Dhirubhai Ambani
Institute of Information and Communication Technology

IT314 - Software Engineering
Lab 4 - Specifying Tools and Technology

Date of submission: 01-03-2023

Due date of submission: 01-03-2023

Group Members:

1. 202001063 - Bhalodiya Hem Pareshbhai
2. 202001066 - Japan Vijay Bhatt
3. 202001068 - Dhrupal Kukadia
4. 202001078 - Shashank Didwania
5. 202001081 - Ronit Jain
6. 202001083 - Patel Vedant Vipulbhai
7. 202001093 - Parmar Dhruv Jayeshbhai
8. 202001106 - Hardi Sanghani
9. 202001115 - Aditya Kothari

Project Name: Location Sharing System

Contents:

Tools, Technologies and Frameworks

Tools : Android Studio, mapbox / Google maps API
Technology : Android, iOS
Frameworks : Flutter

Android studio:

Android Studio is the official Integrated Development Environment (IDE) for Android and App development. It is developed and maintained by Google. So there are many resources available for it. It includes a large community of developers so that one can get help whenever required. It also contains a wide range of plugins and supports different types of languages like Java, Kotlin, C++, etc. It also contains an emulator so one can run and debug their application.

Mapbox :

Mapbox API is for custom maps. Mapbox is a platform for designing and delivering custom maps, geocoding and routing API's and location-based services. It provides developers with a suite of tools for building applications that visualize, analyze, and interact with geographic data. It also supports various languages and frameworks. It provides SDKs for both android and iOS.

Google maps API :

Google Maps API is a service provided by Google that allows developers to integrate maps and location-based features into their applications. The API includes a range of tools and services for creating and customizing interactive maps, geocoding, routing, and location-based services. It also supports various languages and provides SDKs for both android and iOS. It is widely used by developers and has quite good documentations.

Flutter :

It is an open-source app development framework made by Google. It is easy to build high-performance, visually attractive, and natively compiled applications using it. It supports all mobile, web, and desktop platforms from a single codebase. Flutter has a widget-based architecture, which allows developers to create complex user interfaces easily. It uses the Dart programming language to add logic in your application.

Data Base : MongoDB

MongoDB is a NoSQL database that can handle large volumes of geospatial data. It offers geospatial indexing and querying capabilities, allowing for fast retrieval of location-based data. MongoDB can also be easily integrated with various mapping libraries and frameworks. It also includes built-in authentication service for secure access. It also supports storing data on the cloud.

Effort of Project

Step 1 : Calculate Unadjusted Use-Case Points

- **Determine Unadjusted Use-Case Weight**

Use case	Transactions	Complexity
Log in	3	Simple
Search location	3	Simple
Manage friend	4	Average
Share live location	3	Simple
Emergency SOS	4	Average
Meetup with friends	2	Simple
Nearby me	2	Simple

Use-Case Complexity	Use-Case Weight	Number of Use-Cases	Product
Simple	5	5	25
Average	10	2	20
Complex	15	0	0
Unadjusted Use case weight (UUCW)			**45**

- **Determine Unadjusted Actor Weight**

Actors	**Complexity**
Users	Complex
Developer	Average
System	Simple

Actor Complexity	**Actor Weight**	**Number of Actors**	**Product**
Simple	1	1	1
Average	2	1	2
Complex	3	1	3
Unadjusted Actor Weight (UAW)			**6**

- **Calculate Unadjusted Use-Case Points**

  Unadjusted Use-Case Points (UUCP) = UUCW + UAW = 45 + 6 = 51

Step 2: Adjust For Technical Complexity

Factor	Description	Weight (W)	Rated Value (0 to 5) (RV)	Impact (I = W × RV)
T1	Distributed System	2.0	0	0
T2	Response time or throughput performance objectives	1.0	4	4
T3	End user efficiency	1.0	4	4
T4	Complex internal processing	1.0	2	2
T5	Code must be reusable	1.0	3	3
T6	Easy to install	.5	4	2
T7	Easy to use	.5	5	2.5
T8	Portable	2.0	3	6
T9	Easy to change	1.0	3	3
T10	Concurrent	1.0	2	2
T11	Includes special security objectives	1.0	3	3
T12	Provides direct access for third parties	1.0	1	1
T13	Special user training facilities are required	1.0	0	0
Total Technical Factor (TFactor)				**32.5**

TCF = 0.6 + (TF / 100)

= 0.6 + (32.5 / 100)

= 0.925

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	.5	4	2
F3	Object-oriented experience	1.0	3	3
F4	Lead analyst capability	.5	5	2.5
F5	Motivation	1.0	0	0
F6	Stable requirements	2.0	2	4
F7	Part-time staff	-1.0	0	0
F8	Difficult programming language	-1.0	4	-4
Total Environment Factor (EFactor)				**13.5**

Calculate the Environmental Factor (EF) = 1.4 + (-0.03 × EFactor)

= 1.4 + (-0.03 × 13.5)

= 0.995

Step 4: Calculate Adjusted Use-Case Points (UCP)

Calculate Adjusted Use-Case Points (UCP) = UUCP × TCF × EF

= 51 x 0.925 x 0.995

= <u>46.93</u>

Here, We take 6 man hours per UCP

Estimated Efforts = UCP × Hours / UCP

= 46.93 × 6

= <u>281.58 hours</u>

Here, We consider 40 man hours per week.

Thus, Estimated week will be around 7.