

İZMİR UNIVERSITY OF ECONOMICS
FACULTY OF ENGINEERING
SE 216 - Software Development Plan



Seat Tracker System

Authors:

Ejder Aysun
Göktuğ Gürler
Miray Solmaz
Şenay Selin Ünal
Tareq Alhammoodi

10.05.2023

1. Overview-----	3
1.1 Introduction-----	3
1.2 Goals-----	3
2. High-Level Functionality -----	3
2.1 Functional Requirements -----	3
2.2 Non-Functional Requirements -----	4
3. Stakeholders -----	4
4. Project Staffing-----	5
5. Software Process Model -----	5
5.1 Needs From The Organizational Process -----	5
5.1.1 Necessary Needs From The Organizational Process -----	5
5.1.2 Unnecessary Needs From The Organizational Process -----	6
5.2 Software Development Model -----	6
5.2.1 Software Process Description -----	6
5.2.2 Software Process Model -----	7
5.2.3 Reasons To Choose This Model-----	7
6. Schedule and Effort-----	7
7. Measurements -----	8
7.1 Specifying Measurements -----	8
7.2 Example Measurements-----	9
8. Project Risks -----	10
8.1 Likelihood Risk List -----	10
8.2 Impact Risk List -----	11
8.3 Combined Risk List-----	12
9. Software Tools -----	13
9.1 Graphic User Interface (GUI) Design -----	13
9.2 Integrated Development Environment (IDE)-----	14
9.3 Collaboration And Communication-----	15
10. Project Needs-----	16
10.1 Software Needs -----	16
10.2 Hardware Needs -----	16
10.3 Support Needs-----	17
11. Graphical User Interface -----	17
12. Conclusion -----	20

1. Overview

1.1 Introduction

In today's bustling world, cafes and restaurants have become more than just places to enjoy drinks or food in. They have become places where people can socialize, meet, work, or just relax. However, with the increasing popularity of cafes, finding a seat might not be that easy. It's not uncommon thing to spend time wandering from table to table, searching for a place, only to discover that all the seats are occupied. This inconvenience can affect the overall experience of cafe-goers.

In our project, we want to develop a system that can make this process easier for customers. With a quick glance at a display mounted on the wall of the cafe, customers can instantly see which tables are occupied and which ones are vacant. This saves valuable time for them that would otherwise be spent on aimlessly searching for a free place. Based on that information customer will be able to decide where to sit.

1.2 Goals

1. Create a seat availability tracker system that is easy to use for customers and displays table availability in real-time.
2. Develop a simple and attractive interface that is not difficult to navigate and provides customers with the information they need to make informed decisions.
3. Improve customer satisfaction by reducing waiting times.
4. Help cafes and restaurants improve their operations by providing a valuable tool for managing table availability and optimizing space utilization.

2. High-Level Functionality

2.1 Functional Requirements

1. The system shall identify table status by colours, red for occupied and green for vacant.
2. The system must display the number of vacant and occupied tables.
3. The system should display the number of floors and outdoor places like gardens, seaward side or balconies (if any) with their tables.
4. The system should display the total number of customers per day.
5. The device should be responsible for changing the table status automatically.

6. The system should display all the data to customers including seats and their status and total number of occupied/vacant seats on a screen connected to it.

2.2 Non-Functional Requirements

1. The system should support Turkish and English languages.
2. The system should have a simple interface that is easy to navigate by the customers.
3. The system should be easy to maintain and update without disrupting the user experience.
4. The system shall be available 99.99% of the time during business hours.
5. The system shall be intuitive and easy to use for users with basic computer skills, without requiring extensive training or support.
6. The system should be connected to a device that help with taking orders and tracking the sets (by giving it to the customer).

3. Stakeholders

1. **Competitors:** Competitors can offer similar seat tracker systems, and this can lead to market share competition. They might also gain benefits from our project's failure. However, competition can be a reason for developing a better product and offering new features.
2. **Customers:** These stakeholders are important because the main goal of our project is to make customers experience unique and easy that meets their requirements, and the service will be thoroughly tested to ensure it meets their expectations.
3. **Detractors:** These stakeholders can also play a role by discouraging restaurants from using the system. This can happen if they have concerns about its reliability or accuracy. They can spread negative feedbacks about it, which can impact the restaurant's reputation and lead to decreased adoption of the system.
4. **Developers:** There are several benefits that developers can get from project success, including increased job security, improved reputation, personal satisfaction and financial rewards which are represented by bonuses, profit-sharing, stocks, or other financial incentives.
5. **Government:** The government is a key stakeholder in any project, and its role and impact cannot be underestimated which is represented by its benefits from taxes and costs.
6. **Restaurant Owners:** They are the primary stakeholders who own and operate the restaurant and responsible for ensuring that the system meets their business needs.
7. **Sponsor and investors:** These stakeholders provide the financial resources needed to start and grow a restaurant. They may use the seat tracker system in that restaurant.

8. **Restaurant Staff:** These stakeholders are the ones who are going to use the program on a nearly daily basis within their workplace. They use the seat tracker system to know which tables are occupied and which are available for seating and serve their customers accordingly.

4. Project Staffing

1. **GUI Designer:** The graphic user interface designer is responsible for working on the visual layout and the identity of the system and ensuring that the system is comprehensible and easy to use by the user.
2. **Desktop Applications Developer:** The developer is responsible for implementing a system that meet the requirements of the project. This includes designing the software architecture and coding it.
3. **Project Manager:** The project manager is responsible for defining the project scope, objectives, and deliverables. Also, creating a project plan, including requirements, needs, timelines, and resource allocation, to ensure that the project is executed effectively.
4. **Tester:** The tester is responsible for defining test objectives, and determining the appropriate testing techniques and tools to be used. Also, considering different scenarios and test conditions to ensure comprehensive test coverage.
5. **Human Resources Manager:** Human resources manager is responsible for planing, coordinating, and directing the administrative functions of an organization, and attracting skilled professionals for the software project. Also, ensuring compliance with employment laws, regulations, and company policies within the software project.
6. **Hardware Technician:** The Hardware Technician is responsible for installing and configuring hardware components required for the software project. This includes setting up networking equipment, and devices that are connected to the system.

