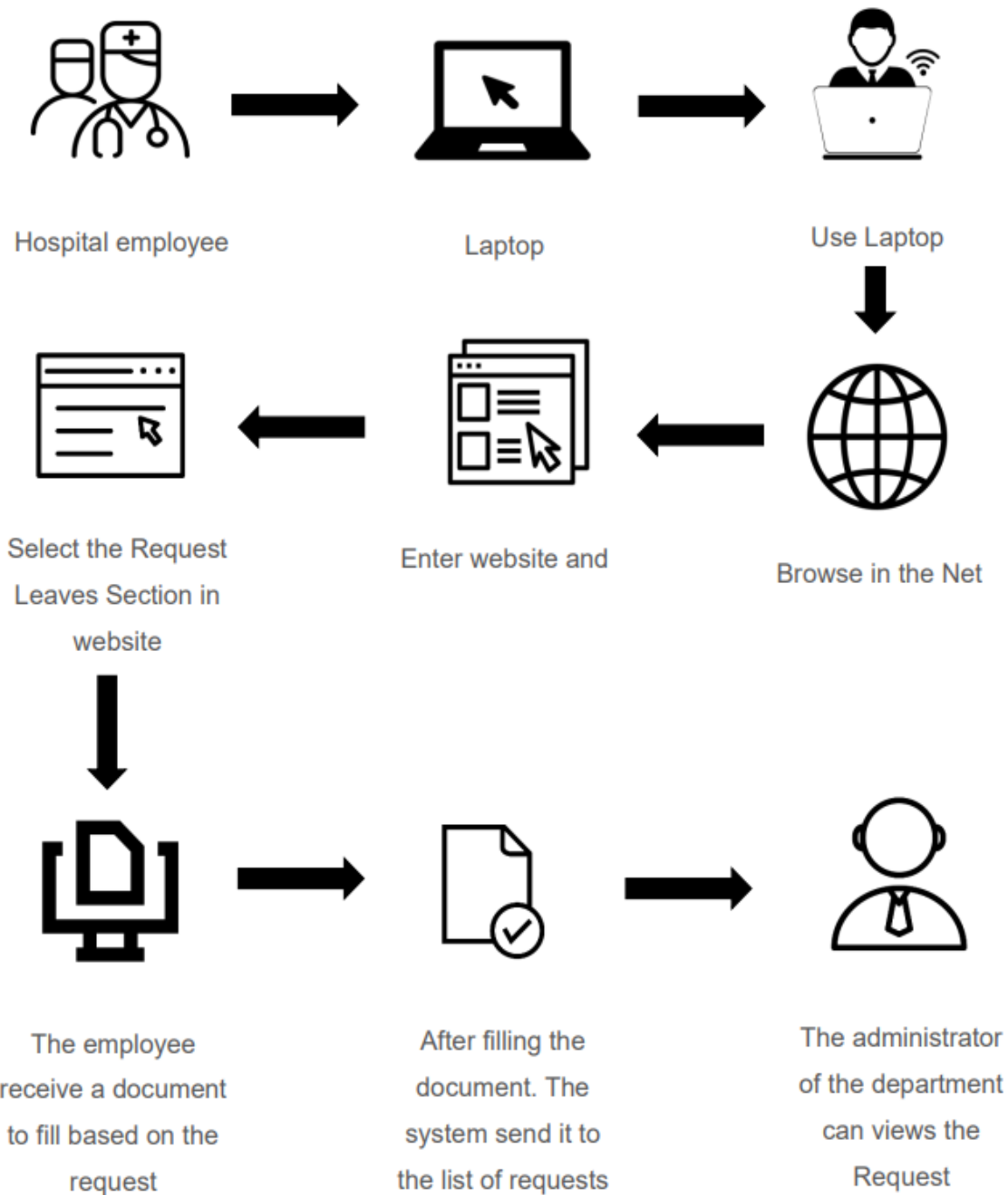
## **1. Executive Summary**

One of the systems the hospital uses has a lack of booking information for employees from different departments in the hospital. They use a manual leave request system to fill out forms. The main problem is about employee vacations, their types, number of days, and how to calculate what is taken from them and what remains of them for each employee in the department. Therefore, we decided to create a website that contains the leave system for all employees, their types, the number of days, and calculations related to employee vacations.

