SE 315 SOFTWARE PROJECT MANAGEMENT

NIGHT JUNGLE

Software Development Plan

Elif Duygu PETENKAYA

Elif Berna KURU

Melike ERDEMİR

Çağatay YALÇIN

05/05/2020

Contents

1.OVERVIEW

Night Jungle is an application that provides its users to find the appropriate entartainment facility.The project based on a mobile application in Android platform and developed in Flutter..There are two types of registiration as customer and venue.

The customer interface has the following properties; reservation, event checking, online payment.

The venue interface has the folowing properties; event add/drop, photo sharing, online payment, campaign sharing, reservation admit/remove. Venue interface will be used by venue owner, business manager and accountant.

2. HIGH-LEVEL FUNCTIONALITY

People can find the venue /event that is appropriate for their criterions. They can easily make their booking and payment processes via the applications.

2.1.Requirements

2.1.1.Functional Requirements

- The system should show information about the venue likewise menu, location, concept, campaigns and contact details.
- The system should allow to user booking and paying.
- The system should share the comments about the venue that are provided by the users.
- The system should provide an event calendar on the main page.
- The system should have two types of registration as venue owner and user.
- The system should allow the user to share photos and videos from the venue.
- The system should register users using their name, surname and e-mail information.
- The system should register venue owners using tax plate number, business name and business e- mail information.

2.2.2.Non-functional Requirements

- The system should disallow to spam comments.
- The system should make the payment with two-step verification.
- The system should refresh itself when an event starts and remove the corresponding announcement about that event from the dashboard.
- The response time should be less than 5 seconds.
- The system should allow the user to change his contacts information whenever it is necessary.
- The system should provide to sign up the venue owners via the system.
- The system should show the comment owner's name as X*** Y***.

3.STAKEHOLDERS

There are three types of stakeholders in this project;

- Development team
- Customers
- Venue Staff

3.1. Development Team

The developers and the project manager are the natural stakeholders of the project. The categorization of all developers is explained in detail at "Project Staffing" section.

3.2. Customers

The customer is the person who will use the system to select his venue/event to go.

3.3. Venue Staff

3.3.1.Venue Owner

The person who will manage the admin page to enroll for his venue.

3.3.2.Accountant

The person who will control the online payments.

3.3.3.Business Manager

The person who will organize the campaigns and menu of the venue.

4.PROJECT STAFFING

There will be six types of project staff; a software project manager, a requirements engineer, a lead designer, a database administrator, a Flutter developer, a test administrator.

- **Software Project Manager:** He will have the responsibility of planning and scheduling the project, assessing risks, managing the team members, organizing tool selection and etc.
- **Requirements Engineer:** He will be responsible for identifying the stakeholders getting the requirements from the customers, analyzing and documenting the software requirements.
- **Lead Designer:** He is responsible for understanding the business requirements and designing a solution that will meet the business needs. There are many potential solutions that will meet the client's needs. The designer determines the best approach.
- **Database Administrator:** He is a specialist that models, designs and creates the databases and tables used by a software solution.
- **Flutter Developer:** He is the developer that provides the needed software applications.
- **Test Administrator:** He ensures that the solution meets the business requirements and that it is free of errors and defects.

5.SOFTWARE PROCESS MODEL

The Scrum model suggests that projects progress via a series of sprints. In keeping with an agile methodology, sprints are timeboxed to no more than a month long, most commonly two weeks.

Scrum methodology advocates for a planning meeting at the start of the sprint, where team members figure out how many items they can commit to, and then create a sprint backlog – a list of the tasks to perform during the sprint.
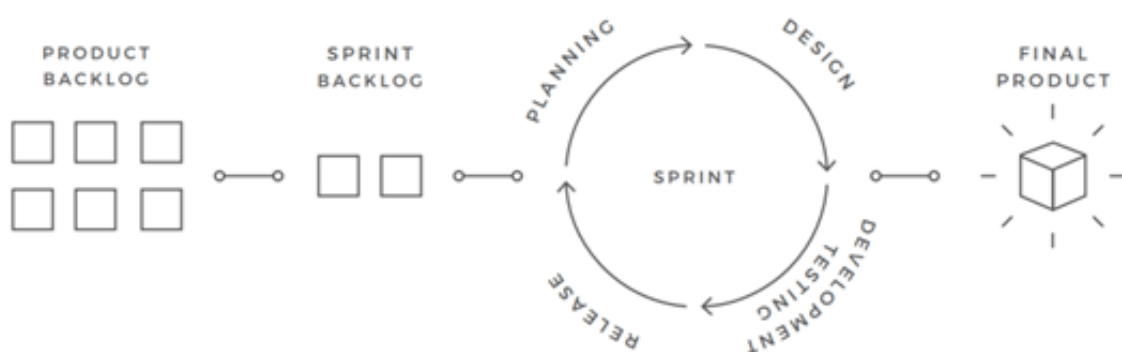
The primary artifact in Scrum development is, of course, the product itself. The Scrum model expects the team to bring the product or system to a potentially shippable state at the end of each Scrum sprint.

