**Project Report**

|  |  |
| --- | --- |
| **Product Name** | Applied Degree in Software Engineering (BDSE) |
| **Qualification Name** | Applied Degree in Software Engineering/ Higher Diploma in Software Engineering |
| **Project title** | Meals on Wheels Web Application |
| **Module Name (BDSE)** | Develop Enterprise Applications |

**Meals on Wheels Web Application**

**Project title**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name** | | **Assessor name** | |
| Syukur Sidiq Nur Alam | | Arvinder kaur | |
| **Date issued** | **Completion date** | | **Submitted on** |
|  |  | |  |

Date:

Student signature:

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

**Learner declaration**

**MEALS ON WHEELS**

**Project Report**

****

**SOFTWARE ENGINEERING**

**Group-07:**

**Asep Supriyadi (L)**

**Abdul Rahman Shalehudin**

**Ajri Muhammad Sidik**

**Darren Farrell Andrian**

**Syukur Sidiq Nur Alam**

**LITHAN EDUCLAAS**

**(2023)**

**Table of Contents**

1. **Background**

The project "Meals on Wheels" aims to provide a comprehensive software solution for MerryMeal, a charitable organization dedicated to delivering nutritious meals to homebound individuals who are unable to cook for themselves due to age, disease, or disability. In collaboration with Unity One Solutions, the project will encompass designing the software, creating a development plan, and implementing a functional application. The service will operate from Monday to Friday, with frozen meals provided over the weekend for members located beyond a 10-kilometer radius. By partnering with food service providers nationwide, MerryMeal seeks to ensure efficient and prompt meal delivery to those in need. As a full-stack web developer at Unity One Solutions, we have been assigned to design and create a software solution for this project.

The application will cater to the following user roles:

1. Members: These are adults who meet the eligibility criteria and require meal services due to age, disease, or disability. They will use the system to register, specify their requirements, and access meal services.
2. Caregivers: Individuals responsible for taking care of the members and assisting them with their food services. Caregivers will use the system to help members register, update information, and coordinate food deliveries.
3. Partners: Food service providers and organizations partnered with Meals on Wheels to support meal delivery operations. Partners will interact with the system to register, provide their details, and collaborate in meal planning, preparation, and delivery.
4. Volunteers: Individuals who volunteer their time and services to assist Meals on Wheels in delivering meals or cooking for members. Volunteers will use the system to register, provide their information, and participate in meal-related activities.
5. Donors/Supporters: Individuals or organizations interested in financially supporting the Meals on Wheels project. Donors and supporters will interact with the system to make donations, view fundraising efforts, and learn about the members.
6. Administrators: Staff responsible for managing and overseeing the operations of Meals On Wheels. Administrators will have privileged access to the system, enabling them to manage user accounts, monitor meal deliveries, evaluate needs, generate reports, and ensure effective program management.

**List of Figures if Any:**

**1.0 Introduction:**

**1.1 Project background**:

The project "Meals on Wheels" aims to provide a comprehensive software solution for MerryMeal, a charitable organization dedicated to delivering nutritious meals to homebound individuals who are unable to cook for themselves due to age, disease, or disability. In collaboration with Unity One Solutions, the project will encompass designing the software, creating a development plan, and implementing a functional application. The service will operate from Monday to Friday, with frozen meals provided over the weekend for members located beyond a 10-kilometer radius. By partnering with food service providers nationwide, MerryMeal seeks to ensure efficient and prompt meal delivery to those in need. As a full-stack web developer at Unity One Solutions, we have been assigned to design and create a software solution for this project.

The application will cater to the following user roles:

1. Members: These are adults who meet the eligibility criteria and require meal services due to age, disease, or disability. They will use the system to register, specify their requirements, and access meal services.
2. Caregivers: Individuals responsible for taking care of the members and assisting them with their food services. Caregivers will use the system to help members register, update information, and coordinate food deliveries.
3. Partners: Food service providers and organizations partnered with Meals on Wheels to support meal delivery operations. Partners will interact with the system to register, provide their details, and collaborate in meal planning, preparation, and delivery.
4. Volunteers: Individuals who volunteer their time and services to assist Meals on Wheels in delivering meals or cooking for members. Volunteers will use the system to register, provide their information, and participate in meal-related activities.
5. Donors/Supporters: Individuals or organizations interested in financially supporting the Meals on Wheels project. Donors and supporters will interact with the system to make donations, view fundraising efforts, and learn about the members.
6. Administrators: Staff responsible for managing and overseeing the operations of Meals On Wheels. Administrators will have privileged access to the system, enabling them to manage user accounts, monitor meal deliveries, evaluate needs, generate reports, and ensure effective program management.

**1.2 Pain points**

Business Problem Statement:

1. Data Management Challenge: the efficiency and accuracy in handling and storing data, posing challenges to the project's operations.
2. Poor Food Management and Food Safety: Inadequate food management system and lack of food safety measures can undermine customer trust in the quality and hygiene of the provided food.
3. Lack of Responsiveness and Customer Support: Insufficient responsiveness, politeness, and customer support can negatively affect the brand image and customer satisfaction.
4. Financial Constraints: Limited funds can restrict the project's development, maintenance, and growth.
5. Delivery Driver Training Issues: Insufficient training for delivery drivers can result in uncertainties in deliveries and impact service quality.
6. Limited Customer Interaction: In offline food delivery business, there is limited customer interaction and reach, which can hinder the project's growth and success.

Based on the pain points or business problem statement, a business analysis is conducted to evaluate **external and internal factors** that can impact the project. The SWOT analysis helps identify key strengths, weaknesses, opportunities, and threats associated with Merry Meals. These factors are summarized as follows:

1. **Strengths**
2. Unique and cost-effective resources provide a competitive advantage.
3. Market recognizes the company's activities and resources as strengths.
4. Online application saves time for customers.
5. Company offers a distinctive delivery proposition.
6. Online system improves user data management.
7. **Weaknesses**
8. Opportunities to improve the application by addressing bugs and risks.
9. Some activities in the Online Food Ordering System can be optimized.
10. Factors that may negatively impact sales need attention.
11. Competitor activities can pose challenges.
12. **Opportunities**
13. Industry trends offer growth opportunities.
14. Embracing technology and market strategies can open new avenues.
15. Government policy changes can benefit the company.
16. Learning from competitors can drive improvement and innovation.
17. Adapting to changes in social patterns and lifestyles can lead to new services or products.
18. **Threats**
19. Competition and competitor activities pose a threat.
20. Maintaining high product and service quality is crucial.
21. Staying updated with changing technologies is necessary.
22. Financial and cash flow challenges can impact the project.

MOST Analysis is a strategic framework used to evaluate and align critical components of an organization or project. It stands for Mission, Objectives, Strategies, and Tactics, and serves as a guide to understand the overarching purpose, set specific goals, devise effective strategies, and implement actionable plans.

1. Mission: The primary objective of Meals on Wheels is to furnish nourishing, hot meals to eligible individuals who are unable to cook or maintain their nutritional well-being due to age, illness, or disability. This mission guides their overall direction and purpose.
2. Objectives:

* Within the initial year of implementing the application, aim to increase the number of registered members and caregivers by 20%.
* Enhance the efficiency of menu planning and meal preparation processes to ensure prompt and suitable meal delivery.
* Improve the meal delivery management system to track and monitor delivery routes, enabling swifter and more efficient service.
* Establish a robust fundraising module to attract donors and supporters, with the goal of raising 30% more funds than previous years.
* Implement a reassessment system to evaluate the evolving needs of members and provide appropriate meal options accordingly.
* Develop a comprehensive management information system to enable effective monitoring and decision-making within the organization.

1. Strategies:

* Employ a user-friendly application interface with intuitive workflows to encourage adoption and ease of use by members, caregivers, partners, and volunteers.
* Collaborate with food service providers nationwide to ensure timely and reliable meal delivery, particularly for members residing more than 10 kilometers away.
* Leverage technological advancements to streamline processes and minimize delays in meal delivery, addressing any potential issues with late food delivery.
* Foster partnerships with local businesses and organizations to raise awareness and garner support and funding for Meals on Wheels.
* Continuously assess the application's performance and gather user feedback for ongoing improvements and enhancements.
* Maintain a strong relationship with government entities to secure ongoing support and foster collaboration for the program.

1. Tactics:

* Implement a responsive web and mobile application with a user-friendly interface for seamless registration, meal selection, and communication.
* Incorporate GPS tracking and optimization algorithms to streamline delivery routes and reduce delivery times.
* Forge partnerships with reputable food suppliers to ensure the quality and freshness of meals, especially for those receiving frozen meals.
* Utilize social media and online platforms to create awareness and engage potential donors and supporters.
* Provide comprehensive training and support to volunteers, partners, and caregivers to maximize the application's features and efficiency.

PESTLE Analysis is a framework used to analyze and understand the external factors that can impact an organization or project. It stands for Political, Economic, Social, Technological, Legal, and Environmental factors. By examining these key elements, organizations can assess the broader external environment and identify potential opportunities and threats that may influence their operations.

PESTLE Analysis for Meals on Wheels:

* Political:
  + Government support and financial backing can positively influence the program's expansion and sustainability.
  + Adherence to food safety regulations is crucial to maintain compliance and safeguard the program's reputation.
* Economic:
  + Economic conditions may impact the availability of resources and funding for the program.
  + Changes in food prices and inflation rates can affect the program's budget for meal preparation and delivery.
* Sociocultural:
  + Public awareness and support for charitable initiatives can contribute to increased engagement and financial contributions.
  + Consideration of members' cultural preferences and dietary requirements is essential when planning menus and meal options.
* Technological:
  + Technological advancements can enhance the efficiency and effectiveness of the application, such as utilizing GPS tracking for streamlined delivery management.
  + Staying updated on technological trends and security protocols is important to ensure user trust and protect their personal information.
* Legal:
  + Compliance with data protection and privacy laws is vital to safeguard user data and maintain legal standards.
  + Adhering to health and safety regulations in food preparation, handling, and delivery is essential for the well-being of program members.
* Environmental:
  + Environmental factors may impact ingredient availability and affordability for meal preparation.
  + Prioritizing sustainable practices, such as minimizing food waste and utilizing eco-friendly packaging, can align with environmental values and garner support.
  1. **Project goal, objectives and scope**

**1.3.1 Project Objectives**

1. **Analyze Meals on Wheels' business requirements:** Conduct a thorough analysis of Meals on Wheels' operations, procedures, and data to identify their specific needs and challenges in providing meals to members.
2. **Develop an appropriate software solution**: Based on the analysis, propose and create a detailed design plan and blueprint for the software solution. This includes defining the system architecture, user interface design, UI/UX plan, storyboards, database structure, and other technical specifications to ensure a solid foundation for development.
3. **Implement a functional business application:** Develop the software solution according to the finalized design plan, considering the waterfall model's sequential approach. This involves coding, testing, and integrating various components to create a fully functional application that meets Meals on Wheels' requirements.
4. **Evaluate application performance and identify improvements:** Assess the application's performance against its intended goals, ensuring that it aligns with the design specifications. Analyze factors influencing performance, identify any shortcomings, and provide recommendations for improvement in subsequent iterations or phases of the project.
   * 1. **Project Goals**

The project aims to provide nutritious hot lunch meals to eligible adults through Meals on Wheels. This involves partnering with food and donation service providers for efficient meal delivery. A dedicated sitter manages the food service, while riders ensure timely delivery. The project also encourages volunteer and partner involvement to support the charitable cause. Ultimately, the goal is to make a positive impact by providing meals, fostering community engagement, and ensuring reliable service delivery.

* + 1. **Project Scope**

**System Overview -** The application will have the following functionality:

1. Members and Caregivers registration with their requirements
2. Partners and Volunteers registration with their details
3. Fund raising through Donors / Supporters
4. Menu Planning and Preparation
5. Meal Delivery Management of partners and riders
6. Food Safety Management
7. Reassessment of need evaluation, and,
8. Management Information System for effective management

**2.0 Project Initiation:**

* 1. **Project Stakeholder**
     1. What are stakeholders

Stakeholders are individuals, groups, or organizations with an interest or impact on a project's outcome. They can be internal or external, and their needs and expectations must be understood and managed for project success. Effective stakeholder management involves identifying them, communicating regularly, addressing concerns, and involving them in decision-making. By engaging and collaborating with stakeholders, project managers gain support, mitigate risks, and ensure project outcomes align with stakeholder requirements. This active management enhances project outcomes and minimizes conflicts or resistance during implementation.