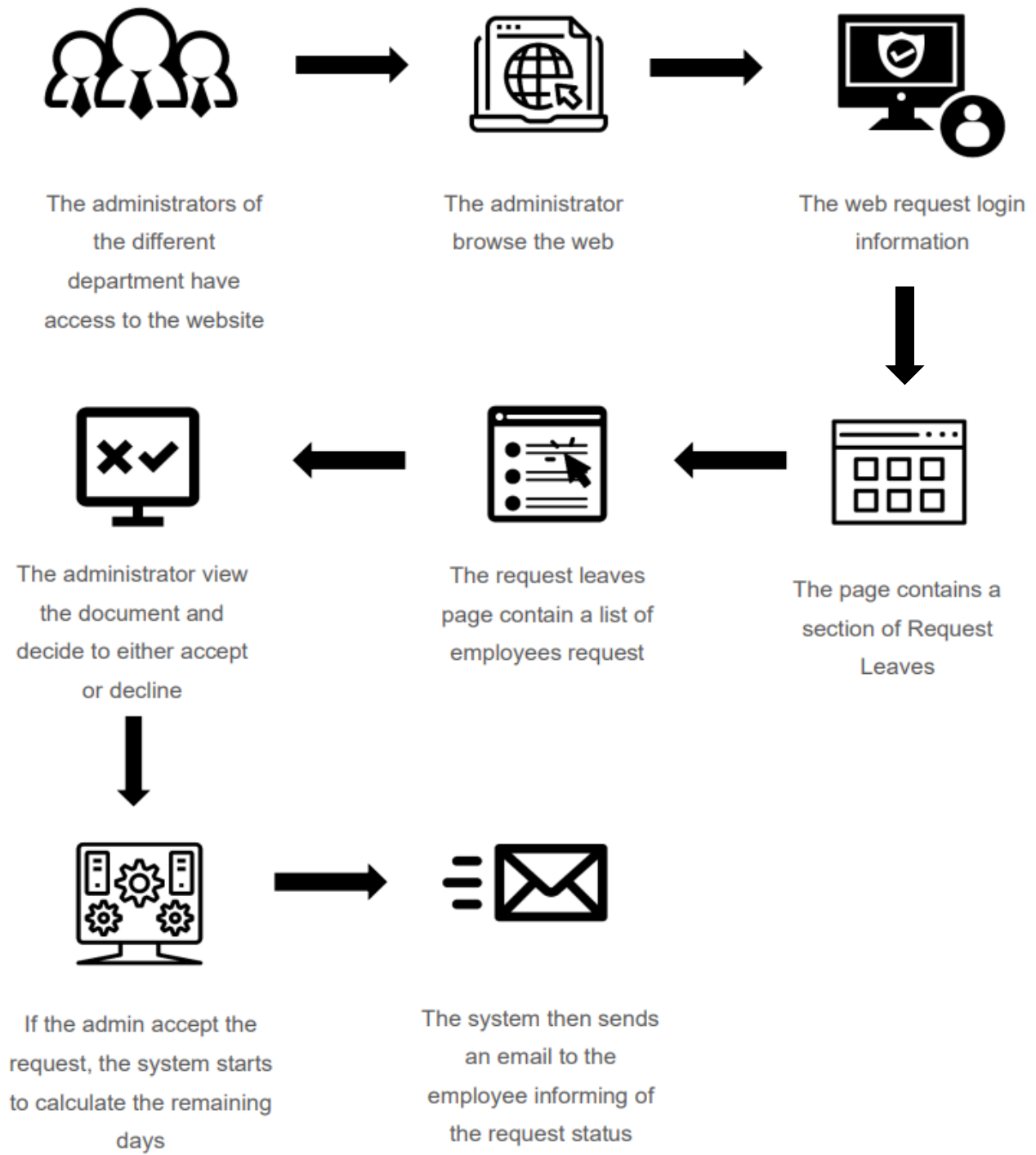
The website collects all employees' leave systems from different departments into one. In addition, each department has an administrator who has superior access to the website to confirm the request or decline it.