The product backlog is another artifact of Scrum. This is the complete list of the functionality that remains to be added to the product. The product owner prioritizes the backlog so the team always works on the most valuable features first.
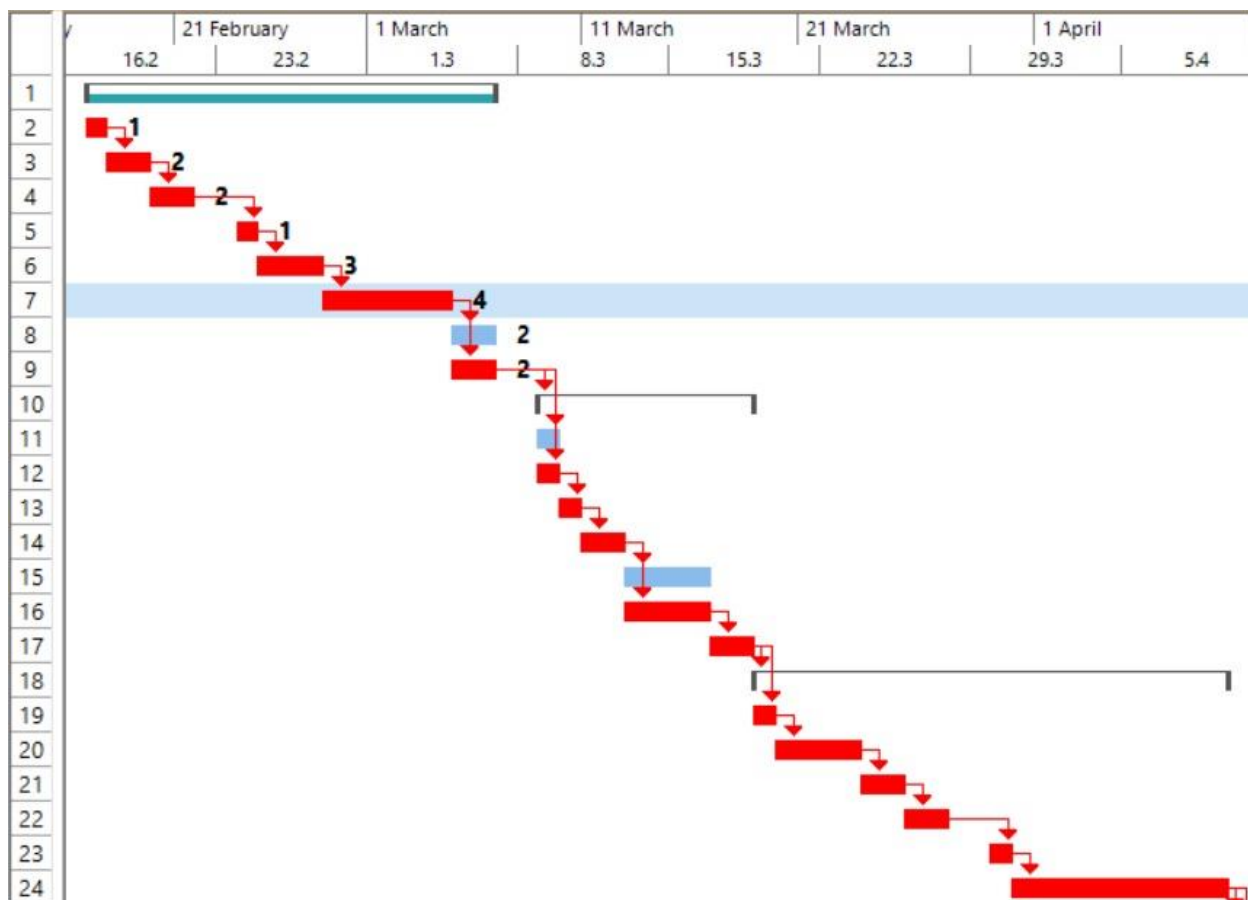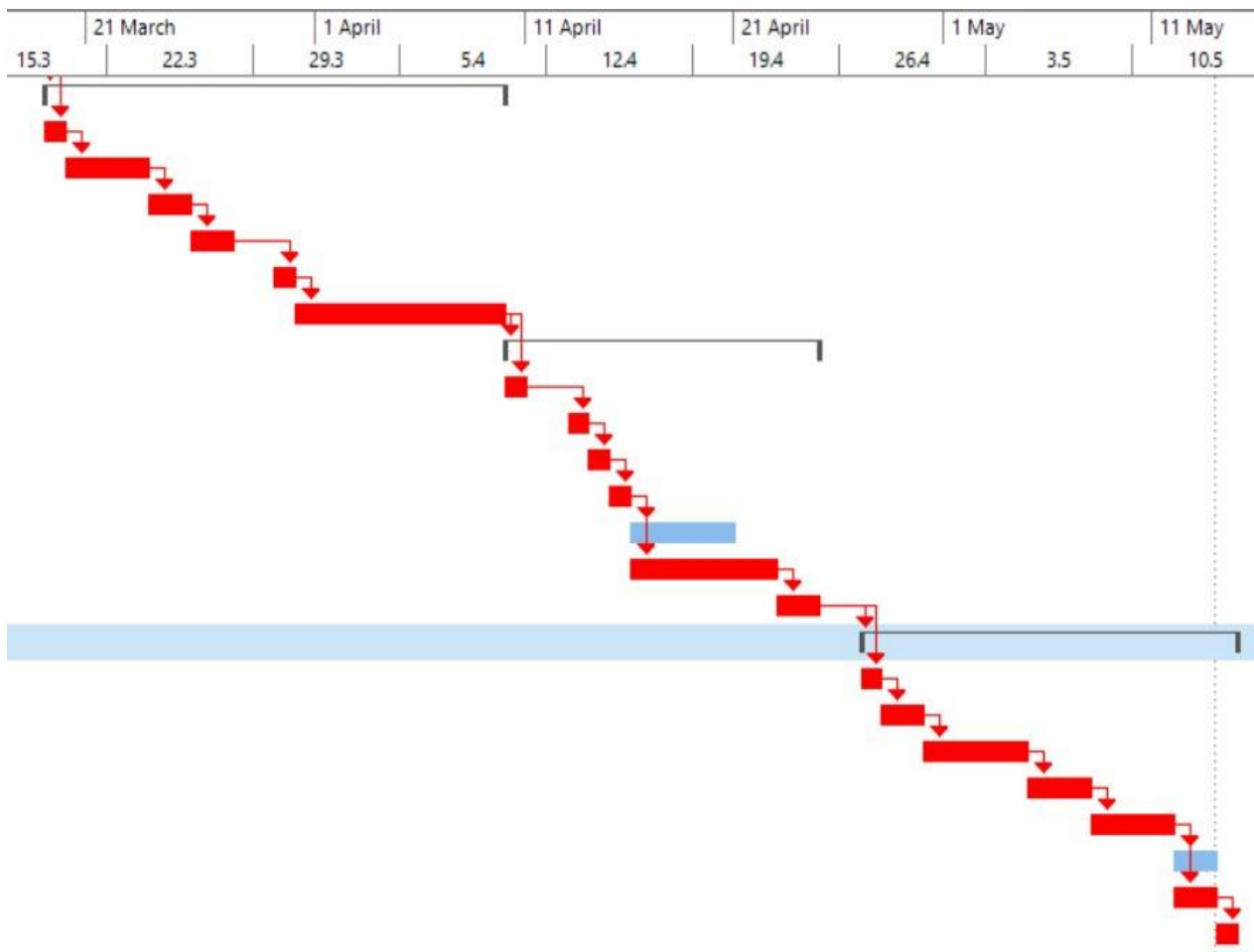
6.PROJECT SCHEDULE AND EFFORT