* + 1. Identification of key stakeholders

Internal Stakeholders

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Role** | **Description** | **Name** |
| 1 | Project Manager | Implementing project management strategy to achieve the project objectives, & A leader who manages a specific team. | Mr. David |
| 2 | Team Members & Application manager | Employees who work as a team to handle all tasks assigned by manager. | Abdul,  Ajri,  Asep,  Darren,  Syukur, |

**External Stakeholders**

|  |  |  |
| --- | --- | --- |
| **No** | **Role** | **Description** |
| 1 | Member | Qualified person who use apps to Request food. |
| 2 | Caregiver | whose job is to deliver food to members. |
| 3 | Partner | A business partner who serves meals to members. |
| 4 | Volunteer | Those who will join the volunteers |
| 5 | Donor | Those who will provider economic support to the company |
| 6 | Driver | Those who will join the driver |

* 1. **Feasibility study**
     1. **Technical Feasibility**

A technical feasibility assessment examines the available technological resources for our project. This investigation analyzes whether we have the necessary equipment, tools, and technical knowledge to achieve our project objectives**.**

* + - 1. **Hardware, Software and Network requirement**

We will conduct a comprehensive analysis of the hardware, software, and network infrastructure needed to support the software application. This assessment includes identifying the required computing resources for development, testing, and deployment, as well as ensuring the compatibility of the application with the targeted platforms (web and mobile). Furthermore, we will evaluate the network requirements to facilitate seamless communication between the application, MerryMeal's staff, food service providers, and delivery drivers. Our objective is to ensure the availability or procurement of the necessary hardware, software, and network components to enable the successful development and implementation of the application.

* 1. **Hardware Requirement**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Hardware** | | |
| **Type** | **Service** | **Client** |
| **1** | Processor | Intel i5 2.6 GHZ | Intel-i3 our above |
| **2** | Memory | 12GB | 4GB our above |
| **3** | Hard Drive | 4TB SSD NVME | 1TB our above |
| **4** | Network | Gigabit Network | Wifi / Internet Providers |

**B. Software Required**

|  |  |  |
| --- | --- | --- |
| **No** | **Software** | |
| **Software Device** | **Functionality** |
| **1** | Window 10 | Operating System |
| **2** | Visual code, eclipse,spring boot | Web Creation |
| **3** | Tomcat | Web server |
| **4** | MySQL | Databases |
| **5** | Figma | Design |
| **6** | Google,Chrome,microsoft edge | Web Browser |

**C. Network Requirements**

|  |  |  |
| --- | --- | --- |
| **No** | **Network** | |
| **Network Device** | **Functionality** |
| **1** | Switch | Cable connecting network from workstation |
| **2** | UTP Cable | Connecting Medium |
| **3** | RJ 45 Connector | Cable connecting network with LAN Card |

* + - 1. **Familiarity with application and technology**
* **Familiarity with application**

Based on the scenario of the applications, particularly from our experience with the Meals on Wheels project, we have a strong understanding of the features and requirements associated with the “Meals on Wheels” application. This knowledge allows us to identify user needs and design the application accordingly, leveraging technologies like JavaScript, Java, React, and Spring. Additionally. With this knowledge, we are confident in our ability to develop an effective and efficient application that meets the needs and expectations of users.

* **Familiarity with technology**

we have a good understanding of the capabilities and features offered by this technology. we have developed web applications using JavaScript and React, for the frontend and developed backend applications using Java and Spring with this we can adapt in implementing the features needed in this application

* + - 1. **Project Size**

We will develop an online food delivery system that involves features such as user registration, login system, food ordering, order management, and delivery notifications. The project will also involve developing a responsive and intuitive user interface, as well as integration with online payment systems. With these various features, this project is considered to be quite large and complex in size. Therefore, it will require significant team effort in terms of development, testing and implementation to ensure the success of this project within the specified timeline.

* + 1. **Operational feasibility**

The operational viability of a project determines whether your firm has the necessary resources, skills, and competences to finish it. It entails assessing workforce needs, organizational structure, and any legal implications. The operational feasibility assessment determines whether your team is capable of completing the project and whether you have the necessary resources and competencies.

**Operational Feasibility**

|  |  |  |
| --- | --- | --- |
| **No** | **Operational Feasibility** | |
| **Requirements** | **Description** |
| **1** | **Performance** | * The system is capable of handling a high volume of simultaneous requests, serving up to 1,000 requests at the same time. * The performance parameter will be evaluated based on response time, throughput, and scalability. |
| **2** | Information | * More accurate information with an accuracy of 5 decimal places. * The accuracy of data will be assessed based on comparison with trusted sources and data validation techniques. |
| **3** | Economy | * The system implementation will lead to cost savings as error reporting can be minimized, reducing potential financial losses. |
| **4** | Control | The system will enforce strict user access control measures to maintain data security and prevent unauthorized access. |
| **5** | Efficiency | The system will save time by avoiding redundant data entry through the use of a centralized database. |
| **6** | Services | The system will enhance service quality by presenting information consistently across all interactions. |
| **7** | Operational Feasibility Value | Locally based system (for organizations only), with Layout common as possible, so it has a value of 8.5. |

based on the operational feasibility that we analyzed, we can get some conclusions which are mentioned below :

1. Will the users use the proposed system?

Yes, users are likely to use the proposed system because it addresses their needs, improves accuracy, efficiency, and service quality, and provides a consistent format for information presentation.

1. What skills are required for all of the team member to develop this project?

Users may need to acquire or enhance skills related to using the system's interface, understanding the workflow, and adapting to new data entry processes. Training on system security and user access control may also be required.

1. How long will it take to retrain the team members to adapt to the new system?

The duration of all team members retraining will depend on factors such as the complexity of the system, the size of the user base, and the availability of training resources. Generally, it can range from a few days to several weeks to ensure a smooth transition and provide enough time for practice and support.

* + 1. **Economic feasibility**

This analysis determines whether the proposed software development project will be financially viable and sustainable. It entails a careful evaluation of the advantages and disadvantages of creating, implementing, and maintaining the application. The possible return on investment as well as development costs, license fees, hardware and software costs, continuing maintenance and support costs, and other expenses will all be taken into account during the economic feasibility analysis. The analysis seeks to ascertain whether the benefits received from the application, such as better operational efficiency, cost savings, and improved service quality, surpass the related expenses by looking at these financial elements. This assessment ensures that MerryMeal can allocate resources wisely and that the initiative will ultimately be financially viable.

**Cost Benefit Analysis**

* + - * **Cost analysis**

|  |  |  |
| --- | --- | --- |
| List of cost needed | Description of activity | Total cost in month |
| Requirement gathering | Activities to gather and document project requirements, including meetings with stakeholders, interviews, research, and analysis. | $400 |
| Hardware | Procurement and installation of a 4TB hard drive for data storage and processing requirements. | $1.500 |
| Software | Purchase or licensing of necessary software tools and applications required for the project development and execution. | $1.200 |
| Hosting | Subscription or rental fees for hosting services, including web servers or cloud infrastructure to deploy and host the application. | $200 |
| Maintenance | Ongoing support, updates, bug fixes, and enhancements to ensure the system's functionality, security, and performance. | $800 |
| Design Implementation | Implementation of the design elements, including user interface (UI) design, user experience (UX) design, and visual design. | $3.000 |
| Human resource | Costs associated with hiring and retaining personnel for project development and management, including salaries, benefits, and training. | $3.000 |
| Marginal / urgent cost | Additional costs that may arise unexpectedly during the project's lifecycle, such as hardware or software upgrades, unforeseen requirements, or urgent fixes. | $400 |
| Total cost | Total cost The sum of all the costs listed above, representing the total expenditure required for the project. | $21.000 |
| Total cost expected |  | $20.000 |

* **Benefit analysis**

|  |  |  |
| --- | --- | --- |
| Benefit gotten |  | Const / Month |
| Direct benefit | $5.000 amount now $25.000 as donation | $25.000 |
| Indirect benefit | Overall perfomance on the websiteis $500, now $1000 | $1000 |

* **Return in investment**

Total Value = 32000

= around 23% return in investment profit

* + - 1. **Direct Cost and Benefit**

Tangible and intangible cost and benefit as following:

* + - * Tangible cost:

Tangible costs for the project include expenses such as the procurement and installation of a 4TB hard drive for data storage ($1,500), the purchase or licensing of necessary software tools and applications ($1,200), subscription or rental fees for hosting services ($200), ongoing support and maintenance ($800), implementation of design elements ($3,000), human resource costs ($3,000), and any additional marginal or urgent costs that may arise ($400). These costs can be measured and quantified in monetary terms.

* + - * Tangible benefit:

On the other hand, tangible benefits for the project include a direct benefit of $25,000, representing a generous donation received, and an indirect benefit of $1,000, resulting from an improvement in the overall performance of the website. These benefits can be easily quantified and measured in terms of their monetary value.

* + - * Intangible cost:

the requirement gathering process, which incurs a cost of $400. This cost is considered intangible because it is difficult to directly quantify or measure in monetary terms. Intangible costs may involve factors such as time investment, effort, or resources allocated to activities that do not have a direct monetary value.

* + - * Intangible benefit:

Intangible benefit will occur when the investment of the design implementation ($3,000) and ongoing maintenance ($800) of the project, the intangible benefit of improved user experience can be expected. A well-designed user interface (UI) and user experience (UX) can lead to enhanced user satisfaction, ease of navigation, and intuitive interaction with the application or website.

* + 1. **Schedule feasibility**

­­Feasibility schedule in this project “Meals on Wheels” make sure that the meal Scheduling feasibility is essential to ensure that food can be provided to target groups in a timely and regular manner. We divide it into a different task that need to be considered in determining the affordability of the schedule include:

1. Week 1 - Initiation Phase (6/12/23 until 6/16/23):

* Conduct interviews and meetings with Meals on Wheels stakeholders to understand their needs and objectives.
* Analyze Meals on Wheels business plan and identify specific requirements for the software application.
* Validate and prioritize requirements with stakeholders.
* Assess the feasibility of the project based on available resources, budget, and timeline.
* Create a detailed project plan and timeline, considering resource allocation, task dependencies, and milestones.

1. Week 2 Design and Development Planning (6/19/23 until 6/23/23):

* Develop design recommendations and strategies that align with Meals on Wheels business plan and requirements.
* Define the system architecture and technology stack to be used.
* Obtain necessary approvals and ensure alignment with stakeholders.
* Start designing the flow diagram, UML, ER Diagram

1. Week 3 Application Development (6/23/23 until 6/5/23):

* Set up the development environment and infrastructure.
* Implement backend functionality using Java Spring Boot and RESTful APIs.
* Develop frontend components using React.js, React Bootstrap, HTML, and CSS.
* Integrate frontend and backend systems.
* Implement specific features, such as member and caregiver registration, partner and volunteer registration, fundraising module, menu planning, meal delivery management, food safety management, reassessment and evaluation system, and management information system.

1. Week 4 Monitoring / Controlling (6/5/23 until 7/7/23):

* Create comprehensive test cases based on the defined requirements.
* Perform functional testing to ensure the application meets the specified criteria.
* Conduct integration testing to verify the seamless interaction between different system components.
* Identify and resolve any bugs or issues discovered during testing.
* Conduct user acceptance testing with stakeholders to ensure the application meets their expectations such as Unit Testing, UAT, performance, compatibility, etc.

In conclusion, the feasibility schedule for the "Meals on Wheels " project outlines the necessary tasks and phases to develop and implement the software application successfully. It includes stages such as feasibility study, design and planning, development, testing, and deployment. By following this schedule, the project aims to meet Meals on Wheels objectives and deliver a functional application within the allocated time and resources.

* + 1. **Risk Feasibility**

**Technology Risk:** Technology risk involves challenges or issues related to the technology used in the development process. This could include issues such as compatibility problems between different software or hardware components, limitations of chosen technologies, or reliance on outdated or unsupported technologies. The likelihood of this risk occurring is rated at 4, and the consequence is rated at 5, resulting in an extreme level of risk. To address this risk, conducting thorough testing and quality assurance measures throughout the development process to identify and address any technological issues is a must. Additionally, staying updated with the latest technological advancements and best practices to ensure the application is built using robust and reliable technologies.

**Physical Risk:** Physical risk refers to risks associated with the physical infrastructure, hardware, and equipment used in the development process. This can include risks such as equipment failure, damage, or loss due to accidents, natural disasters, or security breaches. The likelihood is rated at 2, and the consequence is rated at 5, leading to a high-risk level. To address this risk, ensuring a safe physical workspace for team members by implementing appropriate safety measures and protocols. Regular equipment maintenance and backups will be conducted to prevent data loss or hardware failures. Additionally, securing appropriate insurance coverage to mitigate financial losses in case of physical damage.

**Human Factor Risk:** Human factor risk encompasses risks arising from human error, lack of expertise, or inadequate collaboration and communication within the development team. This can include challenges in project management, resource allocation, skill gaps, or team dynamics that may affect the successful completion of the application. The likelihood is rated at 4, and the consequence is rated at 4, resulting in an extreme level of risk. To address this risk, providing comprehensive training and onboarding programs for the development team to enhance their skills and knowledge. Regular communication, collaboration, and feedback sessions will be conducted to address any concerns, ensure clarity, and promote a culture of continuous improvement.