The website has multiple features that allow the admin to access employees' requests and vacation records with remaining days plus status. It also has features for employees to make leave requests and view their vacation records, remaining days, and request status.

## Employees View

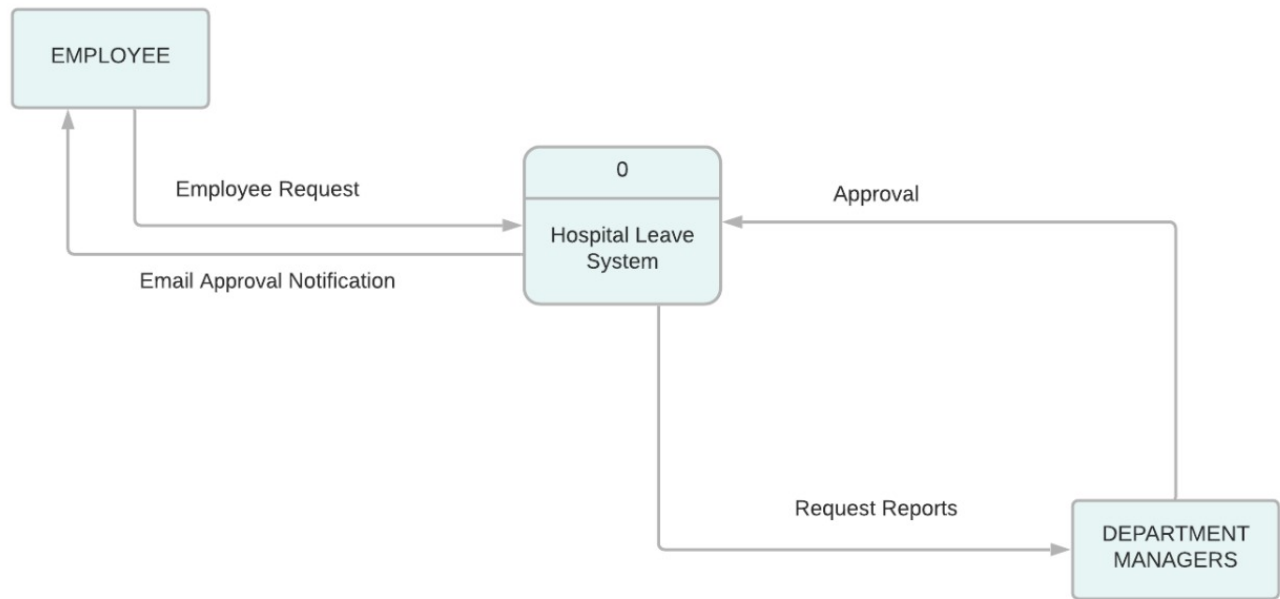


## Department Manager's View

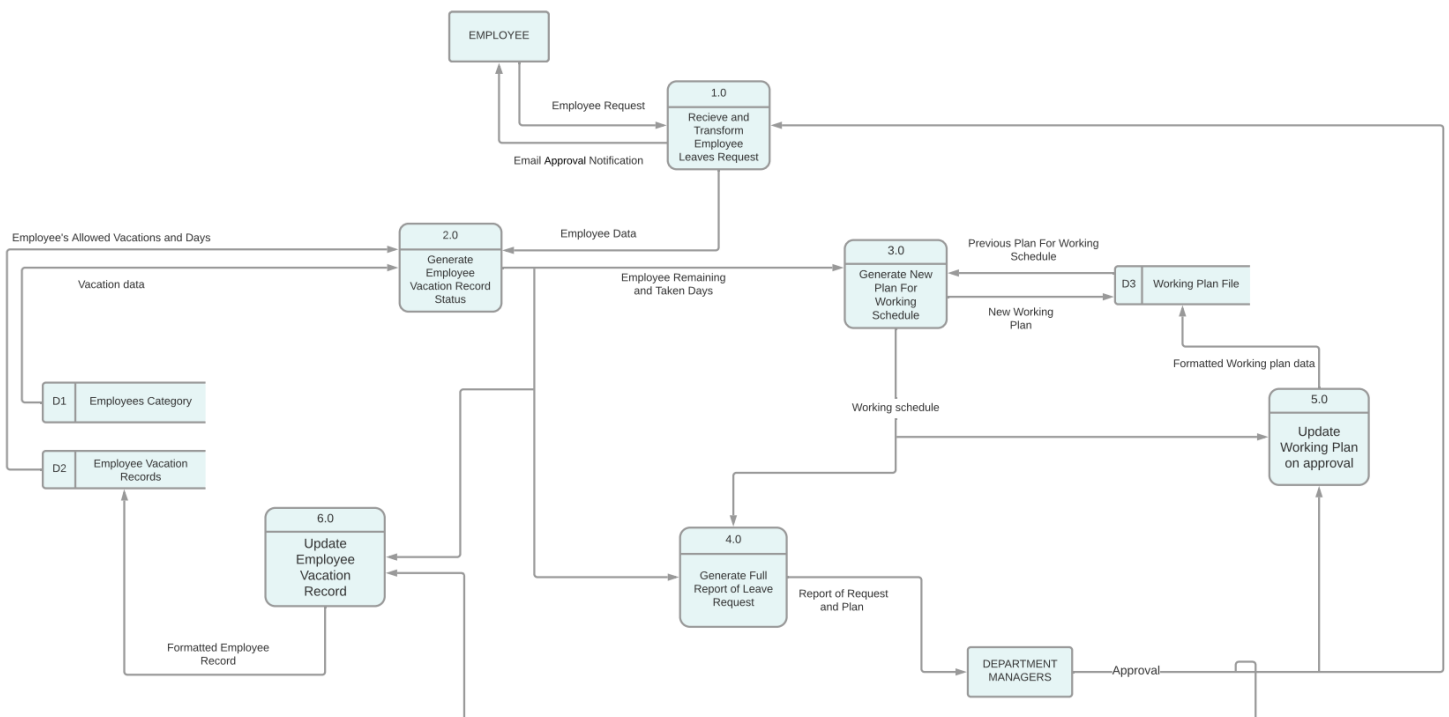


## Data flow Diagram

### Context Diagram



### Level-0 diagram



## **2. System Functions and Modules**

### System Functions

There are multiple features on the website that allow the administrator to view employees' requests and vacations records, along with remaining days, besides allowing employees to make leave requests, it also provides access to their vacation records, remaining vacation days, and request status.

#### **The website from the employees' point of view:**

- The website home page has a login request for id and password
- The employees can log in easily and start using the website for self-service
- There is a Leave request section; it has a filter option to choose which department the employee works for
- After choosing the department, the employee can choose which type of leave request
- The Leave request contains an empty form of leave description and how many remaining days are left
- After filling the form, the employee may proceed to send it to administrators
- The administrator of the selected department will receive the request and proceed to accept or decline it
- The employee receives their request status

### **The website from the administrator's point of view:**

- Administrator of department login to the website
- On the website, there is a section for a list of requests for each employee in the department. Including name, id, and the request type
- The administrator can click on employee requests and read the document request easily
- When the administrator accepts the request
- The system calculates and updates the remaining request left
- The system sends to the employee a request notification in email and on the website in the request status section

### System Modules

- 1.0: Receive and transform employee leaves request
- 2.0: Generate employee vacation record status
- 3.0: Generate a new plan for the working schedule
- 4.0: Generate a full report of leave request
- 5.0: Update working plan on approval
- 6.0: Update employee vacation record