5. Software Process Model

5.1 Needs From The Organizational Process

5.1.1 Necessary Needs From The Organizational Process

1. It is essential to prioritize the requirements that need to be developed first.
2. Important functions should be easily accessible and delivered early to the customer.
3. Developing the main system initially and incorporating minor modifications later would make it easier to resolve errors.

4. It is necessary to obtain feedback from stakeholders.
5. Changes can be made during the various phases of development.
6. The project must be designed in a way that facilitates testing and debugging.
7. It is important to be flexible and adaptable during the software development process.

5.1.2 Unnecessary Needs From The Organizational Process

1. The team may be given some more time to discuss any issues that may exist.
2. Maintenance support can be provided for hardware and software problems that occur after the project.
3. The user interface should have flashy animations and graphics to make it look more impressive, even if they do not improve the user experience.
4. Excessive Documentation.

5.2 Software Development Model

Software Process Name: Incremental Model

5.2.1 Software Process Description

The incremental model is a software development methodology that involves the iterative creation of software through multiple development cycles or increments. The software process description for the incremental model can be broken down into the following phases:

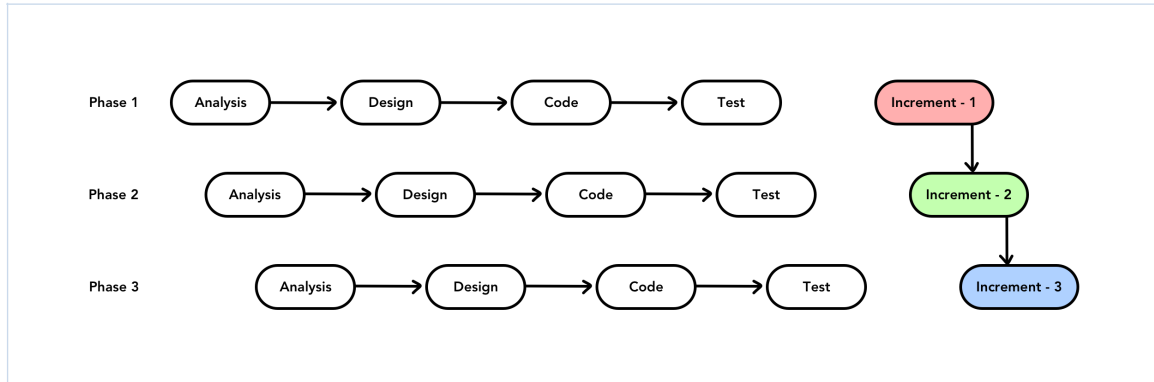
Analysis: The first step in the incremental model is to collect the system requirements and divide them into smaller, more manageable pieces that can be implemented incrementally.

Design: Next the design for the system should be created and the design process is also splitted into smaller pieces that can be implemented incrementally.

Coding: Each increment is implemented and coded as a separate module. The code will be written for each increment and tested before being integrated with the rest of the system.

Testing: Testing is done at each increment to ensure that the increment meets the requirements and specifications then bugs and errors should be handled as they arise.

5.2.2 Software Process Model



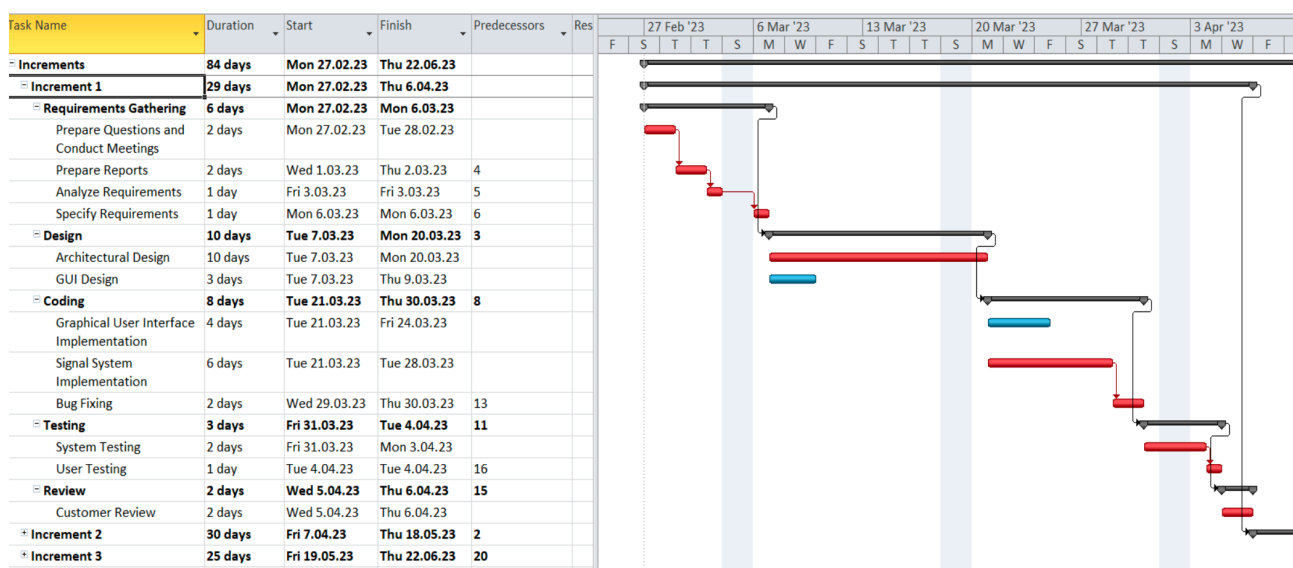
5.2.3 Reasons To Choose This Model

The main reasons why we have chosen incremental model are that:

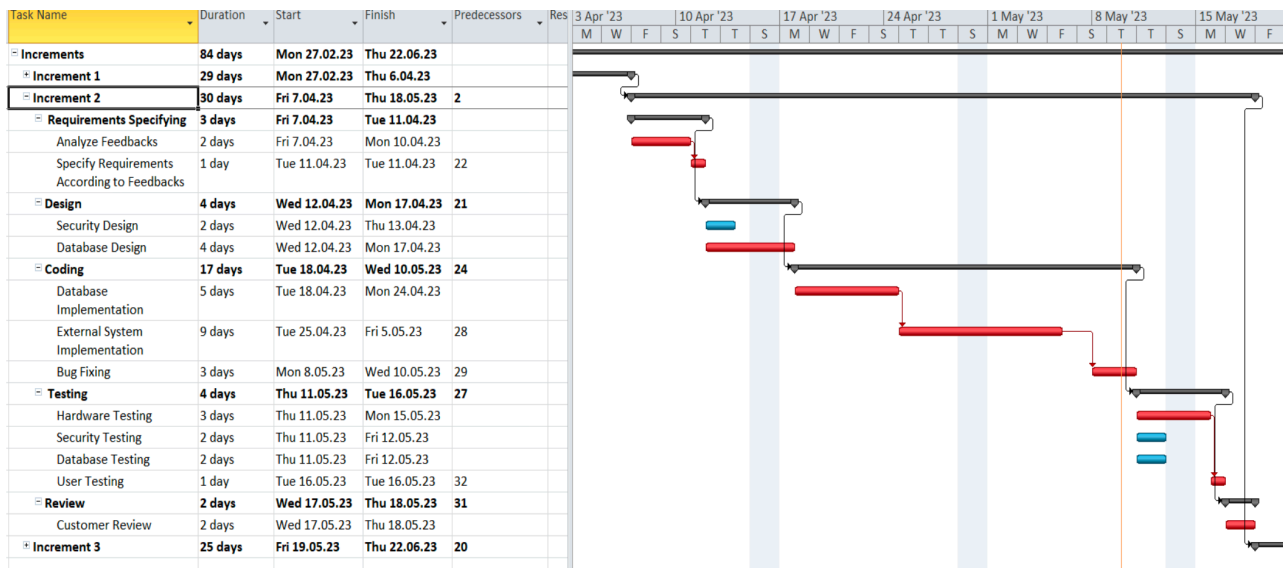
1. Stakeholder involvement is important to us so we want to ensure that they can review the product and make changes accordingly in each phase.
2. We want to notice errors easily through several testing stages.
3. Making changes easily during implementation is one of our needs.
4. Our team members are not skilled enough so we will need to modify the system occasionally.
5. We need easy testing and debugging processes.

6. Schedule and Effort

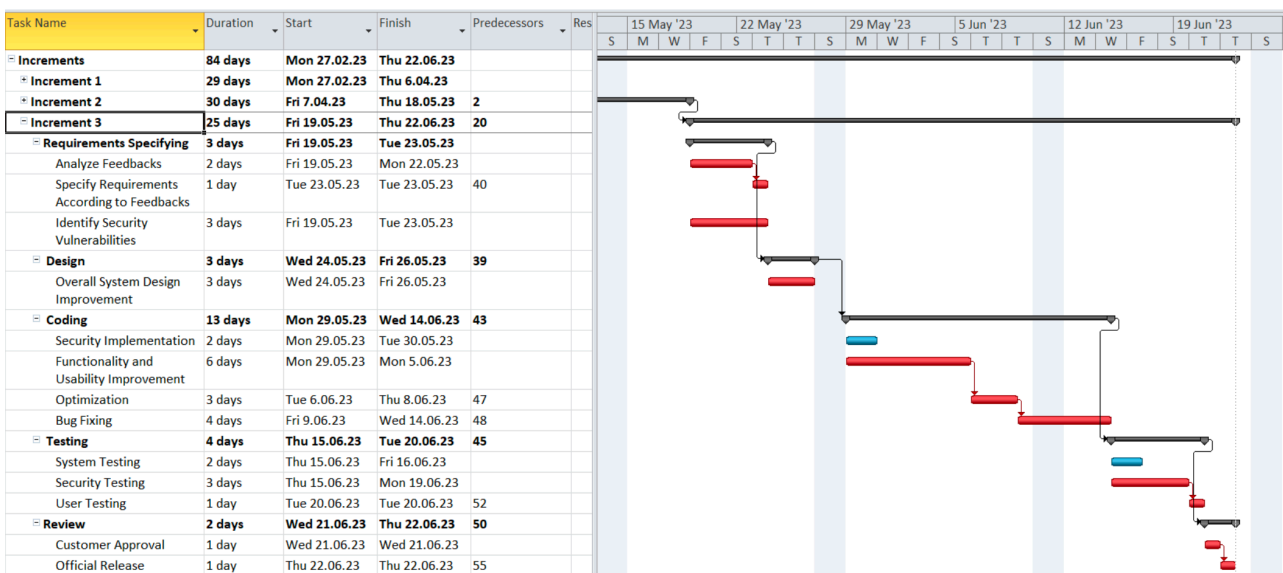
Increment 1



Increment 2



Increment 3



7. Measurements

7.1 Specifying Measurements

1. Questions to identify measurements:

- Did the project adhere to its schedule?
- How many changes has been occurred?

2. Identified measurements:

for project schedule:

- Planned vs actual schedule.
- Number of tasks completed on time.

for changes:

- Total number of changes.
- Change date.

3. Measurement storage and collection:

for project schedule:

What: The comparison between the planned schedule and the actual schedule, and milestone achievement.

When: The measurements of planned vs. actual schedule and milestone achievement will be collected throughout the project lifecycle in each increment.

Format: The measurements will be collected in a spreadsheet format as integer or float numbers to allow easy analysis and comparison of the data.