**Political Risk:** Political risk involves risks associated with changes in political landscapes, government regulations, or policies that could impact the development and deployment of the application. This can include the possibility of restrictions, bans, or legal obstacles imposed by governments or regulatory bodies that may affect the project's scope or viability. The likelihood is rated at 1, and the consequence is rated at 3, resulting in a low-risk level. To address this risk, staying informed about political landscapes, regulations, and policies is crucial. Establishing strong relationships with relevant authorities and industry experts can help navigate potential obstacles. Having a contingency plan that includes alternative strategies and diversifying the project's scope can provide flexibility in case of political changes.

**Natural Risk:** Natural risk pertains to risks related to natural disasters or environmental factors that could disrupt the development process. This includes events such as earthquakes, floods, storms, or other natural calamities that could damage physical infrastructure, result in data loss, or disrupt operations. The likelihood is rated at 1, and the consequence is rated at 4, resulting in a moderate level of risk. To address this risk, implementing disaster preparedness plans is essential. This includes creating backup systems and data storage, conducting regular safety drills, and establishing communication protocols for emergencies. Collaborating with local authorities, insurance providers, and risk management experts can help develop effective response plans and mitigate potential disruptions.

**Economic Risk:** Economic risk involves risks arising from changes in economic conditions, market fluctuations, or financial constraints that could impact the development project. It includes factors such as budget limitations, unexpected cost increases, or changes in resource availability. The likelihood is rated at 2, and the consequence is rated at 3, leading to a moderate level of risk. To address this risk, conducting thorough financial planning and monitoring is crucial. This includes regular budget reviews, cost tracking, and exploring alternative funding options. Developing contingency plans and maintaining good relationships with suppliers and vendors can help mitigate the impact of economic fluctuations or unexpected cost increases.

* + 1. **Legality Feasibility**

Legal feasibility is the feasibility of legality or legal force. This means that the proposed information system may not contradict applicable legislation, whether determined by the government or formed by law in accordance with organizational regulations.

* Intellectual Property Rights: Care should be taken to respect intellectual property rights when using third-party software, content, or designs in the project. Proper licenses or permissions should be obtained to avoid legal issues.
* Data Privacy and Security: The project should comply with applicable data privacy regulations to protect the personal information of members, volunteers, and donors. Implementing measures such as secure data storage, consent management, and encryption will be essential.

By analyzing legal feasibility, the project can operate within legal boundaries and protect the interests of all stakeholders involved.

* 1. **UI/UX Design Consideration**
  2. **2.3.1. Menu Bar (Navbar):**

- Ensure the menu bar prominently displays partner-related features and functionalities, such as menu creation, meal preparation, and inventory management.

- Use clear and intuitive labels in the menu bar to assist partners in easily navigating through their tasks and accessing relevant sections.

**2.3.2. Content:**

**- Layout and Design Clarity:**

- Design a clean and organized interface for partners to manage their tasks efficiently.

- Use a logical and intuitive arrangement of elements, such as ingredient lists, meal preparation instructions, and dietary information, to enhance clarity.

**- Functionality:**

- Implement seamless functionality for partners to create and manage the menu, add or update meal items, and set availability.

- Provide features to accommodate partners' specific needs, such as ingredient quantity tracking and allergy warnings.

**- Responsiveness:**

- Ensure the partner interface is responsive and adaptable to various devices and screen sizes, allowing partners to access and manage their tasks on desktops, mobile devices, or tablets.

- Optimize the layout and content presentation for smaller screens without sacrificing usability.

**- Short Loading Time:**

- Optimize the performance of the partner interface to reduce loading times and provide partners with a smooth and efficient user experience.

- Implement caching mechanisms or asynchronous loading of data to minimize waiting times during menu creation and meal management.

**2.3.3. Visual Design:**

**- Unity:**

- Maintain a consistent visual style throughout the partner interface, using a cohesive color scheme, typography, and design elements.

- Ensure that all sections and components related to partner tasks visually integrate with each other and convey a sense of unity.

**- Balance:**

- Distribute interface elements evenly to create visual balance and prevent the interface from feeling cluttered.

- Group related elements logically, such as placing ingredient lists near meal preparation instructions, to enhance the overall balance.

**- Contrast:**

- Use contrast effectively to highlight important information and guide partners' attention to critical elements.

- Employ color, size, or typography variations to differentiate sections or emphasize actionable items within the partner interface.

**- Scale:**

- Leverage scale to establish visual hierarchy and emphasize the importance of different elements.

- Increase the size or prominence of key features or actions, such as menu creation or updating meal availability, to communicate their significance to partners.

**- Dominance:**

- Highlight critical actions or information relevant to partners by utilizing visual cues, such as prominent placement or distinctive styling.

- Ensure that partners can easily identify and access essential functions related to their role, such as meal preparation and inventory management.

By incorporating these UI/UX design aspects into the partner interface, the Meals on Wheels application can provide partners with a visually appealing, intuitive, and engaging experience that enhances their ability to manage menu creation, meal preparation, and other related tasks effectively.

* 1. **Dependencies** 
     1. **Logical dependencies**

This dependency is critical for project completion and is a component of the project.

* Menu planning and food preparation: Menu planning should be related to proper food preparation.
* Member and caregiver requirements and reassessment: Member and caregiver requirements must be updated regularly.
* Meal delivery management and partners/volunteers registration: Management of food delivery depends on registered partners and volunteers.
* Food safety management and menu planning/preparation: Food safety must be a concern in menu planning and preparation.
* Fundraising and management information system: Financial and supporting information must be integrated with the management system for effective fundraising
  + 1. **Resource dependencies**

Resource dependency are driven by constraint

* Human resources, such as skilled workers in farming, cooking, or volunteers/ support roles.
* Financial resources are crucial for funding the procurement of food, operational expenses, infrastructure, promotion, and project development.
* Physical resources, such as agricultural land, kitchen or food production facilities, storage facilities, and cooking equipment, are necessary for the project's operations.
* Technological resources, like food processing systems or project management software, and time resources for effective scheduling and allocation are also important.
* Additionally, knowledge and information resources, including food research, nutritional guidelines, and best practices in project management, play a role in decision-making and innovation.
  + 1. **External dependencies**

Based on outside factor and unexpected event

* Natural disasters: Delays in food delivery caused by unforeseen natural disasters, such as earthquakes, floods, or storms.
* Economic fluctuations: Affordability and accessibility of food resources affected by sudden economic changes, leading to challenges in maintaining the supply chain within budget constraints.
* Political instability: Uncertainties in the regulatory environment due to political instability or changes in government policies, requiring adjustments to comply with new regulations.
* Public health crises: Disruptions in external dependencies caused by unexpected public health crises like pandemics, affecting the availability of food resources and the project's ability to reach the target population.
* Socio-cultural shifts: Adaptations in external dependencies to address rapid socio-cultural changes or shifts in dietary preferences and trends.
  1. **Project Assumptions** 
     1. **Resources**

Possible resource will be taken in Meals on Wheels Application

* Human resources: Skilled developers, designers, and testers will be required to develop and maintain the Meals on Wheels application.
* End members: The target beneficiaries of the Meals on Wheels Application. These are elderly or disabled individuals who will receive free meals through the application. Consider the number of end members that the application aims to serve and the potential demand for their specific needs.
* Volunteers: Partner volunteers who will cook the meals and caregiver volunteers who will deliver the food to the members.
* Donors: The application assumes the availability of donors who will fund the meals cooked by the partner volunteers.

.

* + 1. **Finance**

Possible finance will be taken in Meals on Wheels Application

* Company finances: Adequate funding will be necessary to cover the operational costs of the company, including expenses related to the procurement of food supplies, transportation, maintenance of the application, and any other necessary expenses to support the operations of Meals on Wheels.
* Donor funding: The application assumes the availability of donors who will contribute financial resources to support the provision of free meals to the elderly or disabled individuals. These funds will be utilized for procuring the necessary ingredients and supplies for the partner volunteers to prepare the meals.
  + 1. **Partner**
* Meals on Wheels food production compliance: The food production process for Meals on Wheels will adhere to the Food Safety System Certification (FSSC) standards to ensure the safety and quality of the meals.



*Notes: This is fake certificate, just for example*

* HALAL certification: The meals provided by Meals on Wheels will have HALAL certification to comply with halal regulations and cater to the dietary preferences of the target beneficiaries.



* + 1. **Project**
* Meals on Wheels Website: The Meals on Wheels application will have a website component that will require ongoing maintenance and updates.
* SSL Certification: The application will have SSL certification to ensure secure and encrypted communication between users and the server.
* Daily backups: The application's data will be backed up every day within a 10-hour timeframe to prevent data loss in case of any issues.
* Project timeline: The assumption is that the project will take approximately one month to complete the development phase, followed by testing conducted by the developers.
  1. **Project Constraint**

Project constraints refer to the limitations or restrictions that affect the planning, execution, and completion of a project. Common types of project constraints include:

* + 1. **Client**
* Hot meals can be delivered under a radius 10 KM.
* Frost meals can be delivered above a radius 10 KM.
* Service only available every day except Saturday and Sunday.
* Different ages of customers will be considered in developing the Merry on Wheels website in order to avoid misleading or misunderstanding stuff.
  + 1. **Time**

Meals on Wheels Project website must be developed within 1 month, of which 2 weeks or more is used to carry out project analysis to make project plans, designs etc, and the rest to carry out project implementation

* + 1. **Scope**
* The project scope, as outlined in the project overview, must be strictly adhered to. Any changes or additions to the functionality or features may require proper assessment, approval, and potential adjustments to the project timeline and budget.
  + 1. **Cost**

• The Meals on Wheels project’s budget is around $25.000, Any cost growth beyond the calculated budget may impact the scope of the project, potentially resulting in the exclusion of certain features or functionalities.

* + 1. **Resources**
* The process of developing the projects itself has limited resources software tools, and hardware infrastructure. These resources need to be effectively managed and utilized to meet project requirements.
  + 1. **Environment**
* Convenience workplace for developer teams.
* Each every team members must have an effective communication channels and collaboration for seamless coordination among team members, stakeholders, and other relevant parties.
* Maintaning workload balance by scheduling each every team members task for team productivity
  + 1. **Skill**
* Backend Development Skill: Proficiency in Java Spring Boot, RESTful API development, and database management.
* Frontend Development Skill: Proficiency in React.js, React Bootstrap, HTML, CSS, and frontend state management.
* Full-Stack Development Skill: Proficiency in both backend and frontend development for efficient coordination and integration.
* Project Management: Skilled project manager or team lead for overseeing tasks and ensuring effective communication.
* Quality Assurance and Testing: QA professionals with expertise in testing frameworks and tools for thorough application testing.
  + 1. **Man Power**
* Team members are proficient in Java, Spring Boot
* Team members are proficient in Javascript, React
* Team members have experience in full stack web development.
* Team members who possess good project management skills.
* Team members who have expertise in application testing.

These constraints often interact with each other and require careful management and trade-offs to ensure project success.

* 1. **Project Milestone: Progress to measure to achieve goals**

Project milestones are important events or achievements that track a project's progress. They provide a visual representation of the project schedule and help communicate the project's advancement to stakeholders. Milestones are useful for keeping everyone informed, including project team members, managers, clients, and investors. They also initiate key actions like approvals, resource allocation, and budget releases. Overall, milestones are valuable for monitoring project health, managing stakeholder expectations, and celebrating accomplishments. the milestone from planning to completion mention below:

* + 1. **Requirement gathering**

Requirement gathering involves collecting and documenting the needs and expectations of stakeholders. This is done through interviews, meetings, surveys, and analyzing existing documentation or according to the Meals on Wheels project manager. The gathered requirements are then organized, prioritized, and validated with stakeholders. A comprehensive requirements document is created, which undergoes review and finalization. It's an iterative process that requires effective communication and collaboration to ensure accurate and complete requirements.

We analyze the requirements based on the functional and non-functional requirements mentioned below:

**Functional Requirement:**

Functional requirements define the specific functions and capabilities that the software application should possess. These requirements describe what the system must do and how it should behave to meet the needs of its users. Based on the project scope, here are some functional requirements ordered:

* The application should allow new members and their caregivers, volunteers or any user to register by providing necessary personal information.
* The application should facilitate online donations and allow donors to contribute funds to support the organization's activities.
* The application should provide tools for planning and managing menus, including options for creating, modifying, and deleting meal options.
* The application should enable efficient management of meal deliveries, including assigning drivers or riders, tracking delivery status, and providing real-time updates to members and caregivers.