In software project management, a schedule consists of a list of a project's terminal elements with intended start and finish dates. It is the most crucial part of a software development process. Each date must be reasonable and included with delay times. A good schedule will probably lead to a quickly completed and qualified product.

Detailed project schedule of the project is given below;

	Task Name	Duration	Start	Finish	Predece ssors
1	⁴ Sprint 1	15 days	Pzt 17.02.20 8	Cum 6.03.20 5	
2	Sprint Planning	1 day	Pzt 17.02.20 8	Pzt 17.02.20 5	
3	Prepare Questions	2 days	Sal 18.02.20 8	Çar 19.02.20 5	2
4	Arrange Stakeholder Meeting	2 days	Per 20.02.20 8	Cum 21.02.20	3
5	Conduct Meeting	1 day	Pzt 24.02.20 8	Pzt 24.02.20 5	4
6	Prepare Meeting Report	3 days	Sal 25.02.20 8	Per 27.02.20 5	5
7	Analyze Requirements	4 days	Cum 28.02.20	Çar 4.03.20 5:	6
8	Tool Selection	2 days	Per 5.03.20 8:	Cum 6.03.20 5	7
9	Development Environment	2 days	Per 5.03.20 8:	Cum 6.03.20 5	7
10	⁴ Sprint 2	8 days	Pzt 9.03.20 8:1	Çar 18.03.20 5	9
11	Sprint Planning	1 day	Pzt 9.03.20 8:1	Pzt 9.03.20 5:1	9
12	Arrange Team Meeting	1 day	Pzt 9.03.20 8:1	Pzt 9.03.20 5:1	9
13	Conduct Meeting	1 day	Sal 10.03.20 8	Sal 10.03.20 5	12
14	Prepare Meeting Report	2 days	Çar 11.03.20 8	Per 12.03.20 5	13
15	UML Class Diagram	2 days	Cum 13.03.20	Pzt 16.03.20 5	14
16	UML Activity Diagram	2 days	Cum 13.03.20	Pzt 16.03.20 5	14
17	Architectural Design	2 days	Sal 17.03.20 8	Çar 18.03.20 5	16
18	⁴ Sprint 3	16 days	Per 19.03.20 8	Per 9.04.20 5:	17
19	Sprint Planning	1 day	Per 19.03.20 8	Per 19.03.20 5	17
20	Arrange Stakeholder Meetings	2 days	Cum 20.03.20	Pzt 23.03.20 5	19
21	Conduct Meetings	2 days	Sal 24.03.20 8	Çar 25.03.20 5	20
22	Prepare Meeting Report	2 days	Per 26.03.20 8	Cum 27.03.20	21
23	Review Requirements	1 day	Pzt 30.03.20 8	Pzt 30.03.20 5	22
24	Implementation	8 days	Sal 31.03.20 8	Per 9.04.20 5:	23

25	◢ Sprint 4	11 days	Cum 10.04.20	Cum 24.04.20	24
26	Sprint Planning	1 day	Cum 10.04.20	Cum 10.04.20	24
27	Arrange Team Meeting	1 day	Pzt 13.04.20 8	Pzt 13.04.20 5	26
28	Conduct Meeting	1 day	Sal 14.04.20 8	Sal 14.04.20 5	27
29	Prepare Meeting Report	1 day	Çar 15.04.20 8	Çar 15.04.20 5	28
30	Graphical Design	3 days	Per 16.04.20 8	Pzt 20.04.20 5	29
31	Implementation (Cont)	5 days	Per 16.04.20 8	Çar 22.04.20 5	29
32	Unit Testing	2 days	Per 23.04.20 8	Cum 24.04.20	31
33	◢ Sprint 5	14 days?	Pzt 27.04.20 8	Per 14.05.20 5	32
34	Sprint Planning	1 day	Pzt 27.04.20 8	Pzt 27.04.20 5	32
35	Arrange Stakeholder Meetings	2 days	Sal 28.04.20 8	Çar 29.04.20 5	34
36	Conduct Meetings	3 days	Per 30.04.20 8	Pzt 4.05.20 5:(	35
37	Prepare Meeting Reports	3 days	Sal 5.05.20 8:(	Per 7.05.20 5:	36
38	Optimization of Implementation	2 days	Cum 8.05.20 8	Pzt 11.05.20 5	37
39	System Testing	2 days	Sal 12.05.20 8	Çar 13.05.20 5	38
40	User Testing	2 days	Sal 12.05.20 8	Çar 13.05.20 5	38
41	Release the Product	1 day?	Per 14.05.20 8	Per 14.05.20 5	40

	21 March		1 April		11 April		21 April		1 May		11 May
15.3	22.3	29.3	5.4	12.4	19.4	26.4	3.5	10.5

7.MEASUREMENTS

There is a number of different software measurements needed to be included in this project.