### Support Systems

- PC
- DB server
- Web server

### 3. Modules and Estimation Results

SLOC Input Dialog - 1.0

**Sizing Method**

- ☒ SLOC
- ☐ Function Points
- ☐ Adaptation and Reuse

**Breakage**  
% of code thrown away due to requirements evolution and volatility

REVL

**Module Size in SLOC**

Language

SLOC

OK Cancel Help

**SLOC** is used for the sizing method of receiving and transforming the employee leaves request process.

Program language is HTML 3.0, there is an estimated 3000 lines of code, and 10% of the code will be discarded due to **requirements evolution and volatility (REVL)**.



SLOC Input Dialog - 2.0

**Sizing Method**

☒ SLOC  
☐ Function Points  
☐ Adaptation and Reuse

**Breakage**  
% of code thrown away due to requirements evolution and volatility  
REVL

**Module Size in SLOC**

Language  SLOC

OK Cancel Help

**SLOC** is used for the sizing method of generating employee vacation record status processes.

Program language is HTML 3.0, there is an estimated 4000 lines of code, and 15% of the code will be discarded due to **requirements evolution and volatility (REVL)**.

SLOC Input Dialog - 3.0

**Sizing Method**

☐ SLOC  
☒ Function Points  
☐ Adaptation and Reuse

**Breakage**  
 % of code thrown away due to requirements evolution and volatility  
 REVL

**Module Size in Function Points**

Language

Ratio Type : ☒ Jones ☐ David

Calculation Method : ☒ Using Table ☐ Input Calculated Function Point

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	7
Outputs	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="0"/>	14
Files	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	7
Interfaces	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Queries	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Total Unadjusted Function Points				28
Equivalent Total in SLOC				420

The function point for cost estimation is used in generating a new plan for the working schedule process, based on the amount of **functionality** in a software project and a set of individual project factors.

Program language is HTML 3.0, and 10% of the code will be discarded due to **requirements evolution and volatility (REVL)**.

The process has two inputs (**low**: Working plan file, **average**: Generate employee vacation record status process). Furthermore, it has two average outputs which are (process 4.0 and process 5.0), and it also has one file (Working plan file).

SLOC Input Dialog - 4.0

**Sizing Method**

☐ SLOC

☒ Function Points

☐ Adaptation and Reuse

**Breakage**  
% of code thrown away due to requirements evolution and volatility

REVL

**Module Size in Function Points**

Language   15

Ratio Type : ☒ Jones ☐ David

Calculation Method : ☒ Using Table ☐ Input Calculated Function Point

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	<input type="text" value="0"/>	<input type="text" value="2"/>	<input type="text" value="0"/>	8
Outputs	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	4
Files	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Interfaces	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Queries	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Total Unadjusted Function Points				12
Equivalent Total in SLOC				180

The function point for cost estimation is used in generating a full report of the leave request process, based on the amount of **functionality** in a software project and a set of individual project factors.

Program language is HTML 3.0, and 10% of the code will be discarded due to **requirements evolution and volatility** (REVL).

The process has two average inputs (process 2.0 and process 3.0). Furthermore, it has one average output which are (Department manager file), and it also has one file (Department manager file).

SLOC Input Dialog - 5.0

**Sizing Method**

☐ SLOC  
☐ Function Points  
☒ Adaptation and Reuse

**Breakage**  
 % of code thrown away due to requirements evolution and volatility  
 REVL

**Adaptation**

Initial SLOC

Language

% Design Modified (DM)	<input type="text" value="60"/> %
% Code Modified (CM)	<input type="text" value="30"/> %
% Integration Modified (IM)	<input type="text" value="25"/> %
Software Understanding (SU)	<input type="text" value="16.6"/> SU
Assesment & Assimilation (AA)	<input type="text" value="4"/> AA
Unfamiliarity with Software	<input type="text" value="0.4"/> UNFM
% Components Automatically Translated (AT)	<input type="text" value="15"/> %
Automatic Translation Productivity (ATPROD)	<input type="text" value="2400"/>