**Non-functional Requirements:**

Non-functional requirements are related to the overall characteristics and qualities of the software application, rather than specific functionalities. They describe how the system should perform, the constraints it should adhere to, and other quality attributes. Here are some examples of non-functional requirements based on the project scope:

* **Performance:** The application should be responsive and provide quick response times to user interactions, even with high concurrent user traffic.
* **Security:** The application should implement robust security measures to protect personal information, donations, and sensitive data from unauthorized access or breaches.
* **Usability:** The application should have an intuitive and user-friendly interface, making it easy for members, partners, and volunteers to navigate and perform tasks efficiently.
* **Reliability:** The application should be stable and available for use during the designated service hours, ensuring uninterrupted access for members, partners, and volunteers.
* **Scalability:** The application should have the ability to handle an increasing number of users, accommodate future growth, and support additional functionalities as the organization expands.
  + 1. **Validate expectations for requirements**

After listing all the requirements requested from the project manager, we clarified the expected requirements which are really required, the way we carry out the verification is mentioned below:

* Review the gathered requirements with stakeholders, including members, partners, volunteers, and the project manager, to ensure they align with their expectations.
* Confirm that the proposed functionalities address the needs of the charitable organization and support their business plan.
  + 1. **Predevelopment planning**

After validating the requirements, the project manager will assign each team member their task, each team member task which is mentioned below:

|  |  |
| --- | --- |
| Team Member’s Name | Task Assigned |
| Abdul Rahman Shalehudin | Handle the functionality of Driver |
| Ajri Muhammad Sidiq | Handle the functionality of Volunteer |
| Asep Supriyadi | Handle the functionality of Administrator |
| Darren Farrell Andrian | Handle the functionality of Member and Donor |
| Syukur Sidiq Nur Alam | Handle the functionality of Partner |

* + 1. **Implementation**

After dividing every tasks to each team member, we start the implemention of the project according to their respective tasks, the process of how we do the implementation are mentioned below :

* Develop the software application based on the functional requirements identified in the proposal.
* Implement features such as member and caregiver registration, partner and volunteer management, donation processing, menu planning, meal delivery management, food safety management, reassessment of needs, and a management information system.
  + 1. **Quality Assurance testing**

After the completion of the implementation we conduct quality assurance testing to measure that a project meets predefined quality standards, the way conduct quality assurance testing are mentioned below:

1. Verification: This stage involves a comprehensive assessment to determine whether the product or deliverables meet pre-established requirements. It entails a thorough review of functionality, design, and implementation in accordance with specified criteria.
2. Validation: The validation process focuses on ensuring that the product or deliverables effectively fulfill the needs of end users and align with desired business objectives. It entails evaluating user satisfaction, usability, and overall congruence of the product with its intended purpose.
3. Functional Testing: Functional testing aims to confirm whether the system or application functions correctly based on predetermined specifications. It entails executing test cases that cover all functional requirements, ensuring that each component operates as intended.
4. Non-Functional Testing: Non-functional testing assesses the non-functional aspects of the system, including performance, security, reliability, and scalability. This type of testing ensures that the system can handle expected workloads, safeguards against potential threats, operates reliably, and accommodates future growth and increased usage.
5. Integration Testing: Integration testing focuses on assessing the interactions between different system components or modules to ensure proper integration. It verifies that data and control flow seamlessly between the parts, ensuring that the integrated system operates as a unified entity.
   * 1. **User Acceptance testing**

We also conduct the User Acceptance Testing (UAT) is done to ensure that the system meets the requirements and expectations of end users or clients before it is deployed. It allows users to test the system and provide feedback to verify its usability, functionality, and overall readiness for production. UAT helps ensure that the system is user-friendly and aligned with user needs. The way we conduct the UAT testing are mentioned below:

* Involve members, partners, volunteers, project manager, or any other stake holders to test the application to ensure it meets their expectations.
* Allow them to provide feedback on the usability, functionality, and overall user experience of the application.
  + 1. **Deployment**

The final step involves making the website accessible to the public once all previous tasks have been completed and there are no pending requirements or unfinished work. During the deployment phase, the website is configured to go live on the chosen server or hosting platform. This process includes adding the necessary files, configuring options, and ensuring the website is fully functional and available for users. Deployment represents the culmination of the development process, as the website is now prepared for user interaction and to fulfill its intended purpose. It is crucial to monitor the website's performance and address any user feedback or issues that may arise after deployment, ensuring optimal functionality and user satisfaction. By following a carefully planned and executed deployment process, the website can be successfully launched, providing value and meeting the needs of its target audience.

* + 1. **Support**

During the post-implementation support phase, Merry Meal's dedicated development team will provide ongoing assistance and maintenance. They will diligently monitor the website, promptly addressing any issues or bugs to ensure uninterrupted operation. Routine maintenance tasks, such as applying updates and patches, will be performed to optimize performance. Technical support will be readily available to users, offering solutions to their inquiries and concerns. User feedback will be collected and utilized to drive future improvements, while staying updated with technological advancements to uphold the website's relevance and user-friendliness. The support phase guarantees website stability, functionality, and user satisfaction, delivering long-term value to Merry Meal and its users.

* 1. **Project deliverables** 
     1. **Internal deliverables**
* Web Design Proposal: The deliverable will include adopting a web design proposal that aligns with Meals on Wheels business plan. The proposal should focus on creating an intuitive and user-friendly interface for members, partners, and volunteers to navigate the application easily.
* Software Design Document (SSD): The deliverable will involve adopting a Software Design Document that outlines the technical design of the software application. It should address the specific requirements of Meals on Wheels , such as member registration, donation management, menu planning, meal delivery, and other functionalities.
  + 1. **External deliverables**
* Enterprise Web Application: The deliverable will consist of developing a functional enterprise web application that supports Meals on Wheels operations in providing hot noon meals to qualifying adults. It should include features like member registration, donation management, menu planning, meal delivery management, management information system, etc.
* User Documentation: The deliverable will involve creating user documentation that provides instructions and guidance on how to use the web application effectively. It should cover topics such as member registration, donation processes, menu planning, meal delivery management, and any other relevant aspects.
  + 1. **Planning deliverables**
* **Project Scope:** The deliverable will define the scope of the project, outlining the specific functionalities and features that will be included in the enterprise web application. It should cover areas such as member registration, donation management, menu planning, meal delivery management, food safety management, reassessment of need evaluation, and the management information system.
* **Budget:** The deliverable will include an estimation of the project's budget, considering the resources required for software development, infrastructure, and any other associated costs.
* **Project Schedule:** The deliverable will outline the project's timeline and milestones, specifying the duration for each phase of development, including design, development, testing, and deployment.
  1. **Functional description (Group)**

|  |  |  |  |
| --- | --- | --- | --- |
| No | User type | Description | Roles based access |
| 1 | Member | Meals on Wheels customer who requests meals, typically adults who are qualified and in need of assistance due to age, illness, or disability. | * Register * Login * Order and view meal * Update profile, view meals track * Send feedback / evaluations |
| 2 | Care giver/Drivers | Meals on Wheels support teams or volunteer who has resposibility to deliver the meals to the members. | * Register * Login * View all food delivery schedule * Update delivery status (Pick Up, On the way, completed) * Edit or view profile page |
| 3 | Partners | These are individuals or organizations that collaborate with Meals on Wheels by providing kitchen facilities and agreeing to process meals according to Meals on Wheels specifications. | * Register * Login * View meals * Process the ordered meals |
| 4 | Volunteers | Participate in Charity activity that’s done by Meals on Wheels Organization, this user can either be a Driver or prepare the food if they have a kitchen | * Register * Login * Based on the roles that they choose, either as a caregiver or rider, they are able to access its role. |
| 5 | Donor / supporter | Individuals or organizations who contribute donations or financial support to Meals on Wheels . | * Register * Login * View & Edit profile * Send donations |
| 6 | Administrator | The one who manage all processes, including meal processing, menu creation, driver assignments, and member coordination, to ensure efficient operations. | * Register * Login * Manage donation * Manage member * Manage menu / order * Manage driver * Manage information systems |

* 1. **Risk analysis and description** 
     1. **Identify possible area of risk in the application development**

In the context of application development, it is important to identify potential areas of risk that could impact the success of the project. The identified risk areas are categorized into internal and external risks.

* + - 1. **Internal risk**

Internal risks are those that originate within the organization or project team. In the context of application development, three primary internal risks are identified:

* **Technology Risk**

Technology risk involves challenges or issues related to the technology used in the development process. This could include issues such as compatibility problems between different software or hardware components, limitations of chosen technologies, or reliance on outdated or unsupported technologies. Such risks can impact the performance, functionality, or stability of the application being developed.

* **Physical Risk**

Physical risk refers to risks associated with the physical infrastructure, hardware, and equipment used in the development process. This can include risks such as equipment failure, damage, or loss due to accidents, natural disasters, or security breaches. Ensuring the physical security and stability of the development environment is crucial to mitigate these risks.

* **Human Factor Risk**

Human factor risk encompasses risks arising from human error, lack of expertise, or inadequate collaboration and communication within the development team. This can include challenges in project management, resource allocation, skill gaps, or team dynamics that may affect the successful completion of the application. Addressing these risks involves effective team management, training, and fostering a culture of open communication and collaboration.

* + - 1. **External Risk**

External risks are factors that originate from outside the organization or project team. In the context of application development, three primary external risks are identified:

* **Political Risk**

Political risk involves risks associated with changes in political landscapes, government regulations, or policies that could impact the development and deployment of the application. This can include the possibility of restrictions, bans, or legal obstacles imposed by governments or regulatory bodies that may affect the project's scope or viability.

* **Natural Risk**

Natural risk pertains to risks related to natural disasters or environmental factors that could disrupt the development process. This includes events such as earthquakes, floods, storms, or other natural calamities that could damage physical infrastructure, result in data loss, or disrupt operations. Implementing disaster preparedness plans and data backup strategies can help mitigate these risks.

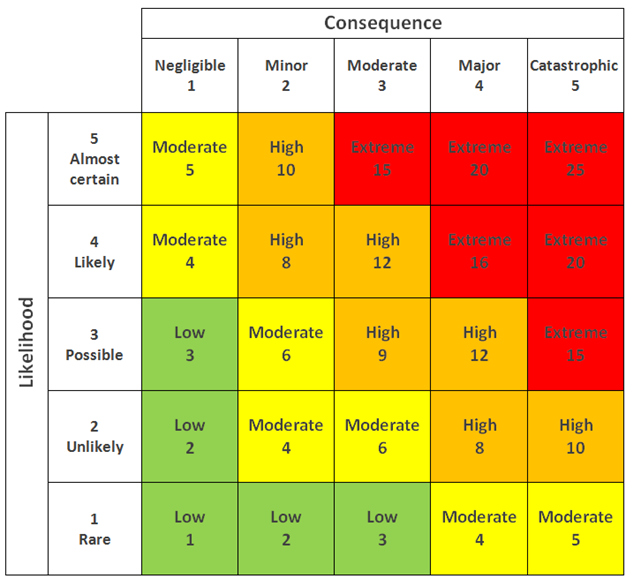
* **Economic risk**

This risk involves risks arising from changes in economic conditions, market fluctuations, or financial constraints that could impact the development project. It includes factors such as budget limitations, unexpected cost increases, or changes in resource availability. Conducting thorough financial planning, monitoring costs, and exploring alternative funding options can help mitigate this risk.

* + 1. **Analyze and Evaluate**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Risk type** | **Risk description** | **Risk Impact** |
| **1** | Technological | Poor quality code when developing application | If the code quality is poor, it can result in software defects, system crashes, and compromised functionality, leading to a negative user experience and increased maintenance efforts. |
| **2** | Human Factor | human error, lack of expertise, or inadequate collaboration and communication within the development team | If there is a lack of communication and coordination, it can lead to misalignment of tasks, misunderstandings, delays, and errors in the development process. |
| **3** | Economic | Unexpected budget constraints | If there are unexpected budget constraints, it can impact resource allocation, limit the scope of the project, and potentially hinder the successful completion of all planned features and functionalities. |
| **4** | Physical | Damage or loss of hardware or infrastructure | If there is damage or loss of hardware or infrastructure, it can disrupt the development process, cause data loss, and result in delays in the project timeline. |
| **5** | Natural | Natural disaster affecting the development environment | If a natural disaster occurs and affects the development environment, it can lead to disruptions, damage to infrastructure, and loss of resources, potentially causing significant delays and setbacks in the project. |
| **6** | Political | Government banning a software or platform used in the project | If one of the software or platforms used in the project is banned by the government, it can result in the need to find and adapt to an alternative solution, causing delays in the project timeline and potentially requiring significant adjustments in the development process. |

* + 1. **Prepare Risk Matrix**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Risk type** | **Risk description** | **Likelihood (1-5)** | **Consequence (1-5)** | **Risk rating** | **Risk level** |
| 1 | Technological | Poor quality code when developing the application. This can occur due to project underestimation or developers rushing to complete iterations, leading to stakeholder dissatisfaction. | 4 | 5 | 20 | Extreme |
| 2 | Human Factor | Insufficient expertise in handling complex project requirements. | 4 | 4 | 16 | Extreme |
| 3 | Economic | Unexpected cost increases due to changes in market conditions or resource availability. | 2 | 3 | 6 | Moderate |
| 4 | Physical | Disruption of the workspace due to equipment failure or damage. | 2 | 5 | 10 | High |
| 5 | Natural | Natural disaster could cause significant damage to the workstation and result in the loss of human resources. | 1 | 4 | 4 | Moderate |
| 6 | Political | This risk involves political factors, such as government regulations or restrictions that could affect the availability or usage of certain software platforms or tools. | 1 | 3 | 3 | Low |

