

IT314 Software Engineering

Lab 4: Specifying Tools and Technology

Date: 09-03-2023

Group - 13



- Group Members:

ID No	Name
202001133	DANKHARA MANAN SHAILESHKUMAR
202001163	SHAH ISHA JATINKUMAR
202001149	BUCH KAVAN JAYESHBHAI
202001168	PRAJAPATI SHREYA
202001135	PATEL DHARVA YOGESHKUMAR
202001148	BRIJRAJSINH GOHIL
202001137	MAKWANA KIRTAN DHARMENDRABHAI
202001152	SANEPARA KHUSHI SURESHBHAI
202001155	RAMOLIYA HARSH MADHUKANTBHAI

a. Tools, technologies and framework we are using for our project are as follows:

❖ Front-end:

- We are using **HTML, CSS, JavaScript** as frontend and rather than HTML we are using Embedded JavaScript template(Ejs) framework.
- We will be using the **Bootstrap framework** for designing purposes as it provides an easy and simple way to design some basic features with responsive design.

❖ Back-end:

- For the backend we will be using **Node JS** which is used for scalable network applications. One important functionality of Node Js is that it can run multiple functions concurrently and upon completion of each function a callback is fired, but if there is no work going on Node Js sleeps.
- Along with Node Js, we are using the **Express JS framework** which provides a robust set of features for web applications.
- Some other frameworks like http, bcrypt, express-session, json web token and many more frameworks of Node JS are used as and when needed in a project.

❖ IDE:

- We are using **Visual Studio code IDE** for development. VS code helps in using extensions which enhance our development experience.
- It also allows us to integrate with git so we can directly manage our version control from the editor.

b. NO SQL Database for our system

- NO SQLs databases are non-tabular databases and store data differently rather than relational tables. There are main four types of storing patten:
 1. Document format
 2. Key-Value format
 3. Wide-Column format
 4. Graph format
- We are using **MongoDb** as our database, which stores data in document form and the type of Document is Javascript Object Notation (JSON).
- Storing data in JSON format provides flexibility in storing data of all types.
- These databases serve a wide variety of use cases and can employ complex querying using multikey or full-text search.
- The best feature of MongoDB is that it has a scale out feature so that applications can run anywhere on-premises or on cloud.

C. Estimate the effort of your project and narrow down the scope based on the estimation.

- **Use-case points:**

- Use-case point = UUCW(Unadjusted use-case weight) + UAW(Unadjusted Actor Weight) + TCF(Technical Complexity Factor) + ECF(Environmental complexity Factor)
- So, First we find Unadjusted use-case weight.

- **Unadjusted use-case weight:**

- First, we have to find unadjusted use-case weight

- We have to find use-case complexity according to below table:

Use-case Classification	No.of Transaction	Weight
Simple	1 to 3 transactions	5
Average	4 to 7 transactions	10
Complex	8 or more transactions	15

- **Use-case complexity for Our System:**

Use-case	Classification
Manage course content	Simple
Add course content	Average
Remove course content	Average
Discuss Doubts	Average
Search Course	Simple
View course content	Simple
Rate the courses	Average
Login	Average
Details Validation	Simple
Add Users	Simple
Remove Users	Average
Manage Course	Simple
Add Course	Average
Remove Course	Average

Final UUCW

= (No of Simple use-case) * 5 + (No of average use-case)*10 + (No of Complex use-case)*15

= (6*5) + (7*10) + (0*15)

= 30 + 70 + 0

= 100

→ Now,we have to find Unadjusted actor weights.

- We find unadjusted actor weight as below table

Actor Classification	Type of Actor	Weight
Simple	External system that interact with the system using well-defined API	1
Average	External system that should interact with our system using standard communication protocol	2
Classification	Human actor using an GUI interface	3

- Unadjusted-actor weight for our system:

Actors	Classification of actor
Educator	Complex
Student	Complex
Admin	Complex
System	Complex

Final UAW

$$= (4*3) + (0*2) + (0*1)$$

$$= 12$$

Technical Complexity Factor (TCF):

Factor	Description	Weight	Relevance
T1	Distributed System	2.0	2.0
T2	Response time/performance objectives	1.0	5.0
T3	End-user efficiency	1.0	3.0
T4	Internal processing complexity	1.0	3.0
T5	Code reusability	1.0	4.0
T6	Easy to install	0.5	0.0
T7	Easy to use	0.5	5.0
T8	Portability to other platforms	2.0	2.0
T9	System maintenance	1.0	4.0
T10	Concurrent/parallel processing	1.0	4.0
T11	Security features	1.0	4.0
T12	Access for third parties	1.0	2.0
T13	End user training	1.0	4.0

$$\text{Technical Factor(TF)} = \sum \text{Weight} * \text{Relevance}$$

$$= 43.5$$

$$\text{Technical Complexity Factor(TCF)} = 0.6 + \text{TF} / 100$$

$$= 0.6 + 43.5/10 = 1.035$$

Environmental Complexity Factor (ECF):

Factor	Description	Weight	Relevance
E1	Familiarity with development process used	1.5	3.0
E2	Application experience	0.5	4.0
E3	Object-oriented experience of team	1.0	4.0
E4	Lead analyst capability	0.5	4.0
E5	Motivation of the team	1.0	5.0
E6	Stability of requirements	2.0	5.0
E7	Part-time staff	-1.0	2.0
E8	Difficult programming language	-1.0	3.0

$$\text{Environment Factor(EF)} = \sum \text{Weight} * \text{Relevance}$$

$$= 22.5$$

$$\text{Environment Complexity Factor(ECF)} = 1.4 + (-0.03 \times \text{EF})$$

$$= 1.4 + (-0.03 \times 22.5)$$

$$= 0.725$$

$$\text{Use-case point(UCP)} = (\text{UUCW} + \text{UAW}) * \text{TCF} * \text{ECF}$$

$$= (100 + 12) * 1.035 * 0.725$$

$$= 112 * 1.035 * 0.725$$

$$= \mathbf{84.042}$$

❖ Now, Assuming to develop one use-case point we require 4 man-hours.

So, Man-hour/use-case point = 4 hours

$$\text{Total time Estimation} = \text{UCP} * 4$$

$$= 84.042 * 4$$

$$= 336.17 \text{ man-hours}$$

$$= \mathbf{337 \text{ man-hours}}$$