How: The responsible team member for change management will collect the measurements, and these will be stored in a centralized location for the sake of convenient accessibility and analysis.

for changes:

What: Number of change requests and date of change.

When: This measurement will be collected throughout the project lifecycle in each increment.

Format: To facilitate analysis and comparison of the data, the measurement will be stored in a spreadsheet format as integer numbers.

How: The measurement will be collected by team member responsible for change management. The data will be stored in a centralized location for easy access and analysis.

7.2 Example Measurements

1. Change Management: Evaluate the project's adherence to the planned schedule.

Ex: Planned vs. actual schedule, and number of tasks completed on time.

2. Schedule Adherence: Monitor and manage changes in the project.

Ex: Total number of changes and changing date.

3. Project Progress: Track the completion of project milestones and deliverables.

Ex: Percentage of completed milestones, number of tasks completed vs. total tasks.

4. Time Management: Monitor the time spent on tasks and compare it to the planned time.

Ex: The actual time spent on tasks vs. estimated time, team member hours logged, and the average time to complete tasks.

5. Budget Management: Ensure the project stays within the allocated budget by tracking expenses and comparing them to the initial budget.

Ex: The actual costs vs. planned costs, and cost variance.

6. Quality Control: Assess the quality of the project deliverables by tracking defects and their resolution.

Ex: Number of defects identified, percentage of defects resolved, and defect density.

8. Project Risks

8.1 Likelihood Risk List

LIKELIHOOD RANK	RISK DESCRIPTION
1	Limited Resources Risks: Factors that may restrict the development team's ability to do the project effectively, such as time, human resource and information limitations, and other resource constraints, are included.
2	Unexpected Software Failure: Software can fail for a variety of reasons, like crashes, errors and other unexpected behaviors.
3	Performance Risks: The risks associated with software quality such as poor performance, slow load time and insufficient testing.
4	Inadequate Project Management: Poor project management of a project can lead to various problems, which can disrupt the project process.
5	Hardware Risks: The device taken by the customer must work properly when connected to the table and transmit information correctly about the table's status.
6	Testing Risks: the system might be difficult to test in real life as it needs to be installed in restaurant.
7	Supplier Risks : If hardware suppliers are unable to deliver the device on time, the completion of the project may be delayed.
8	Lack of Communication: Risks associated with communication breakdowns among project stakeholders, including developers, clients, and project managers.
9	Scheduling Risks: The risks associated with project scheduling like unexpected setbacks or missed deadlines can lead to delays.
10	Cost Risks: In the development process, the budget may not be sufficient, or there may be disagreement regarding the license and maintenance costs for the restaurant where this project will be used.
11	User Adoption Risks: The system may not be adopted by restaurant staff because it's not easy to use, or because of their ages or cultural background. This issue could affect the system's overall effectiveness.

8.2 Impact Risk List

IMPACT RANK	RISK DESCRIPTION
1	Unexpected Software Failure: Software can fail for a variety of reasons, like crashes, errors and other unexpected behaviors.
2	Testing Risks : the system might be difficult to test in real life as it needs to be installed in restaurant.
3	Hardware Risks : The device taken by the customer must work properly when connected to the table and transmit information correctly about the table's status.
4	Performance Risks : The risks associated with software quality such as poor performance, slow load time and insufficient testing.
5	Supplier Risks : If hardware suppliers are unable to deliver the device on time, the completion of the project may be delayed.
6	Inadequate Project Management : Poor project management of a project can lead to various problems, which can disrupt the project process.
7	Limited Resources Risks : Factors that may restrict the development team's ability to do the project effectively, such as time, human resource and information limitations, and other resource constraints, are included.
8	Cost Risks : In the development process, the budget may not be sufficient, or there may be disagreement regarding the license and maintenance costs for the restaurant where this project will be used.
9	User Adoption Risks : The system may not be adopted by restaurant staff because it's not easy to use, or because of their ages or cultural background. This issue could affect the system's overall effectiveness.
10	Lack of Communication : Risks associated with communication breakdowns among project stakeholders, including developers, clients, and project managers.
11	Scheduling Risks : The risks associated with project scheduling like unexpected setbacks or missed deadlines can lead to delays.

8.3 Combined Risk List

LIKELIHOOD RANK	IMPACT RANK	COMBINED RANK	RISK DESCRIPTION
2	1	3	Unexpected Software Failure : Software can fail for a variety of reasons, like crashes, errors and other unexpected behaviors.
3	4	7	Performance Risks : The risks associated with software quality such as poor performance, slow load time and insufficient testing.
1	7	8	Limited Resources Risks : Factors that may restrict the development team's ability to do the project effectively, such as time, human resource and information limitations, and other resource constraints, are included.
5	3	8	Hardware Risks : The device taken by the customer must work properly when connected to the table and transmit information correctly about the table's status.
6	2	8	Testing Risks : the system might be difficult to test in real life as it needs to be installed in restaurant.
4	6	10	Inadequate Project Management : Poor project management of a project can lead to various problems, which can disrupt the project process.
7	5	12	Supplier Risks : If hardware suppliers are unable to deliver the device on time, the completion of the project may be delayed.
8	10	18	Lack of Communication : Risks associated with communication breakdowns among project stakeholders, including developers, clients, and project managers.
10	8	18	Cost Risks : In the development process, the budget may not be sufficient, or there may be disagreement regarding the license and maintenance costs for the restaurant where this project will be used.
9	11	20	Scheduling Risks : The risks associated with project scheduling like unexpected setbacks or missed deadlines can lead to delays.
11	9	20	User Adoption Risks : The system may not be adopted by restaurant staff because it's not easy to use, or because of their ages or cultural background. This issue could affect the system's overall effectiveness.

9. Software Tools

Project tasks which require software tools support:

- Graphical user interface (GUI) design.
- Integrated development environment (IDE).
- Collaboration and communication.