* + 1. Risk Response Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Risk type** | **Risk description** | **Risk Response Plan** |
| **1** | Technological | This risk relates to potential technical issues or limitations that could impact the development, functionality, or performance of the software application. | We will conduct thorough testing and quality assurance measures throughout the development process to identify and address any technological issues. Additionally, we will stay updated with the latest technological advancements and best practices to ensure the application is built using robust and reliable technologies. |
| **2** | Human Factor | This risk involves challenges arising from human error, lack of expertise, or insufficient training that could impact the successful development and implementation of the software application. | We will provide comprehensive training and onboarding programs for the development team to enhance their skills and knowledge. Regular communication, collaboration, and feedback sessions will be conducted to address any concerns, ensure clarity, and promote a culture of continuous improvement. |
| **3** | Economic | This risk pertains to financial constraints, unexpected cost increases, or changes in the economic landscape that may impact the project's budget or viability. | We will conduct thorough financial planning and monitoring, including regular cost reviews and risk assessments. Contingency plans and reserves will be established to mitigate potential economic risks. Additionally, we will explore cost-saving measures and evaluate alternative funding options if needed. |
| **4** | Physical | This risk involves physical damage to the workspace, equipment, or infrastructure that could disrupt the project's progress or availability of resources. | We will ensure a safe physical workspace for team members by implementing appropriate safety measures and protocols. Regular equipment maintenance and backups will be conducted to prevent data loss or hardware failures. Additionally, we will secure appropriate insurance coverage to mitigate financial losses in case of physical damage. |
| **5** | Natural | This risk pertains to natural disasters such as earthquakes, floods, or fires that could cause significant damage to the workspace or result in the loss of human resources. | We will establish a disaster preparedness plan that includes measures to ensure the safety of team members and critical resources. This can include creating backup systems and data storage, implementing off-site data recovery solutions, and providing job insurance to stakeholders to mitigate the impact of potential losses. |
| **6** | Political | This risk involves political factors, such as government regulations or restrictions that could affect the availability or usage of certain software platforms or tools. | We will explore alternative software platforms or tools that are compliant with the political landscape and can provide similar functionalities. Additionally, we will employ dedicated virtual private networks (VPNs) or domain name system (DNS) solutions to bypass country or region restrictions and ensure uninterrupted access to necessary resources. |

**3.0 User and System Requirements:**

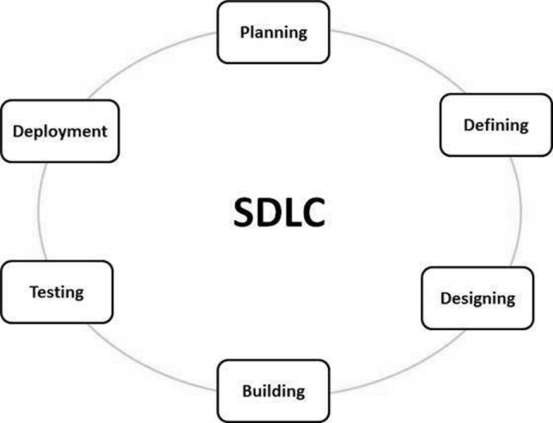
**User Requirements:** means enlist application feature which application will be providing.

|  |  |  |
| --- | --- | --- |
| **No.** | **Feature** | **Description** |
| 1 | Member Registration | Individuals can register their details to become a part of the Meals on Wheels program. |
| 2 | Partner Registration | Organizations or entities can register their information to collaborate with Meals on Wheels for meal prep. |
| 3 | Caregiver/Driver Registration | Caregivers or drivers can register their details to assist in meal delivery to the program members. |
| 4 | Volunteer Registration | Individuals interested in volunteering for Meals on Wheels can register their information. |
| 5 | Member Login | Registered members can log in to access their account and place meal orders. |
| 6 | Partner Login | Registered partners can log in to access their account and receive meal preparation requests. |
| 7 | Caregiver/Driver Login | Registered caregivers or drivers can log in to access their account and view assigned delivery tasks. |
| 8 | Volunteer Login | Registered volunteers can log in to access their account and view volunteer opportunities. |
| 9 | Member Meal Ordering | Registered members can log in and place their meal orders for specific days. |
| 10 | Partner Meal Preparation | Partners can log in, receive meal preparation requests, and cook the meals accordingly. |
| 11 | Driver Meal Delivery | Assigned drivers can log in, view delivery tasks, and ensure timely and efficient meal delivery. |
| 12 | Donation | Donors can contribute funds to support the Meals on Wheels program. |
| 13 | Admin Member Management | Administrators can manage member-related information, including registration, profiles, and services. |
| 14 | Admin Donation Management | Administrators can manage and track donations received from donors. |
| 15 | Admin Meal Management | Administrators can manage meal-related information, including menu planning and nutritional content. |
| 16 | Admin Order Management | Administrators can manage member meal orders and track their status. |
| 17 | Admin Partner Management | Administrators can manage partner-related information, including registration and collaboration details. |
| 18 | Admin Caregiver/Driver Management | Administrators can manage caregiver or driver-related information, including registration and assignments. |
| 19 | Admin Login | Administrators can log in to access the administrative panel for overall management and system control. |
| 20 | Admin Partner and Caregiver Approval | Administrators have the authority to review partner and caregiver registrations and make acceptance or rejection decisions. |

**SDLC (software development life cycle):**



### **SDLC (software development life cycle):**



SDLC, or Software Development Life Cycle, is a process that guides the development and modification of software products. It involves a series of planned activities to ensure efficient and high-quality software development. SDLC provides a framework for designing, developing, testing, and deploying software applications. It helps teams manage the entire development process, from start to finish, while meeting budget and requirements. By following SDLC, software development teams can create reliable and effective software solutions.

**How does SDLC work?**

SDLC works by providing a structured and systematic approach to software development. It outlines a series of phases or stages that software projects typically go through. Here's an overview of how SDLC works:

1. **Planning and Requirement Analysis:** This stage involves gathering requirements from customers, analyzing market surveys, and conducting feasibility studies. The project approach and quality assurance requirements are planned, and risks are identified.
2. **Defining Requirements:** The product requirements are clearly defined and documented in a Software Requirement Specification (SRS) document. Approval from the customer or market analysts is obtained.
3. **Designing the Product Architecture:** Based on the SRS, the product architecture is designed. Multiple design approaches may be proposed and evaluated. The best approach is selected, and the architectural modules and data flow are defined in a Design Document Specification (DDS).
4. **Building or Developing the Product:** The actual development work begins in this stage. The programming code is generated based on the DDS, following coding guidelines and using appropriate programming languages. This stage involves writing the code and implementing the defined design.
5. **Testing the Product:** Product testing is a continuous activity throughout the SDLC stages. In this specific stage, defects are reported, tracked, fixed, and retested until the product meets the defined quality standards in the SRS.
6. **Deployment in the Market and Maintenance:** Once the product is tested and ready, it is deployed in the market. This may happen in stages or targeted market segments. User acceptance testing (UAT) may be conducted to gather feedback. After deployment, maintenance is performed to support the existing customer base.

Based on feedback received, the product may be released with or without suggested enhancements in the target market segment. After the product is released, maintenance is performed to support existing customers and ensure its smooth functioning.

### **SDLC models: consider at least 4 models research:**

#### **Waterfall model**

* **What is Waterfall Model?**

The waterfall model is a straightforward and sequential approach to software development. It divides the development process into distinct phases, where each phase must be completed before moving on to the next. The model follows a linear flow without overlapping phases.

* **Characteristic**

1. **Sequential and Linear Process Flow :** The waterfall model follows a step-by-step approach where each phase is completed before moving on to the next. There is a clear order and no overlap between phases.
2. **Phases Do Not Overlap:** In the waterfall model, each phase is distinct and finishes before the next one begins. There is a sequential progression from one phase to another.
3. **Emphasis on Thorough Documentation:** Documentation is crucial in the waterfall model. Each phase requires comprehensive documentation, ensuring clarity and facilitating the transition to the next phase.
4. **Clear and Stable Requirements Upfront:** The waterfall model assumes that project requirements are well-defined and stable from the start. It focuses on gathering detailed requirements and freezing them early to minimize changes.

* **When we use waterfall model ?**

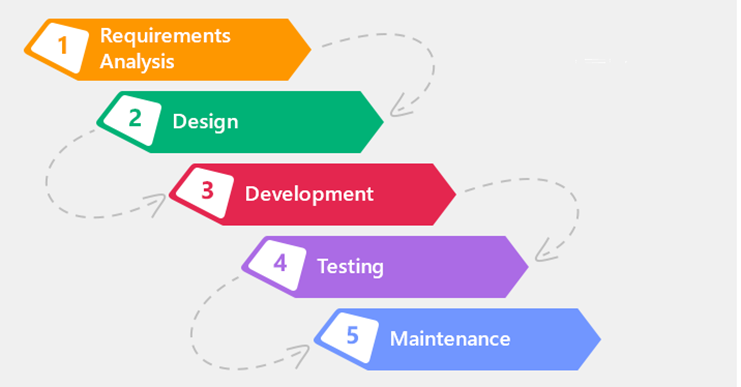
The waterfall model is most suitable for projects that have the following characteristics:

1. **Clear, well-defined requirements:** The project requirements are clearly understood and documented upfront, without significant changes expected during the development process.
2. **Stable technology and minimal uncertainty:** The technology and tools to be used in the project are well-established and stable, with minimal risk of unexpected issues or uncertainties arising during development.
3. **Fixed scope and strict deadlines:** The project has a well-defined scope that is unlikely to change significantly, and there are strict deadlines that need to be adhered to.
4. **Limited or no requirement for user involvement during development:** The project does not require extensive user involvement or frequent feedback and iterations during the development process. User input is not a critical factor in shaping the product.

* **Pros and Cons about waterfall model**

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| 1. Simple and easy to understand | 1. Limited flexibility for changes and adaptations |
| 1. Clear project requirements and scope | 1. Limited customer feedback until late in the cycle |
| 1. Thorough documentation for each phase | 1. High risk of customer dissatisfaction |
| 1. Well-suited for small and straightforward projects | 1. Difficult to estimate accurate time and cost |
| 1. Enables a structured and systematic approach | 1. Challenges in handling complex and evolving projects |

* **Waterfall Model Structure**



#### **Spiral model**

* **What is spiral model**

The spiral model is a software development process model that combines iterative development with the systematic, controlled aspects of the waterfall model. It emphasizes risk analysis and allows for incremental releases or refinement of the product through each iteration around a spiral.

* **Characteristic**

1. **Iterative and Incremental:** The development process occurs in repeated iterations, with each iteration refining the product based on feedback and lessons learned.
2. **Risk Analysis:** The model places a high emphasis on identifying, estimating, and mitigating risks throughout the development process.
3. **Phased Approach:** The model is divided into distinct phases, such as identification, design, construct/build, and evaluation, which are executed in iterations.
4. **Customer Involvement:** Continuous communication and feedback from the customer play a crucial role in shaping the product at each iteration.

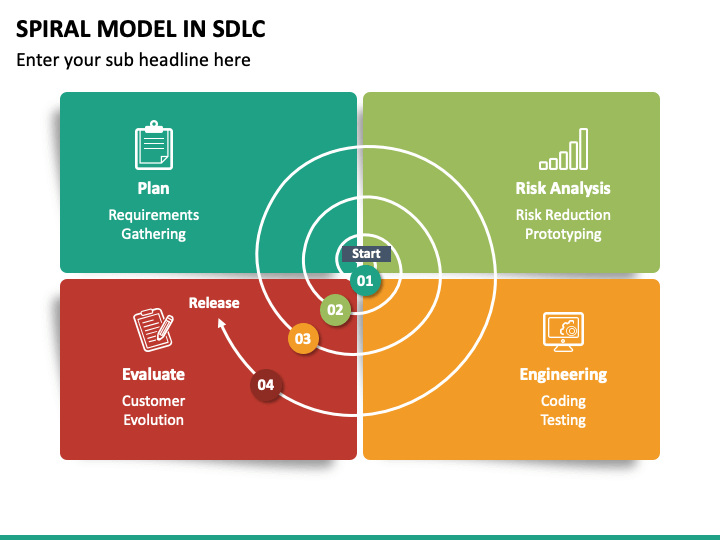
* **When we use spiral model ?**

1. Projects with changing or uncertain requirements.
2. Medium to high-risk projects that require a focus on risk analysis and management.
3. Long-term projects where economic priorities and requirements may evolve over time.
4. Projects with complex requirements that need evaluation and clarification.
5. Projects introducing a new product line that requires phased releases to gather customer feedback.
6. Projects anticipating significant changes during the development cycle.

