IT314 - Software Engineering

Lab 4 - Specifying Tools and Technologies

Educational Networking Tool for College Students

Group - 29

Group Members:

- 202001150 - Devdeep Shetranjiwala
- 202001173 - Harsh Patel
- 202001270 - Naisarg Shah
- 202001412 - Zeel Bhanderi
- 202001424 - Kaushal Patel
- 202001431 - Shrey Shah
- 202001441 - Priyanshi Parmar
- 202001449 - Achyut Shah
- 202001452 - Krish Patel
- 202001461 - Harsh Agheda
- 202001462 - Om Patel

Tools and Technology, Frameworks and Database

Tools and Technology: Material UI, VS Code, JWT, Postman, NodeMailer
Frameworks: React JS, Node JS, Express JS
Database: ElasticSearch / MongoDB

Postman:

Postman is an API creation and testing tool for developers that allows them to design, construct, and test APIs. It makes interacting with APIs easier by offering a user-friendly interface for making queries, configuring environments, and evaluating replies. It supports a wide range of protocols and formats, making it a versatile API creation tool.

VS Code:

Visual Studio Code, often referred to as VS Code, is a free and open-source code editor developed by Microsoft. It offers an extensive library of extensions and a customizable user interface, making it a popular choice for software developers. With built-in support for debugging, version control, and various programming languages, VS Code is a versatile tool that can be used for a wide range of coding projects.

Material UI:

Material UI is a popular React component library that provides a set of pre-built UI components based on Google's Material Design principles. It includes components such as buttons, cards, menus, and icons that can be easily integrated into React applications. Material UI also offers customization options and theming support to enable developers to create visually appealing and consistent user interfaces.

JWT:

JWT (JSON Web Token) is an open standard for securely transmitting information between parties as a JSON object. It is commonly used for authentication and authorization purposes in web applications and APIs, as it can verify the integrity of the data and protect against tampering or unauthorized access.

NodeMailer:

Nodemailer is a popular open-source Node.js module used to send emails from Node.js applications. Nodemailer provides a simple and straightforward API for sending emails, with support for attachments, HTML content, and more. It also includes features such as email templating and email scheduling to further enhance the functionality of the module.

ReactJS:

React is a JavaScript library for building reusable and efficient user interfaces. We can use React to make a single page application. It uses a virtual DOM to update the UI, minimizing the number of actual DOM manipulations and improving performance. React can be used to build web, mobile, and desktop applications but we will first use it for web and it has a large and active developer community which can be helpful at the times of bugs and fallbacks.

NodeJS:

Node.js is an open-source, cross-platform, server-side JavaScript runtime environment that allows developers to build fast and scalable network applications. It has a large and active community of developers, which has led to the creation of a vast ecosystem of modules and tools that can be easily integrated into Node.js applications.

ExpressJS:

Express.js is a popular open-source framework for building web applications on top of Node.js. It provides a set of flexible and powerful features for building APIs, handling requests and responses, and managing middleware. Express.js is easy to use and has a large and active community of developers creating useful extensions and libraries.

ElasticSearch/ MongoDB:

Elasticsearch is a distributed search and analytics engine for fast and reliable search and analysis of large volumes of data. MongoDB is a flexible NoSQL document-oriented database used for storing and retrieving data of any structure, with support for indexing, replication, and high availability. We will be implementing ElasticSearch at first and use MongoDB as a backup.

Effort of Project:

Step 1: Calculate Unadjusted Use-case Points

Determine Unadjusted Use-case Weight

Use Case	Transactions	Complexity
Register	4	Average
Login	3	Simple
Search	3	Simple
Add Post	8	Complex
Delete Post	6	Average
Upvote/Downvote	4	Simple
Follow User	3	Simple
Unfollow User	3	Simple
Update Profile Details	7	Average
Comment on Post	6	Average
Chat with User	8	Complex

Use Case Classification	Number of Use Case	Weight Assigned	Product
Simple	5	5	25
Average	4	10	40
Complex	2	15	30

Unadjusted Use Case Weight (UUCW) = 5x5 + 4x10 + 2x15 = 95

Actors	Complexity
Users	Complex
System/Database	Average

Actor Complexity	Actor Weight	Number of Actors	Product
Simple	1	0	0
Average	2	1	2
Complex	3	1	3

Unadjusted Actor Weight (UAW) = 1x0 + 2x1 + 3x1 = 5

Unadjusted Use-case Points (UUCP) = UUCW + UAW = 100

Step 2: Adjust for Technical Complexity

Factor	Description	Weight	Rated Value (0 to 5) (RV)	Impact (I = W × RV)
T1	Distributed System	2.0	5	10
T2	Response time or throughput performance objectives	1.0	5	5
T3	End-user efficiency	1.0	3	3
T4	Complex internal processing	1.0	4	4
T5	Code must be reusable	1.0	5	5
T6	Easy to install	0.5	4	2
T7	Easy to use	0.5	4	2
T8	Portable	2.0	5	10
T9	Easy to change	1.0	4	4
T10	Concurrent	1.0	3	3
T11	Includes special security objectives	1.0	3	3
T12	Provides direct access for third parties	1.0	3	3
T13	Special user training facilities are required	1.0	2	2

Impact of the Factor
= Impact Weight × Rated Value
= 10 + 5 +3+4+5+2+2+10+4+3+3+3+2
= 56

TCF = 0.6 + (0.01 × TFactor) = 0.6 + 0.56 = 1.16

Step 3: Adjust for Environmental Complexity

Factor	Description	Weight (W)	Rated Value (0 to 5) (RV)	Impact (I = W × RV)
F1	Familiar with the project model that is used	1.5	4	**6**
F2	Application experience	0.5	0	**0**
F3	Object-oriented experience	1.0	3.5	**3.5**
F4	Lead analyst capability	0.5	1	**0.5**
F5	Motivation	1.0	4	**4**
F6	Stable requirements	2.0	3	**6**
F7	Part-time staff	-1.0	2	**-2**
F8	Difficult programming language	-1.0	3	**-3**
Total Environment Factor (EFactor)				**15**

EF = 1.4 + (-0.03 * EFactor)

For Educational Networking Tool for College Students,
EF = 1.4 + (-0.03*15) = 0.95

EF = 0.95

Step 4: Calculating Adjusted Use case points

UCP = UUCP * TCF * EF
= 100 * 1.16 * 0.95
=110.2

Here, We take 4 man hours per UCP

Estimated Efforts = UCP * Hours/UCP
= 110.2*4
= 440.8 hours

Here, We consider 50 man hours per week.

Thus, the Estimated week will be around 8.