9.1 Graphic User Interface (GUI) Design

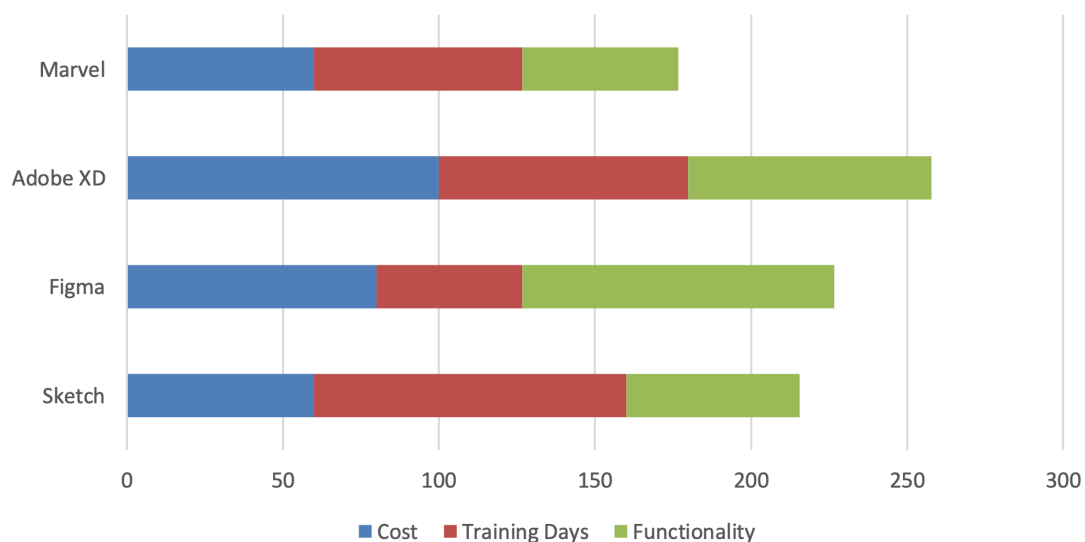
Tool cost/Training/Functionality data

Tool	Sketch	Figma	Adobe XD	Marvel
Cost	9\$ / month	12\$ / month	14.99\$ / month	9\$ / month
Training Days	15 days	7 days	12 days	10 days
Functionality	50	90	70	45

Normalized cost/Training/Functionality data

Tool	Sketch	Figma	Adobe XD	Marvel
Cost	60	80	99.93	60
Training Days	100	46.66	80	66.66
Functionality	55.55	100	77.77	50

Normalized graph



Which tool has been selected? Why?

After considering the chart, we decided to use Figma because it is the most functional tool that can be learned in the shortest amount of time. Although it is not cheap, however, we believe that the usability and functionality of Figma make it worth the investment. Therefore, we have decided to go with Figma for our user interface design needs.

9.2 Integrated Development Environment (IDE)

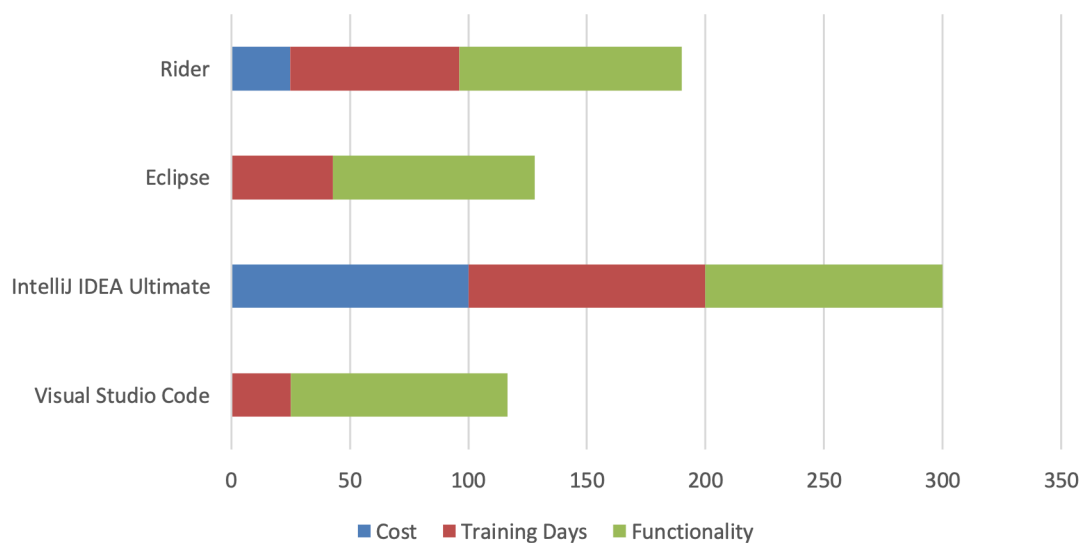
Tool cost/Training/Functionality data

Tool	Visual Studio Code	IntelliJ IDEA Ultimate	Eclipse	Rider
Cost	0\$ / month	59.90\$ / month	0\$ / month	14.90\$ / month
Training Days	7 days	28 days	12 days	20 days
Functionality	86	94	80	88

Normalized cost/Training/Functionality data

Tool	Visual Studio Code	IntelliJ IDEA Ultimate	Eclipse	Rider
Cost	0	100	0	24.87
Training Days	25	100	42.86	71.43
Functionality	91.49	100	85.11	93.62

Normalized graph



Which tool has been selected? Why?

We chose Visual Studio Code because our team has prior experience in coding and developing applications using this IDE. It is easy to learn and has comparable performance to other IDEs. We decided on this IDE as our familiarity with it will save us the cost of learning a new one and any potential monthly fees associated with it.