* **Pros and Cons about spiral model**

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| 1. Accommodates changing requirements | 1. **Management complexity:** The iterative nature of the model increases the complexity of project management. |
| 1. Extensive use of prototypes | 1. **Project end may not be known early:** It can be challenging to estimate the project's completion due to the iterative nature of the model. |
| 1. Improved requirement capture | 1. **Not suitable for small or low-risk projects:** The spiral model may be too costly and time-consuming for smaller or low-risk projects. |
| 1. Early user involvement and feedback | 1. **Process complexity:** The model requires skilled project management and coordination to ensure smooth execution. |
| 1. Better risk management | 1. **Spiral may go on indefinitely**: Without proper management, the iterative nature of the model may lead to indefinite iterations. |

* **Spiral model structure**



#### **V Model**

* **What is V-Model?**

The V-Model is a software development life cycle (SDLC) model that follows a sequential and structured approach. It is called the V-Model because the execution of processes happens in a sequential manner in a V-shaped. It is also known as the Verification and Validation model, as it emphasizes testing at each corresponding development stage.

* **Characteristic**

1. **Sequential and structured:** The V-Model follows a sequential order, with each phase having a corresponding testing phase.
2. **Verification and validation:** It emphasizes the verification of requirements during development and the validation of the final product.
3. **Testing-centric:** Testing activities are integral to each phase, ensuring quality and adherence to requirements.
4. **Rigidity:** The V-Model is less flexible to changes, and modifications to requirements can be costly.
5. **Clear deliverables:** Each phase has specific deliverables and a review process, making it easier to manage and monitor progress.

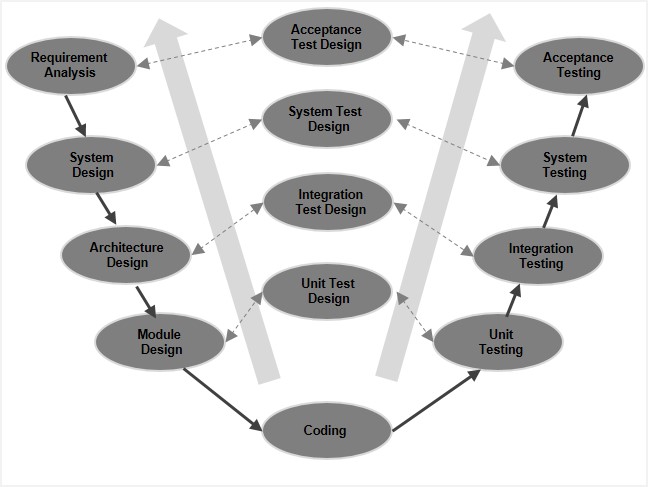
* **When we use V model?**

1. **Well-defined requirements:** The V-Model is suitable when requirements are clear, documented, and fixed.
2. **Stable product definition:** When the product's definition is stable and unlikely to change significantly during development.
3. **Well-understood technology:** The project team should have a good understanding of the technology used.
4. **Short project duration:** The V-Model works well for smaller projects with a shorter duration.
5. **Minimal requirement changes:** It is not suitable for projects where requirements are prone to frequent changes.

* **Pros and Cons**

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| 1. **Easy to understand and apply:** The simplicity of the model makes it easy to comprehend and implement. | 1. **High risk and uncertainty:** The rigidity of the model increases the risk of changes and uncertainty. |
| 1. **Phases completed in a disciplined manner:** Each phase is completed before proceeding to the next, ensuring thoroughness. | 1. **Not suitable for complex projects:** It may not be suitable for complex and object-oriented projects that require more flexibility. |
| 1. **Suitable for well-defined requirements:** Works well when requirements are clear and documented. | 1. **Challenging for long-term projects:** The V-Model may not be ideal for projects with long durations and ongoing development. |
| 1. **Clear deliverables and review process:** Specific deliverables and reviews make it easier to manage and monitor progress. | 1. **Difficult to accommodate requirement changes:** Changes in requirements are costly and challenging to implement once in the testing stage. |

* **V model Structure**



#### **Agile model**

* **What is agile model?**

The Agile model is an iterative and incremental software development approach that focuses on adaptability and customer satisfaction by delivering working software products in small incremental builds. It involves cross-functional teams working simultaneously on various areas such as planning, requirements analysis, design, coding, unit testing, and acceptance testing. Each iteration, typically lasting from one to three weeks, concludes with a working product demonstration to the customer and stakeholders.

* **Characteristic**

1. Iterative and incremental development: The project is divided into small iterations, with each iteration delivering a working software build that incrementally adds features.
2. Adaptability: Agile recognizes that each project is unique and adapts the development approach accordingly to best suit the project's requirements.
3. Customer collaboration: Continuous customer interaction is essential to understand and incorporate evolving product requirements.
4. Rapid delivery: Agile emphasizes rapid delivery of working software, allowing for early feedback and course correction.
5. Self-organization and teamwork: Agile promotes self-organization and motivation within cross-functional teams, fostering collaboration and synergy.
6. Emphasis on working software: Rather than relying solely on documentation, Agile prioritizes demoing working software as the primary means of communication with customers.

* **When we use agile model?**

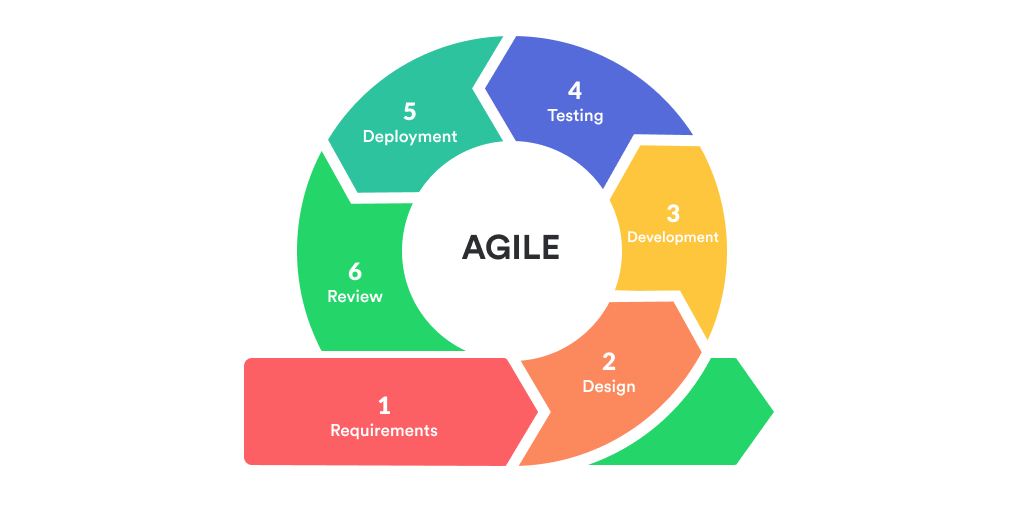
The Spiral model is a risk-driven software development approach that combines elements of both waterfall and iterative development models. It is typically used when a project has significant risk factors and a high degree of uncertainty. Here are some key points about the Spiral Model:

1. Risk management: The Spiral Model incorporates risk analysis and mitigation throughout the software development process.
2. Iterative approach: The development process progresses in iterations, with each iteration refining and adding new functionality.
3. Prototyping: The Spiral Model encourages the use of prototyping to gather feedback and validate design decisions.
4. Phases: The model consists of multiple phases, including planning, risk analysis, engineering, and evaluation. Each iteration focuses on these phases.
5. Feedback and evaluation: The project continually gathers feedback and evaluates the software to make informed decisions for subsequent iterations.

* **Pros and Cons about agile model**

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| 1. **Risk management:** The model emphasizes risk analysis and mitigation, reducing the chances of major failures. | 1. **Complexity:** The Spiral Model can be more complex and time-consuming to implement compared to other models. |
| 1. **Flexibility:** The Spiral Model accommodates changing requirements and can adapt to evolving project needs. | 1. **Resource-intensive:** The emphasis on risk analysis and prototyping may require additional resources and expertise. |
| 1. **Feedback-driven:** Frequent evaluation and feedback loops ensure that the product aligns with stakeholder expectations. | 1. **Lack of transparency:** The model's iterative nature may make it challenging to estimate project timelines and costs accurately. |
| 1. **Iterative refinement:** The model allows for incremental development and improvement over successive iterations. | 1. **Documentation overhead:** The Spiral Model typically requires comprehensive documentation, which can be time-consuming and resource-intensive. |

* **Agile model structure**



**3.3 SDLC chosen Method – Waterfall Model**

Our teams have decided to choose the waterfall model for this project due to several factors :

1. The project description provided a comprehensive understanding of the project's background, scope, and requirements.
2. The features and deliverables were well-defined
3. The waterfall model aligns with this need for a clear understanding of project requirements from the outset.

By choosing the waterfall model, we can ensure that the project remains focused on meeting the predefined requirements and scope. The well-defined nature of the waterfall model allows for a structured approach, minimizing the need for frequent changes and iterations. This approach provides clarity and stability throughout the development process, ensuring that the project stays on track to meet its goals within the specified timeframe.

**Technologies:**

**Front-end Tech**

|  |  |  |
| --- | --- | --- |
| Technologies | Pros | Cons |
| HTML | Simple and easy to learn | Limited interactivity and dynamic functionality |
| Widely supported and compatible | Requires manual handling of state and UI updates |
| Good for static content | Steeper learning curve for complex layouts |
| React JS | Efficient and fast rendering | Steeper learning curve for beginners |
| Component-based architecture for reusability | Requires additional tools and configuration for complex applications |
| Large ecosystem and community support | Not suitable for small and simple projects |
| Vue JS | Easy integration with existing projects | Smaller ecosystem compared to React and Angular |
| Simple syntax and gentle learning curve | Limited enterprise-level support |
| Flexible and scalable | May have performance issues with larger applications |
| Angular JS | Full-featured framework with extensive tooling | Steeper learning curve compared to other front-end frameworks |
|  | Strong support for large-scale applications | Requires adherence to Angular conventions and patterns |
|  | Two-way data binding for automatic UI updates | Limited performance optimizations for smaller projects |

**Back-end Tech.**

|  |  |  |
| --- | --- | --- |
| Technologies | Pros | Cons |
| Laravel | Elegant syntax and developer-friendly | Requires PHP as the programming language |
| Robust ecosystem and community support | Limited scalability for extremely high-traffic applications |
| Built-in ORM for database interaction | Slower performance compared to some other frameworks |
| Spring Boot | Highly scalable and enterprise-ready | Requires knowledge of Java programming language |
| Excellent support for microservices architecture | Steeper learning curve for beginners |
| Integrated security and dependency management | Initial setup and configuration can be complex |
| Java | Platform-independent and widely used | Verbose syntax compared to some other languages |
| Strong ecosystem and extensive libraries | Requires knowledge of object-oriented programming concepts |
| Excellent performance and scalability | Longer development time for certain tasks |
| Express JS | Minimalistic and flexible framework for building APIs | Requires additional modules for advanced features and functionalities |
|  | Middleware support for handling HTTP requests | Lacks some built-in features compared to more opinionated frameworks |
|  | Efficient and lightweight | Limited support for database interaction out of the box |

**DATABASE:**

|  |  |  |
| --- | --- | --- |
| Technologies | Pros | Cons |
| MySQL | High performance and scalability | Lacks some advanced features of other databases |
| Wide community support and extensive documentation | Limited support for NoSQL capabilities |
| Good for relational data and complex queries | May have limitations in handling large datasets |
| PostgreeSQL | Feature-rich and highly extensible | Requires more resources compared to some other databases |
| Excellent support for complex queries and advanced SQL | Steeper learning curve for beginners |
| High data integrity and reliability | May have limited support for certain programming languages |
| SQLite | Lightweight and easy to set up | Not suitable for large-scale deployments |

**IDE:**

|  |  |  |
| --- | --- | --- |
| Technologies | Pros | Cons |
| VS Code | Lightweight and fast editor with a wide range of extensions | Lack of some advanced features found in full IDEs |
| Excellent support for web development and JavaScript ecosystem | Steeper learning curve for beginners |
| Intuitive user interface and customizable settings | Requires additional configuration for certain languages |
| IntelliJ IDE | Robust and feature-rich IDE for Java and JVM languages | Heavier and slower compared to lightweight code editors |
| Advanced code analysis and debugging tools | Steeper learning curve for beginners |
| Intelligent code completion and refactoring capabilities | Consumes more memory and resources than lightweight editors |
| Seamless integration with build tools and version control | Paid version (Ultimate Edition) includes additional features |
| SQLite | Specialized IDE for Spring Framework development | Limited support for other programming languages |
| Built-in tools for Spring Boot configuration and deployment | Steeper learning curve for beginners |
| Integrated support for testing and debugging Spring projects | Not as feature-rich as general-purpose IDEs |
| Free and open-source | Relatively smaller community compared to other IDEs |

**System Requirements:** enlist here the system support needed to implement the user requirements

**Manpower requirement**

|  |  |
| --- | --- |
| Team Member’s Name | Task Assigned |
| Abdul Rahman Shalehudin | Handle the functionality of Driver |
| Ajri Muhammad Sidiq | Handle the functionality of Volunteer |
| Asep Supriyadi | Handle the functionality of Administrator |
| Darren Farrell Andrian | Handle the functionality of Member and Donor |
| Syukur Sidiq Nur Alam | Handle the functionality of Partner |

