Lab-4

IT314_Guesthouse_booking_system_16

GROUP NO: 16

Members:

202001156 - KHAMAR AYUSH KINCHITKUMAR
202001146 - MAKADIYA MUDIT BHAVESHBHAI
202001138 - GANDHI SOMIN
202001129 - DEEP JAIMESH SHAH
202001159 - RADADIYA DARPAN VITTHALBHAI
202001161 - GAJERA AYUSH KAMLESHBHAI
202001165 - PATEL HARSH BHARATBHAI
202001175 - ANKIT LOCHAN
202001170 - RAVAL KEVIN KIRITKUMAR
202001126 - PATEL AYUSH MUKESHBHAI

1.TOOLS AND TECHNOLOGY

1) Frontend : We are Planning to use **EJS** because It allows for dynamic rendering of HTML templates with embedded JavaScript code**, React.js** because it provides a powerful toolset for building reusable UI components. This enables us to create complex user interfaces with ease, improve the performance of web applications and **javascript** and **CSS.**

2) Backend : We are planning to use **Node.js** because of its high performance, scalability, JavaScript-based architecture**, express.js** because it provides a robust framework for building scalable and maintainable web applications. Additionally, using **NPM packages** allows us to easily incorporate third-party libraries and tools to enhance our project's functionality**.**

3) Database : We are planning to use open source NoSQL database **MongoDB** because of its flexibility and scalability. It offers a document-based data model that allows for easy storage and retrieval of complex data structures

4) IDE : We are planning to use **Visual Studio Code** as our IDE because of its open-source nature, cross-platform support, and rich extension ecosystem, which provides a flexible and customizable environment.

2.EFFORT ESTIMATION USING USE CASE SIZE POINT

2.1 Unadjusted Use-Case Weight (UUCW)

User case Complexity	Number of Transactions	Use-Case Weight
Simple	<=3	5
Average	4 to 7	10
complex	>7	15

Use case name	Number of transaction	Category
Customer		
Authenticate	2	Simple
Customer Login	2	Simple
View property / rooms	1	Simple
Filter results	2	Simple
Register Checkin	1	Simple
Reserve facility/room	1	Simple
Register checkout	1	Simple
Make Payment	2	Simple
View Payment Report	2	Simple

Apply for cancellation	2	Simple
View Cancellation status	1	Simple
Contact admin	1	Simple
Admin		
Admin login Authentication	2	Simple
Admin Login	1	Simple
Check room availability	2	Simple
Handle waiting list	2	Simple
View Payment report	1	Simple
Check that user requesting cancellation is a member or not	1	Simple
Apply appropriate cancellation policy	1	Simple
Validate documents to verify hotel	2	Simple
Check if hotel is registered	1	Simple
Hotel staff		
Hotel staff authentication	2	Simple
Hotel staff login	1	Simple
Check room availability	2	Simple
Handle waiting list	2	Simple
Check check-in register to lend room	1	Simple
View Payment report	1	Simple

Register property	4	Average
Deregister hotel	2	Simple

Use-Case Complexity	Weight	Number of Use-Cases	Product
Simple	5	27	135
Average	10	1	10
Complex	15	0	0
Unadjusted Use-case Weight (UUCW)			145

2.2 Unadjusted Actor Weight (UAW)

Actor Complexity	Example	Actor Weight
Simple	A System with defined API	1
Average	A System interacting through a Protocol	2
Complex	A User interacting through GUI	3

Actor name	category	weight
Admin	complex	3
Hotel Staff	complex	3

Customer	complex	3
Payment Gateway(API)	simple	1
Unadjusted Actor Weight (UAW)		10

2.3 Unadjusted Use Cast Point (UUCP)

Unadjusted Use Case Point = Unadjusted Actor Weight(UUCW) + Unadjusted Use Case Weight(UAW)

Unadjusted Use Case Point = 10 + 145 = **155**

2.4 Technical Complexity Factor (TCF)

Factor	Description	Weight (W)	Rated Value (0 to 5) (RV)	Impact (I = W × RV)
T1	Distributed System	2.0	3	6
T2	Response time or throughput performance objectives	1.0	5	5
T3	End user efficiency	1.0	5	5
T4	Complex internal processing	1.0	4	4

T5	Code must be reusable	1.0	5	5
T6	Easy to install	0.5	0	0
T7	Easy to use	0.5	5	2.5
T8	Portable	2.0	4	8
T9	Easy to change	1.0	4	4
T10	Concurrent	1.0	5	5
T11	Includes special security objectives	1.0	5	5
T12	Provides direct access for third parties	1.0	2	2
T13	Special user training facilities are required	1.0	0	0
Total Technical Factor (TFactor)				51.5

Technical Complexity Factor can be calculated as follows:

∴ TCF = 0.6 + (0.01 × TFactor)

∴ TCF = 0.6 + 0.01*51.5

∴ TCF = 0.6 + 0.515

∴ TCF = **1.115**

2.5 Environmental Complexity Factor (EF)

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	4	2
F3	Object-oriented experience	1.0	5	5
F4	Lead analyst capability	0.5	4	2
F5	Motivation	1.0	4	4
F6	Stable requirements	2.0	4	8
F7	Part-time staff	-1.0	0	0
F8	Difficult programming language	-1.0	2	-2
Total Environment Factor (EFactor)				25.5

Environmental Factor = 1.4 + (-0.03 × EFactor)

= 1.4 - 0.765 = **0.635**

Factor	Description	Weight
UUCP	Unadjusted use case point	155
TCF	Technical Complexity Factor	1.115
EF	Environmental factor	0.635

UCP = UUCP × TCF × EF

UCP = **109.75**

Total Working Hours = UCP X (Working Hours per UCP)

Working Hours/UCP = 5

Total Working Hours = **548.8 hrs**