- **Schedule and Effort**: This is given in detail at schedule and efforts section already.
- **Number of Changes**: Requirements and some design properties may change during the development process. It is needed to be tracked to measure the quality of the work.
- **Defect Count**: Defects are natural consequences of all projects. Defects should be fixed whenever they are found in the system. The number of these defects will show how robust the system.

8.PROJECT RISKS

Project risks are going to be analyzed with combined risk list.

8.1.Likelihood Risk List

LIKELIHOOD RANK	RISK DESCRIPTION
1	**Schedule-** the project schedule is too constricted to develop such a project
2	**Design Complexity -** the team has little experience with the deployment platform and communication protocol
3	**Requirements Volatility** - requirements canbe changed during the development cycle when requested from customer.
4	**Tools -** the team must learn about new configuration management tools
5	**Acquisition of Hardware -** deployment hardware must be purchased and installed to test the software product

8.2.Impact Risk List

IMPACT RANK	RISK DESCRIPTION
1	**Acquisition of Hardware -**deployment hardware must be purchased and Installed to test the software product
2	**Tools -** the team must learn about new configuration management tools
3	**Design Complexity** - the team has little experience with the deployment platform and communication protocol
4	**Requirements Volatility -** requirements can be changed during the development cycle when requested from customer.
5	**Schedule -** the project schedule is too constricted to develop such a project

8.3.Combined Risk List

LIKELIHOOD RANK	IMPACT RANK	COMBINED RANK	RISK DESCRIPTION
2	3	5	**Design Complexity -** the team has little experience with the deployment platform and communication protocol.
5	1	6	**Acquisition of Hardware –** deployment hardware must be purchased and installed to test the software product.
1	5	6	**Schedule -** the project schedule is too constricted to develop such a project.
4	2	6	**Tools -** the team must learn about new configuration management tools.
3	4	7	**Requirements Volatility** - requirements can be changed during the development cycle when requested from customer.

9.SOFTWARE TOOLS

Software tools are chosen by using a comparison graph. As an example for one of the chosen tools are as follows;

9.1.Software Tool for Map

9.1.1. Tool Cost/Training/Functionality Data

Tool	Google Maps	Yandex Maps	HERE WeGo Maps	MapFactor
Cost	Free(for below $200)	free	free	Free
Training Days	3	3	2	2
Functionality	80	60	50	40

9.1.2. Normalized Tool Cost/Training/Functionality Data

Tool	Google Maps	Yandex Maps	HERE WeGo Maps	MapFactor
Cost	0	0	**0**	0
Training Days	100	100	**66**	66
Functionality	100	75	**62,5**	50