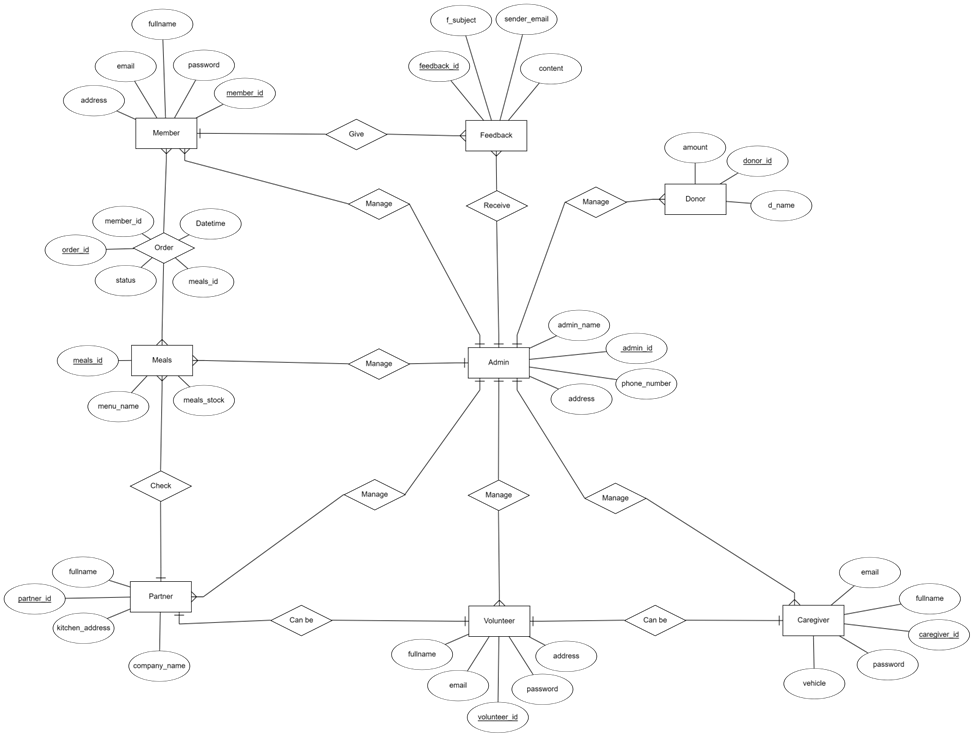
**Hardware Requirement**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Hardware** | | |
| **Type** | **Service** | **Client** |
| **1** | Processor | Intel i5 2.6 GHZ | Intel-i3 our above |
| **2** | Memory | 12GB | 4GB our above |
| **3** | Hard Drive | 4TB SSD NVME | 1TB our above |
| **4** | Network | Gigabit Network | Wifi / Internet Providers |

**Software Requirement**

|  |  |  |
| --- | --- | --- |
| **No** | **Software** | |
| **Software Device** | **Functionality** |
| **1** | Window 10 | Operating System |
| **2** | Visual code, eclipse,spring boot | Web Creation |
| **3** | Tomcat | Web server |
| **4** | MySQL | Databases |
| **5** | Figma | Design |
| **6** | Google,Chrome,microsoft edge | Web Browser |

**Db Design: ERD Diagram and EERD**

****

****

**4.0 Project Team:**

4.1 Team member’s information

|  |  |  |  |
| --- | --- | --- | --- |
| **Partner** | **Learner ID** | **Full Name** | **Group ID** |
| STTB | bdse-0922-079 | Asep Supriyadi | Bdse-0922 |
| STTB | bdse-0922-077 | Ajri Muhammad Sidiq | Bdse-0922 |
| STTB | bdse-0922-089 | Syukur Sidiq Nur Alam | Bdse-0922 |
| STTB | bdse-0922-075 | Abdul Rahman Shalehudin | Bdse-0922 |
| LASG | bdse-0922-051 | Darren Farrell Andrian | Bdse-0922 |

* 1. Team member’s roles and responsibilities

|  |  |  |
| --- | --- | --- |
| **Name** | **Module** | **Details Tasks** |
| Abdul Rahman Solehudin | 1. Handle the functionality of Partner | * Implement driver registration functionality, allowing individuals to register as drivers for Meals on Wheels meal deliveries. * Develop a secure login system for drivers to access their accounts. * Create a view for drivers to see the assigned food delivery schedule and pickup locations. * Implement the ability for drivers to update the delivery status (e.g., picked up, on the way, completed) for each meal delivery. * Integrate a notification system to inform drivers about new delivery assignments or changes to existing deliveries. * Implement a feature for drivers to view and manage their profile information, including updating contact details or vehicle information. |
| Ajri Muhamad Sidik | 1. Handle the functionality of Volunteer | * Develop the volunteer registration functionality, allowing individuals to register as volunteers. * Implement a login system for volunteers to access their accounts securely. * Based on the roles they choose during registration (caregiver or rider), provide access to their respective roles. * Create a view for volunteers as a driver to see the assigned food delivery schedule. * Implement the ability for volunteers as a driver to update the delivery status (pick up, on the way, completed). * Develop an edit or view profile page for volunteers to manage their personal information. * Implement the ability for volunteers that have a kitchen to process the ordered meals according to Meals on Wheels specifications. |
| Asep Supriyadi | 1. Handle the functionality of Administrator | * Develop a login system with role-based access control, allowing only authorized administrators to access the system. * Create a management interface for administrators to handle donation management, including viewing and managing donations. * Implement member management functionality, allowing administrators to manage member information, registrations, and profiles. * Develop menu/order management functionality, enabling administrators to create, modify, and delete meal options. * Implement driver management functionality, allowing administrators to assign drivers or riders to delivery tasks. * Create an information system management interface for administrators to effectively manage the overall system. |
| Darren | 1. Handle the functionality of Member and Donor | * Develop the member registration functionality, allowing qualified adults to register as members. * Implement a secure login system for members to access their accounts. * Create a view for members to order and view meals based on their requirements. * Implement the ability for members to update their profiles and track their meal deliveries. * Develop the donation functionality, allowing individuals or organizations to register as donors/supporters. * Implement a login system for donors to access their accounts securely. * Create a view for donors to edit their profiles and send donations to support Meals on Wheels activities. |
| Syukur Sidiq Nur Alam | 1. Handle the functionality of partners | * Implement partner registration functionality, allowing individuals or organizations to register as partners. * Develop a login system for partners to access their accounts securely. * Create a view for partners to see the meals that have been ordered. * Implement the process to process the ordered meals according to Meals on Wheels specifications. |

**Module ID\_Module owner name**

4.3 Task (Role: Partner):

1. Meal Preparation:

- Follow the provided menu guidelines and recipes to prepare meals according to the specified standards and portion sizes.

- Pay attention to dietary restrictions or special requests provided by the members and accommodate them accordingly.

- Maintain cleanliness and adhere to food safety standards during the food preparation process.

2. Quality Control:

- Conduct quality checks on the prepared meals to ensure they meet the required standards of taste, presentation, and hygiene.

- Address any issues or concerns regarding the meals promptly and take appropriate measures to rectify them.

- Seek feedback from members regarding the quality and satisfaction with the meals.

3. Collaboration:

- Coordinate with the admin, riders, and volunteers to ensure smooth coordination and timely delivery of meals.

- Communicate any changes or updates regarding the meal preparation process promptly to the relevant stakeholders.

- Collaborate with the admin and riders to ensure accurate meal packaging and delivery instructions.

4. Inventory Management:

- Maintain an inventory of ingredients and supplies needed for meal preparation.

- Regularly monitor ingredient quantities and inform the admin when supplies need to be restocked.

- Ensure proper storage and organization of inventory to maintain freshness and quality.

5. Meal Packaging:

- Package the prepared meals securely and appropriately, considering factors such as temperature control and spillage prevention.

- Label the packaged meals with relevant information, including the meal name, date of preparation, and any heating instructions if required.

6. Performance and Feedback:

- Continuously strive for improvement in meal preparation and quality.

- Seek feedback from members regarding the meals and incorporate suggestions for enhancement.

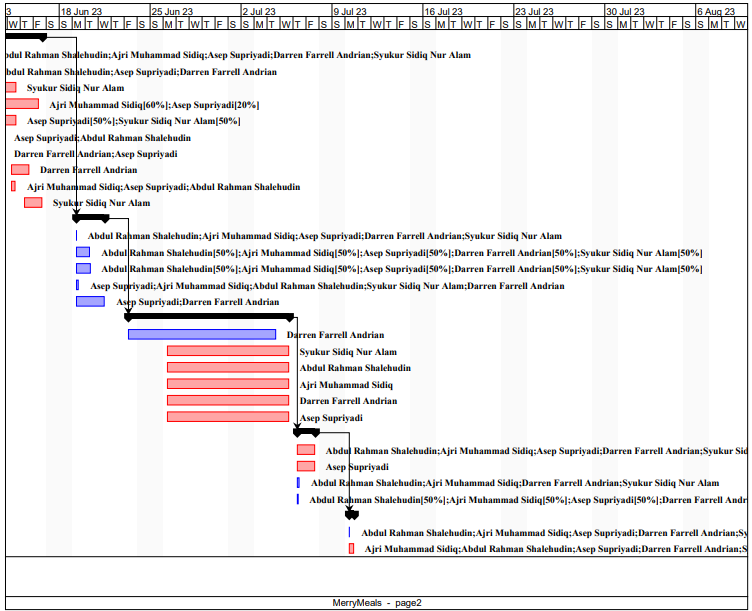
- Collaborate with the admin and members to address any concerns or issues related to meal quality or delivery.

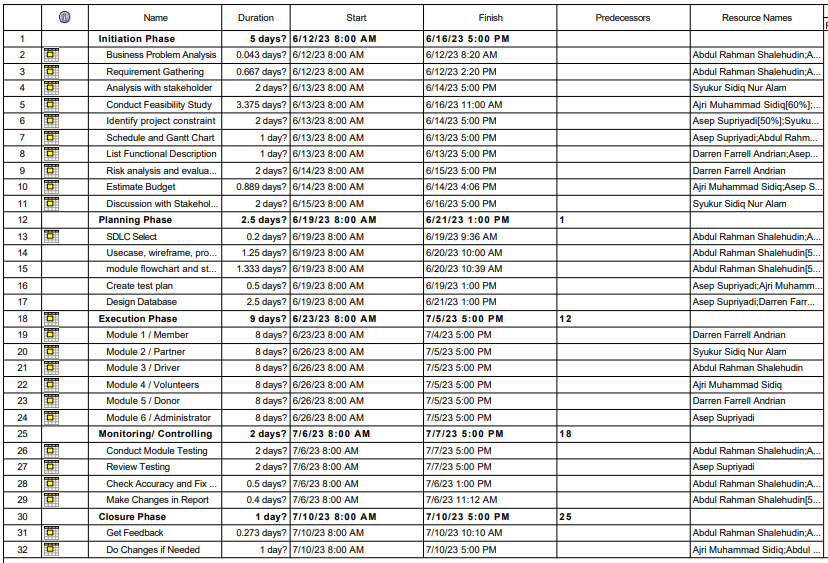
**4.4 Tools and technology**

1. Front-End Tech: HTML, CSS, JS, Boostrap / React Boostrap
2. Back-end Tech.: Spring Boot
3. Database: MySQL
4. IDEs: VSCode for both Front-End and Back-End
5. Tools: Figma, draw.io, Microsoft, ERDplus