Computed Adaptation Adjustment Factor	40.5
Computed ASLOC	1059

OK Cancel Help

**Adoption and Reuse** method that we used to sizing the update working plan on approval process.

The process is coded using HTML 3.0 programming language, there is an estimated 2500 source lines of code adapted from existing software used in developing the new product, 60% of the adapted software's design that received modification, 30% modification applied on the code, 25% effort needed for integrating and testing of the adapted software in order to combine it into the new product, the reuse effort due to Software Understanding (SU) is 16.6 which means it is understandable, the reuse effort due to Assessment and Assimilation (AA) is 4. Programmer Unfamiliarity with Software (UNFM) is 0.4 UNFM which means it is familiar, and the AT is 15%.

SLOC Input Dialog - 6.0

**Sizing Method**

☐ SLOC  
☐ Function Points  
☒ Adaptation and Reuse

**Breakage**  
 % of code thrown away due to requirements evolution and volatility  
 REVL

**Adaptation**

Initial SLOC

Language

% Design Modified (DM)  %

% Code Modified (CM)  %

% Integration Modified (IM)  %

Software Understanding (SU)  SU

Assesment & Assimilation (AA)  AA

Unfamiliarity with Software  UNFM

% Components Automatically Translated (AT)  %

Automatic Translation Productivity (ATPROD)

Computed Adaptation Adjustment Factor 40.5

Computed ASLOC 1059

OK Cancel Help

**Adoption and Reuse** method that we used to sizing the update employee vacation record process.

The process is coded using HTML 3.0 programming language, there is an estimated 2500 source lines of code adapted from existing software used in developing the new product, 60% of the adapted software's design that received modification, 30% modification applied on the code, 25% effort needed for integrating and testing of the adapted software in order to combine it into the new product, the reuse effort due to Software Understanding (SU) is 16.6 which means it is understandable, the reuse effort due to Assessment and Assimilation (AA) is 4. Programmer Unfamiliarity with Software (UNFM) is 0.4 UNFM which means it is familiar, and the AT is 15%.

## 4. Conclusion

Project Name:  Scale Factor:  Schedule

Development Model:

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
1.0	S:3300	5000.00	1.04	HTML 3.0	12.3	12.7	258.9	63726.19	19.3	1.1	0.0	
2.0	S:4600	5000.00	1.04	HTML 3.0	17.1	17.8	258.9	88830.45	19.3	1.5	0.0	
3.0	F:462	5000.00	1.04	HTML 3.0	1.7	1.8	258.9	8921.67	19.3	0.1	0.0	
4.0	F:198	3000.00	1.04	HTML 3.0	0.7	0.8	258.9	2294.14	11.6	0.1	0.0	
5.0	A:1059	2000.00	1.04	HTML 3.0	4.1	4.2	249.4	8492.63	8.0	0.4	0.0	
6.0	A:1059	2000.00	1.04	HTML 3.0	4.1	4.2	249.4	8492.63	8.0	0.4	0.0	

	Estimated	Effort Sched	PROD	COST	INST	Staff	RISK
Total Lines of Code: 10678	Optimistic	33.2	11.2	321.2	144606.16	13.5	3.0
Hours/PM: 152.00	Most Likely	41.6	12.0	257.0	180757.70	16.9	3.5
	Pessimistic	51.9	12.9	205.6	225947.13	21.2	4.0

- We have been dealing with the Post Architecture model
- No RISK assessment
- The total lines of code estimated is 10678

One standard deviation from the mean.