9.1.3.Normalized Tool Graph
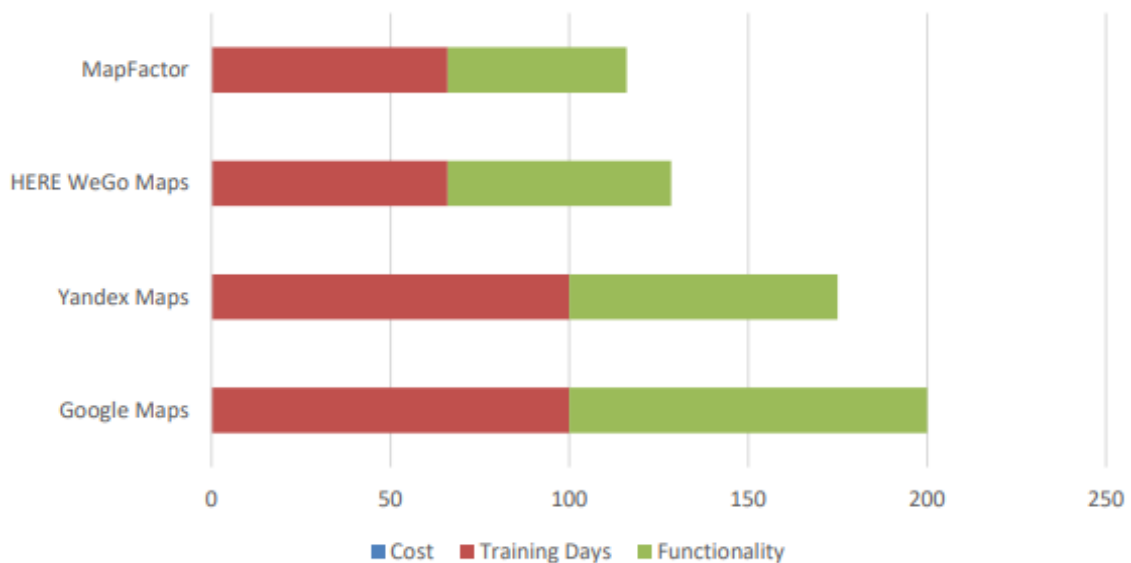
The selected tool for map is Google Maps because even all of the options are cost free but Google Maps band widht is larger than the others. Therefore, this functionality can make easier for us to use it, especially on android and web platforms.

9.2.Software Tool for Database

9.2.1. Tool Cost/Training/Functionality Data

Tool	Microsoft SQL	MySQL	Oracle	MongoDB
Cost	$900	free	$20000	free
Training Days	30	20	40	30
Functionality	60	58	65	58

9.2.2. Normalized Tool Cost/Training/Functionality Data

Tool	Microsoft SQL	MySQL	Oracle	MongoDB
Cost	4,5	0	**100**	0
Training Days	75	50	**100**	75
Functionality	92,3	90	**100**	90

9.2.3. Normalized Tool Graph
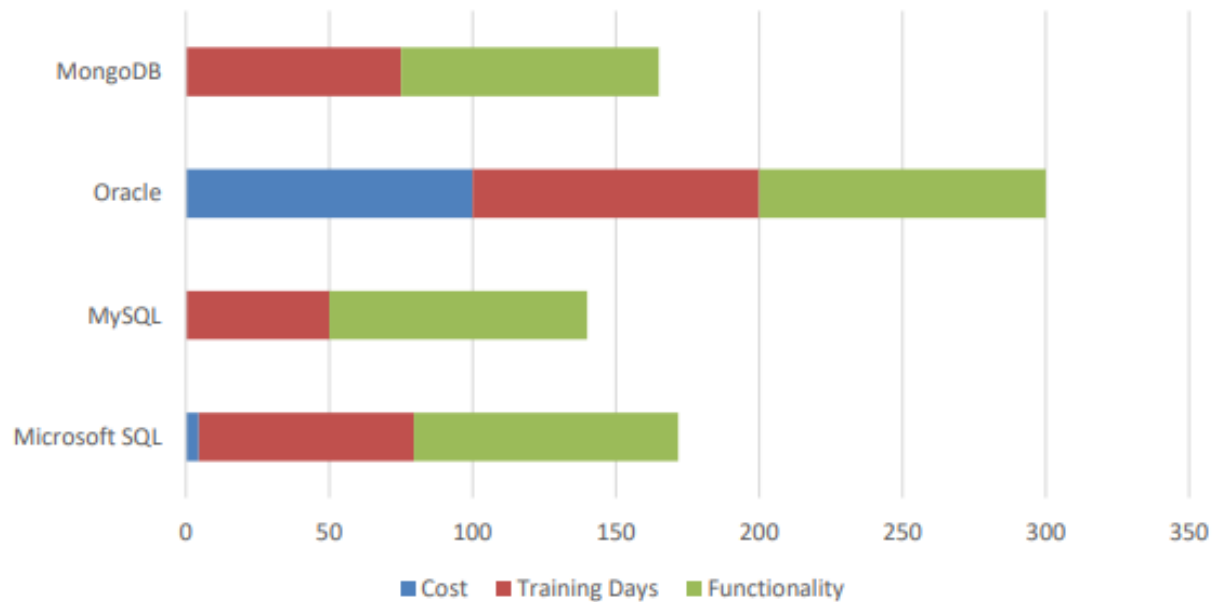
The selected tool is MySQL because it is cost free, we dont need to spent money from our limited budget. It has less training days even the functionality are the same with another tool, and this will save us time.