9.3 Collaboration And Communication

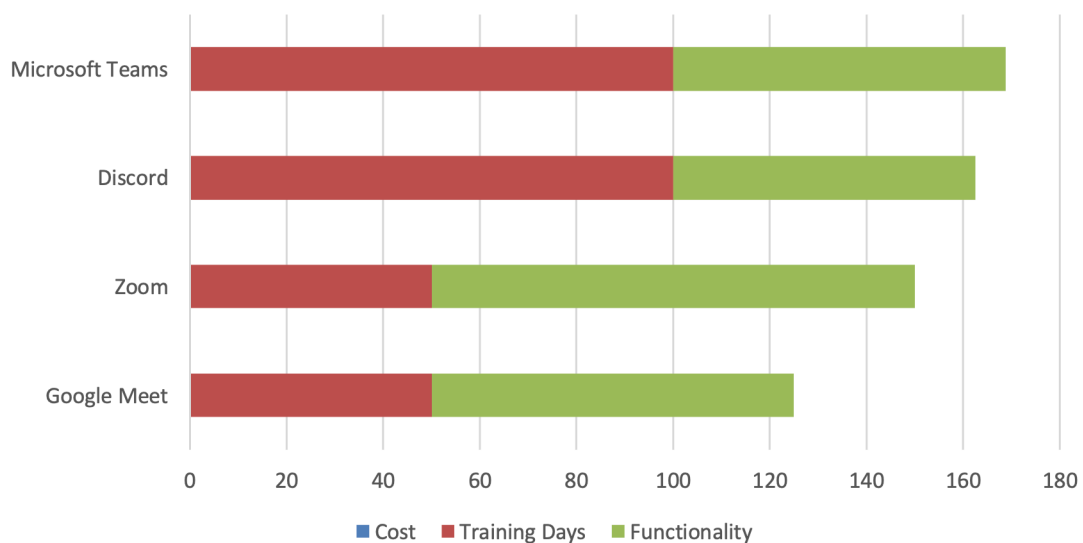
Tool cost/Training/Functionality data

Tool	Google Meet	Zoom	Discord	Microsoft Teams
Cost	0\$ / month	0\$ / month	0\$ / month	0\$ / month
Training	1	1	2	2
Functionality	60	80	50	55

Normalized cost/Training/Functionality data

Tool	Google Meet	Zoom	Discord	Microsoft Teams
Cost	0	0	0	0
Training Days	50	50	100	100
Functionality	75	100	62.5	68.75

Normalized graph



Which tool has been selected? Why?

We have agreed to use Discord, although there may be better and easier alternatives. However, our team is quite familiar with using this app as everyone has at least 2 years of experience with Discord. Despite its minimal functionality, our knowledge of Discord is sufficient to bridge this gap. Additionally, using a new collaboration and communication application would incur a time cost, which we can avoid by sticking with Discord.

10. Project Needs

10.1 Software Needs

1. **Operating Systems:** The system must be supporting Windows 10 or higher and MacOS 10.13 or higher.
2. **Integrated Development Environment (IDE):** The development team will use Visual Studio Code as the primary IDE 1.64 or higher version is recommended.
3. **Collaboration and Communication Tools:** To facilitate collaboration and communication between team members working on the system, the development team will use Discord for this purpose (14.10.2 or higher).
4. **Programming Languages:** The system will be built using C++ (ISO/IEC 14882:2020) for communicating with hardware, JavaScript (ECMAScript 2021) for user interface, Node.js (18.14.0(LTS)) to create server-side running application, and MQTT (5.0) for real-time communication protocols.
5. **Documentation:** The system documentation will be created using Google Docs to track the process easily and work on the same document simultaneously, and stored in GitHub.
6. **Database Management:** MySQL (version 8.0.31) and SQL (16.0.1000.6) will be used to store and manage data such as orders and tables status.
7. **User Interface (UI) library:** The system will use a UI library such as Riot.js (v 7.1.0) to create a responsive and simple interface for the restaurant staff and customers.

10.2 Hardware Needs

1. **Interactive Device:** The system will be connected to a device that updates the table status as occupied or vacant.
2. **Screen:** A high-resolution display with a minimum of 32-inch screens are required to display tables status to the customers.
3. **Networking Equipment:** The system will require a WiFi network that can be set up to provide local connectivity between devices with no necessity of an internet connection.
4. **Computer:** The development team will require computers with at least 8 GB RAM and Intel Core i5 processors to work on the project, on the other hand, computers for the system to operate and for employees to manually track and control the system are also needed.
5. **Backup Power Source:** A backup power source such as an uninterruptible power supply (UPS) will be required to prevent data loss in the event of power outages.

10.3 Support Needs

1. **Technical Support:** When problem occurs, IT support can be requested from specialized experts in different areas, such as network, software and hardware, to provide assistance.
2. **Staff Training Support:** Restaurant staff should receive appropriate training on how the system works and be prepared for future updates by experts in the field.
3. **Integration Support:** To prevent possible errors, the "Seat Tracker" system must be compatible with the other programs used in the restaurant.
4. **Maintain Support:** Regular maintenance and updates are necessary to ensure the system's performance is optimized.
5. **Owner Support:** Feedbacks that are given by restaurant owners are really important. These feedbacks are used to improve the system and update it.
6. **Quality Assurance Support:** Ensuring that the equipment and software meet the necessary qualifications.
7. **Security Support:** Security measures to protect system data such as customers numbers and orders.

11. Graphical User Interface

1. Login Page

SeatTracker

Welcome Back, Please login to your account.