- **NOM DEV:** Nominal Person Man Months exclusive of EAF
- **EST DEV:** Median Person Months *inclusive* of EAF
- **PROD:** SLOC / EST DEV Effort. So the unit is Source Lines Of Code per Man Month
- **Cost:** If we had entered a Labor Rate, the cost would be calc'd
- **INST COST:** calculated most likely cost per instruction. This number is calculated from Cost/SLOC in each module
- **Staff:** most likely estimate for the number of full-time developers that would be needed to complete a module in the estimated development time
- **RISK:** total\_risk = schedule\_risk + product\_risk + personnel\_risk + process\_risk + platform\_risk + reuse\_risk. Then total risk of a module = total\_risk/373. \*100

**Process 1.0:**

Nominal Person-Month (NOM PM) is 12.3. Estimated Person-Month (EST PM) is 12.7

Productivity (PROD) is 258.9. Staff is 1.1

This process is based on SLOC (the S:) and it has an equivalent 3300 lines of code.

**Process 2.0:**

Nominal Person-Month (NOM PM) is 17.1. Estimated Person-Month (EST PM) is 17.8

Productivity (PROD) is 258.9. Staff is 1.5

This process is based on SLOC (the S:) and it has an equivalent 4600 lines of code.

**Process 3.0:**

Nominal Person-Month (NOM PM) is 1.7. Estimated Person-Month (EST PM) is 1.8

Productivity (PROD) is 258.9. Staff is 0.1

This process is based on Function points (the F:) and it has an equivalent 462 lines of code.

**Process 4.0:**

Nominal Person-Month (NOM PM) is 0.7. Estimated Person-Month (EST PM) is 0.8

Productivity (PROD) is 258.9. Staff is 0.1

This process is based on Function points (the F:) and it has an equivalent 198 lines of code.

**Process 5.0:**

Nominal Person-Month (NOM PM) is 4.1. Estimated Person-Month (EST PM) is 4.2

Productivity (PROD) is 249.4. Staff is 0.4

This process is based on Adaptation and Reuse (the A:) and it has an equivalent 1059 lines of code.

**Process 6.0:**

Nominal Person-Month (NOM PM) is 4.1. Estimated Person-Month (EST PM) is 4.2

Productivity (PROD) is 249.4. Staff is 0.4

This process is based on Adaptation and Reuse (the A:) and it has an equivalent 1059 lines of code.



base + Incr % = rating

Product:

RELY

DATA

DOCU

CPLX

RUSE

base

NOM

HI

NOM

NOM

NOM

Incr%

0%

0%

0%

0%

0%

Platform:

TIME

STOR

PVOL

base

NOM

NOM

NOM

Incr%

0%

0%

0%

Personnel:

ACAP

PCAP

PCON

APEX

LTEX

PLEX

base

NOM

NOM

NOM

NOM

HI

NOM

Incr%

0%

0%

0%

0%

0%

0%

Project:

TOOL

SITE

base

NOM

NOM

Incr%

0%

0%

User:

USR1

USR2

base

NOM

NOM

Incr%

0%

0%

EAF is also affected by Schedule

EAF: 1.04

OK

Cancel

Help

**Effort Adjustment Factor (EAF) Column** - This displays the product of the cost drivers for each specific module.

**LTEX:** Language and tool experience [HI]

**DATA:** Database size [HI]

Product Parameters - Default model values used

	VLO	LO	NOM	HI	VHI	XHI
RELY	0.82	0.92	1.00	1.10	1.26	XXXX
DATA	XXXX	0.90	1.00	1.14	1.28	XXXX
DOCU	0.81	0.91	1.00	1.11	1.23	XXXX
CPLX	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	XXXX	0.95	1.00	1.07	1.15	1.24

OK Reset Cancel Help

Scale Factors

	base	Incr%
Precedentedness .....	NOM	0%
Development Flexibility .....	NOM	0%
Architecture / risk resolution	NOM	0%
Team cohesion .....	NOM	0%
Process maturity .....	NOM	0%

Scale Factor : 18.97

OK Cancel Help

Project Note

Project Name: Leaves System

▲ Elaf Yousef Aloufi - 1911265  
 Manar Mutlaq Altairy - 1906775  
 Randa Ibrahim Zhrani - 1907096  
 Ghaliyah Banounah - 1906829

▼

OK Cancel