9.3.Software Tool for Implementation

9.3.1. Tool Cost/Training/Functionality Data

Tool	VS Code	Atom	Sublime Text	Visual Studio
Cost	free	free	$80	$1200
Training Days	5	3	3	10
Functionality	70	60	70	80

9.3.2. Normalized Tool Cost/Training/Functionality Data

Tool	VS Code	Atom	Sublime Text	Visual Studio
Cost	0	0	6,6	100
Training Days	50	30	30	100
Functionality	87,5	65	87,5	100

9.3.3. Normalized Tool Graph
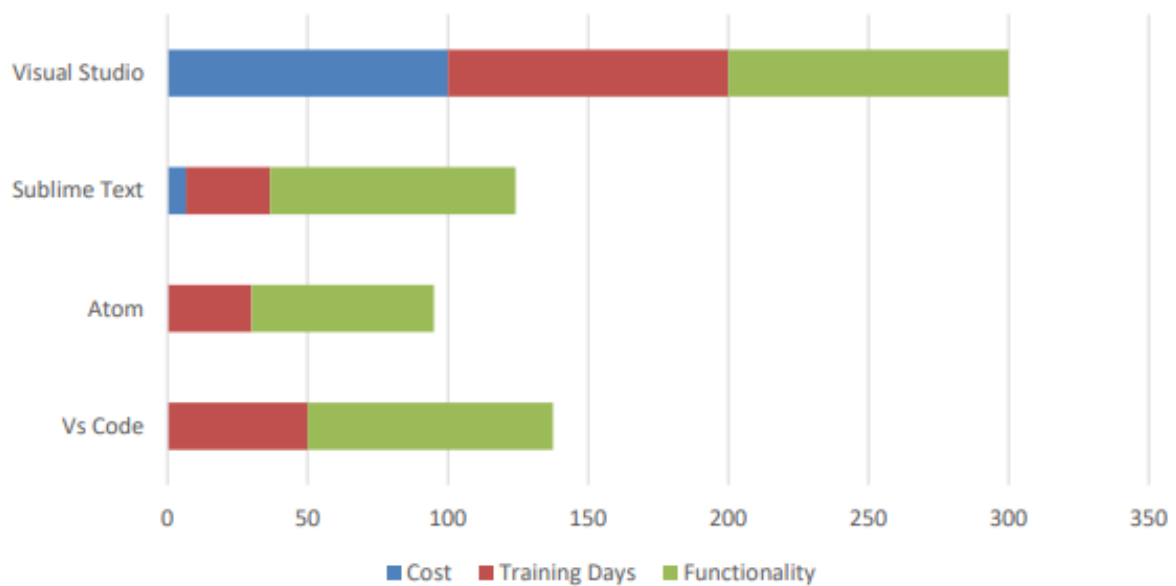
The selected tool is VS Code because it is cost free, so we can save money. Also it has highest functionality. Besides, although it is not the tool with the least training days, it will not force us.

10.PROJECT NEEDS

10.1.Hardware Needs

- **Mobile Phone:** Since the project is a mobile application, we need an Android mobile phone to implement our codes. Mobile phone should have developer option, at least 4" screen, and Android 4.0.0 and later versions.
- **Computer:** To implement the project code, we need a computer which has at least i5 model CPU, 13" screen, 8 GB RAM.
- **Cash Register**
- **Pos Device**

10.2.Software Needs

- **Flutter:** Since the product is going to be a mobile application, flutter is the best alternative to create it.
- **VS Code:** To implement flutter, the latest version of VS Code is used.
- **MySQL & MySQL Workbench:** To keep database information and to work on it, MySQL products have been choosen.
- **Mobile Payment API**

10.3.Support Needs

- **Bank's Technical Staff:** For synchronization of pos device in its every update, we need the technical staff from the bank.
- **Maintenance Team:** The system may give errors and to fix them we provide a maintenance service.
- **Configuration Team:** To follow up the new versions of tools and to make the system current we need a configuration team.

11.GRAPHICAL USER INTERFACES

12. CONCLUSION

In this document, everything needed to know about this project is explained in detail. This will be very helpful to the project manager if it is followed during development.