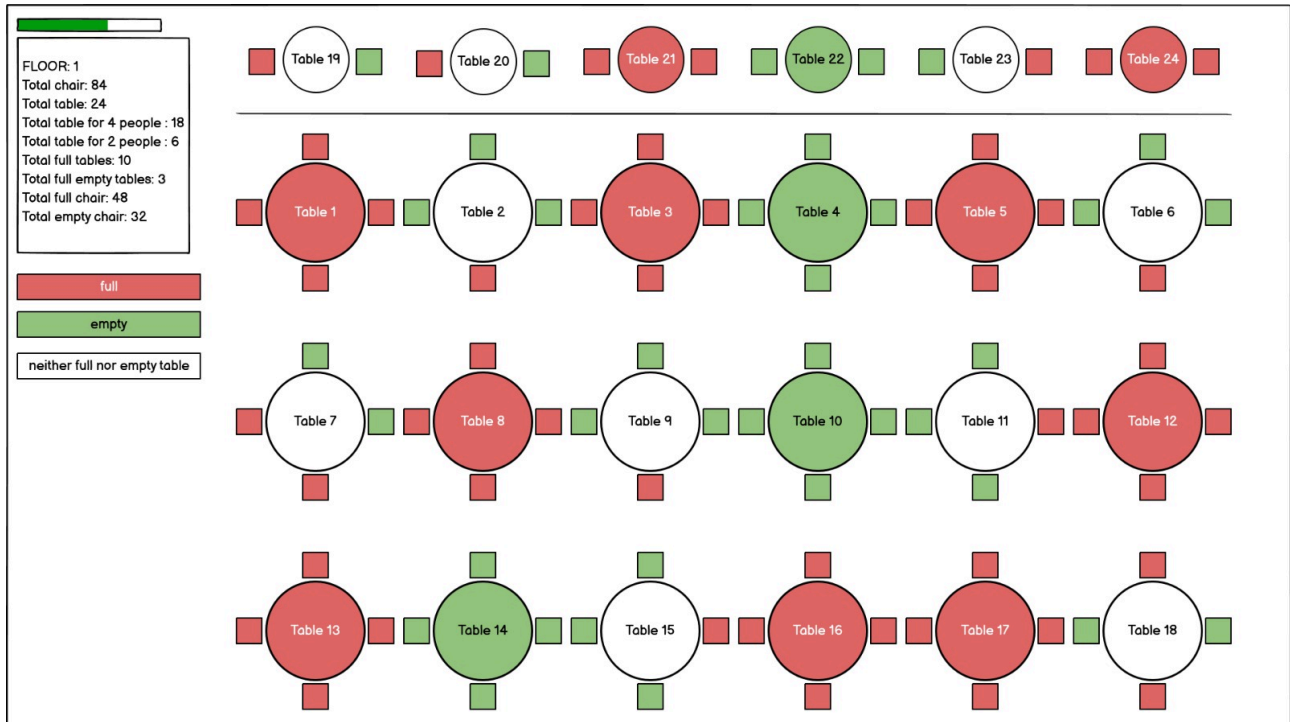
Username
write username here...

Password
write password here...

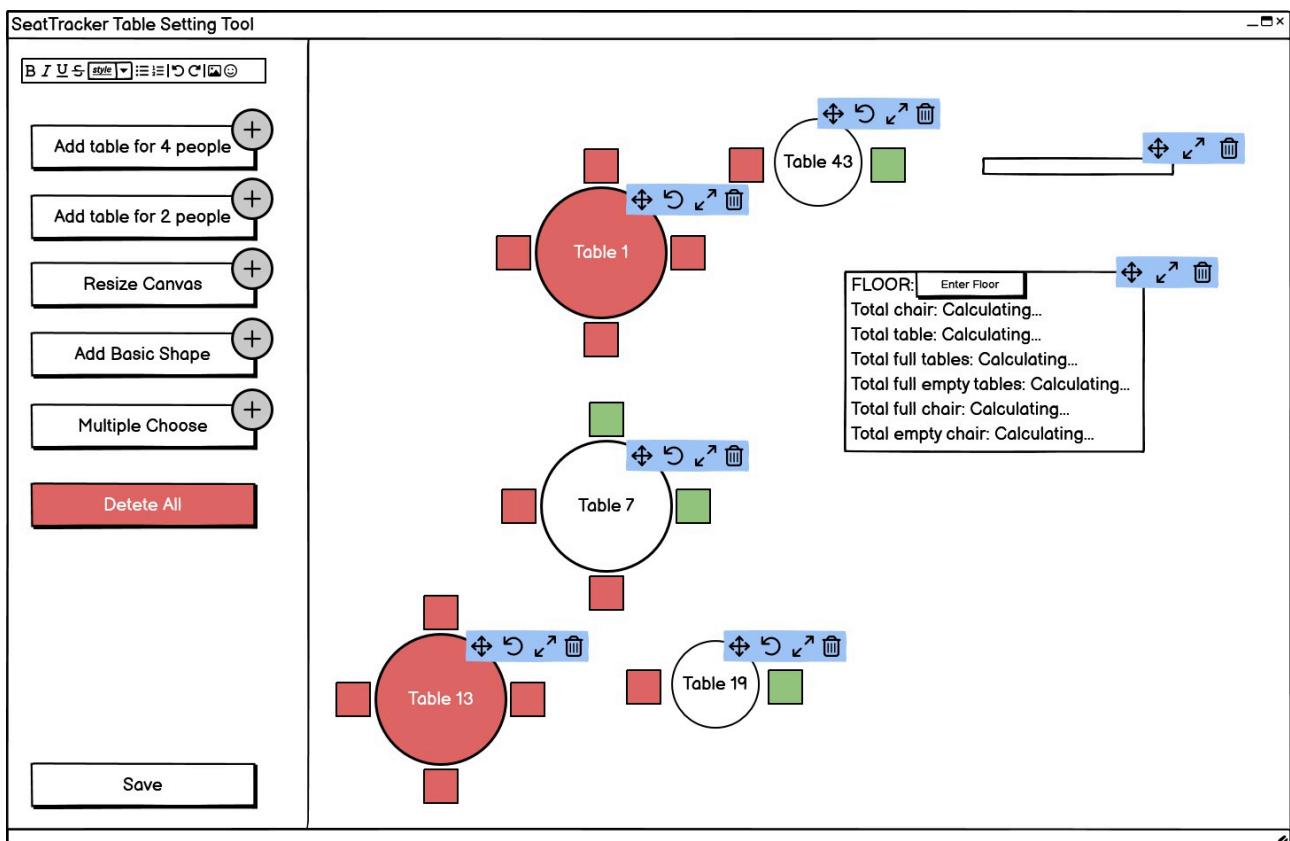
☐ Remember me

[Forgot Password](#)

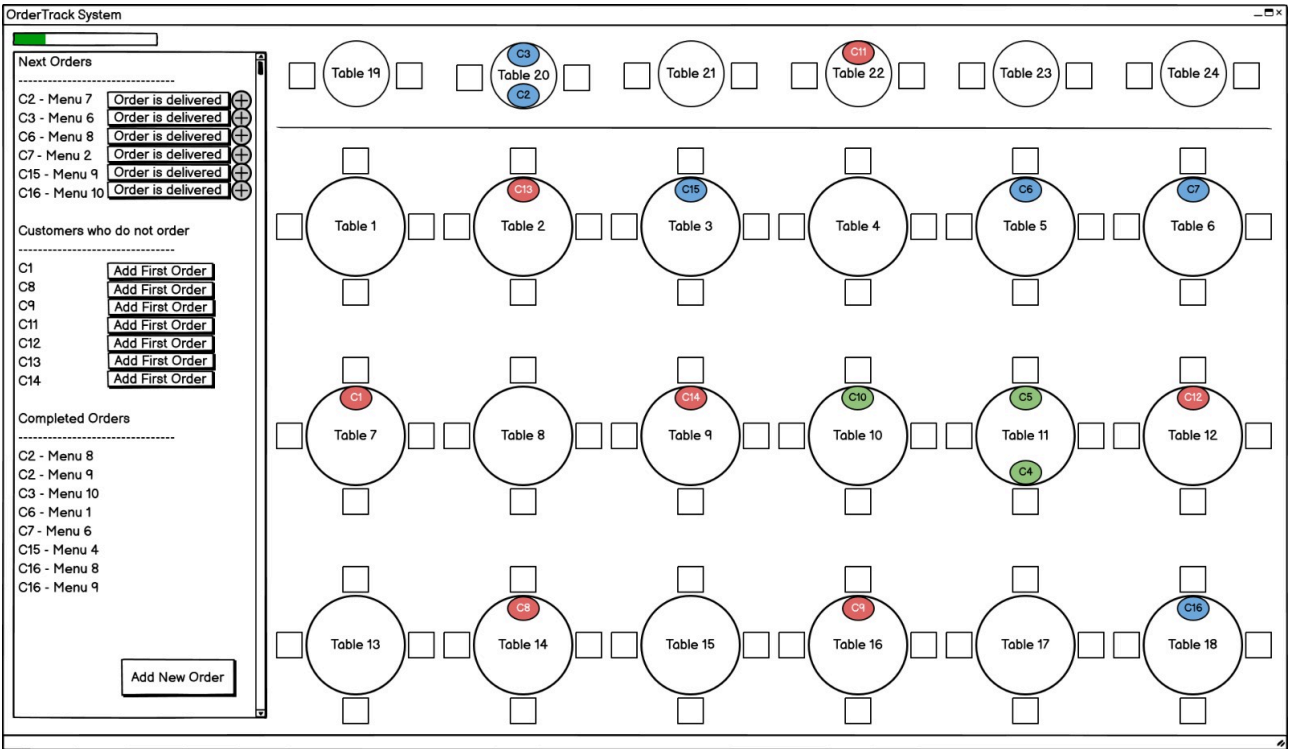
2. Main Page



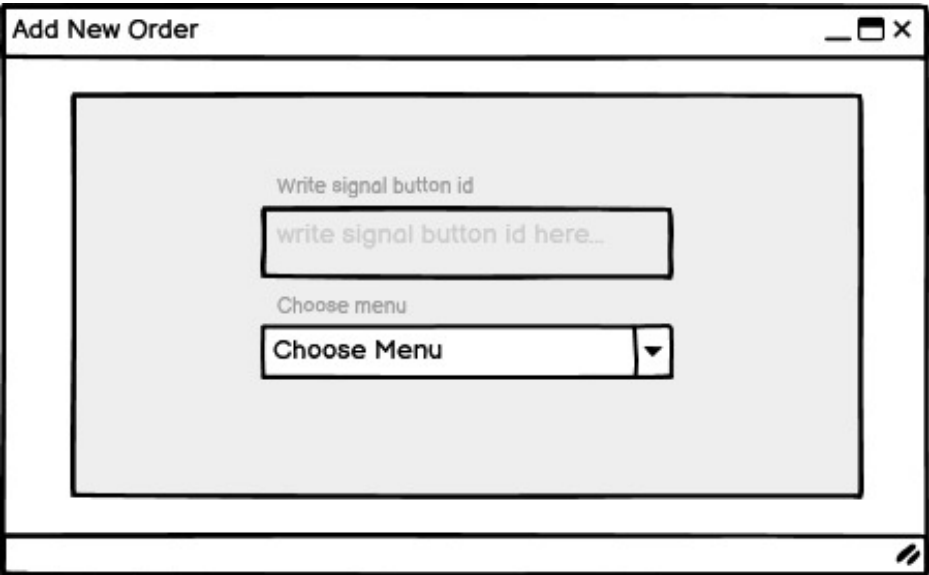
3. Settings



4. Orders Tracing Page



5. Add New Order



12. Conclusion

In conclusion, the main purpose of this project is to make the process of finding a place easier for customers in cafes and restaurants. Our project has been developed according to our plan. First, the requirements were determined, and a decision was made to use the incremental model due to its ability of being broken down into multiple phases, which will help in optimizing our plan. Next, all the required software tools that will be used were determined. Following that, the measurements were specified, a risk analysis was conducted for the project and we analyzed stakeholders whether they will benefit from our project or not. Then, the software, hardware, and support needs were specified, and a schedule for the project was created, including a graphical user interface (GUI) design. Finally, the code of the project was implemented.