**5.0 Project planning:**

**5.0.1 WBS and Gantt chart-group level**





**Initiation phase:**

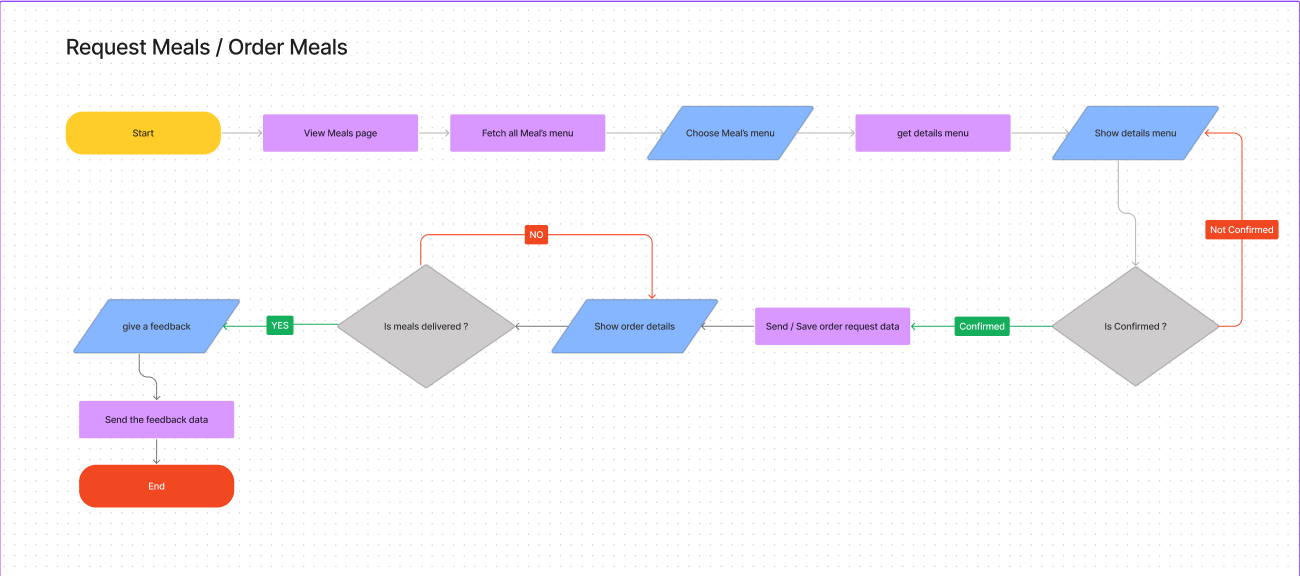
1. Business Challenge Assessment: The project focuses on addressing the operational challenges faced by Meals on Wheels in delivering meals to individuals unable to cook for themselves due to age, illness, or disability. The aim is to develop a software solution that streamlines their operations and improves service delivery to members.
2. Requirement Gathering and Analysis: Extensive analysis and discussions with stakeholders are conducted to gather software requirements. Stakeholders involved include members, caregivers, partners, volunteers, donors/supporters, and administrators. Requirements cover member registration, caregiver support, partner collaboration, volunteer management, donation handling, menu planning, meal delivery management, food safety, need assessment, and management information system.
3. Feasibility Evaluation: A thorough feasibility study assesses the viability and potential success of the proposed software solution. It examines technical feasibility, operational feasibility, economic feasibility, and legal compliance. Technical feasibility evaluates hardware, software, and network compatibility. Operational feasibility ensures alignment with organization needs. Economic feasibility analyzes costs and benefits. Legal compliance ensures adherence to applicable laws.
4. Software Requirements Specification (SRS): An SRS document is created, detailing the functional and non-functional requirements based on the gathered information and feasibility study. The SRS serves as a reference for the development team, ensuring a clear understanding of implementation needs.
5. Project Aim, Objectives, and Scope: The project aims to design and develop a software solution that optimizes meal delivery operations and meets member needs. Objectives include requirements analysis, software development, application implementation, performance evaluation, and identification of improvement areas. The scope covers member and caregiver registration, partner and volunteer management, fundraising, menu planning, meal delivery management, food safety, need assessment, and management information system.
6. Stakeholder Involvement: Key stakeholders include the Merry on Wheels application sponsor, Mr. David (Project Manager at Unity One Solution), Merry Meals charity organization members, and Merry Meals project team members. Their active participation ensures project success.
7. Milestones Planning: A milestones plan is devised, outlining major project milestones, deliverables, and timelines. This plan facilitates progress tracking, resource allocation, and adherence to project objectives, considering dependencies and critical path activities.
8. Risk Assessment: A comprehensive risk analysis is conducted to identify and mitigate potential risks. Risks may involve technical challenges, resource limitations, changing requirements, stakeholder conflicts, or external factors. Proactive risk management minimizes their impact, increasing project success probability.

**Planning**:

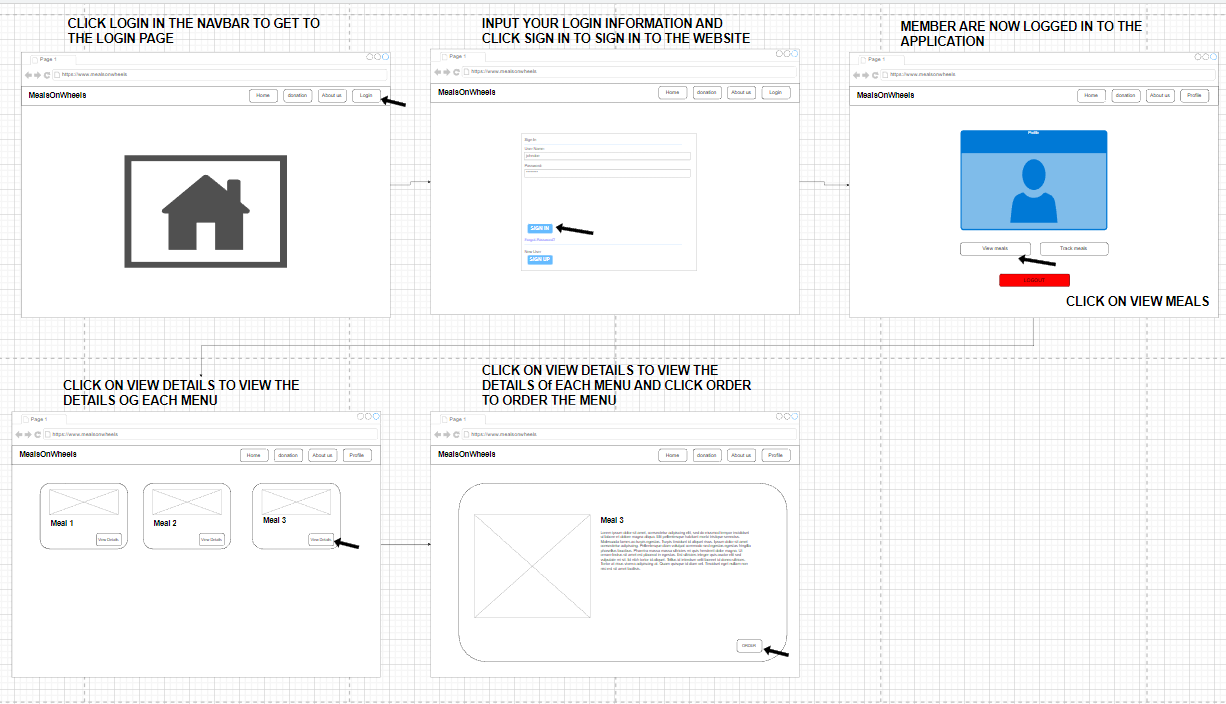
Design Plan:

**Flowchart**

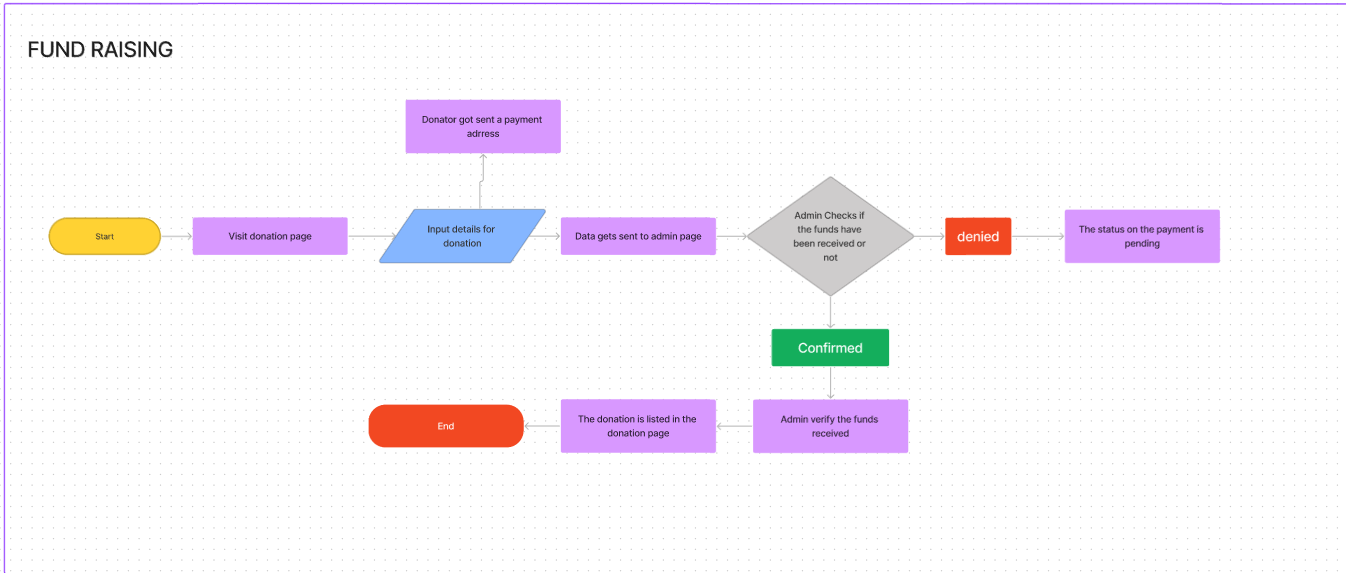
**Member Flowchart**  
 - Ordering food

****

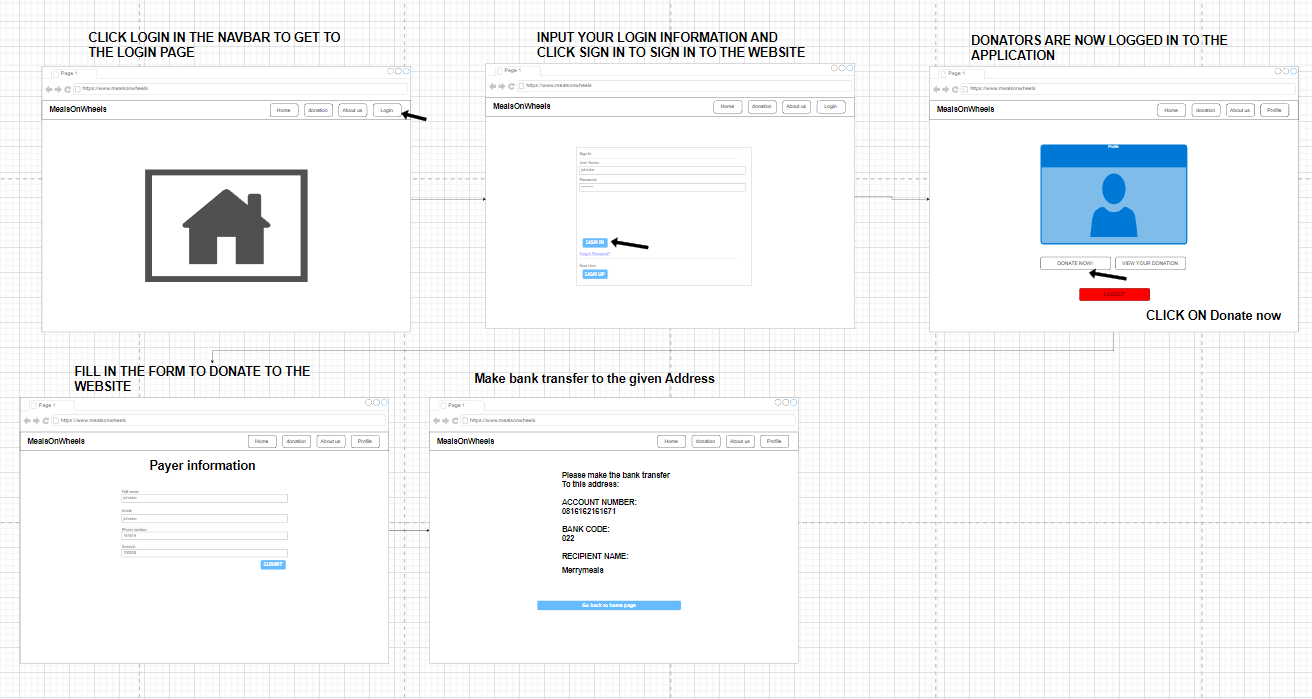
Storyboard

****

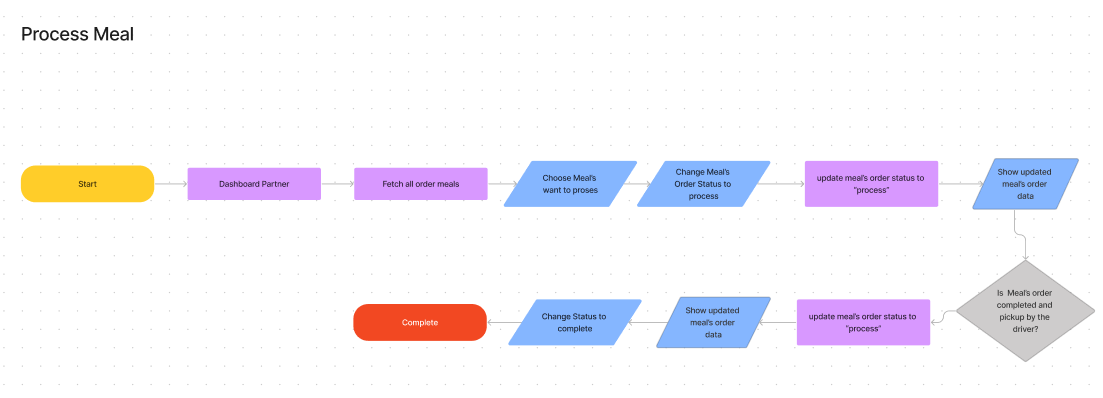
**Donor flowchart**

****

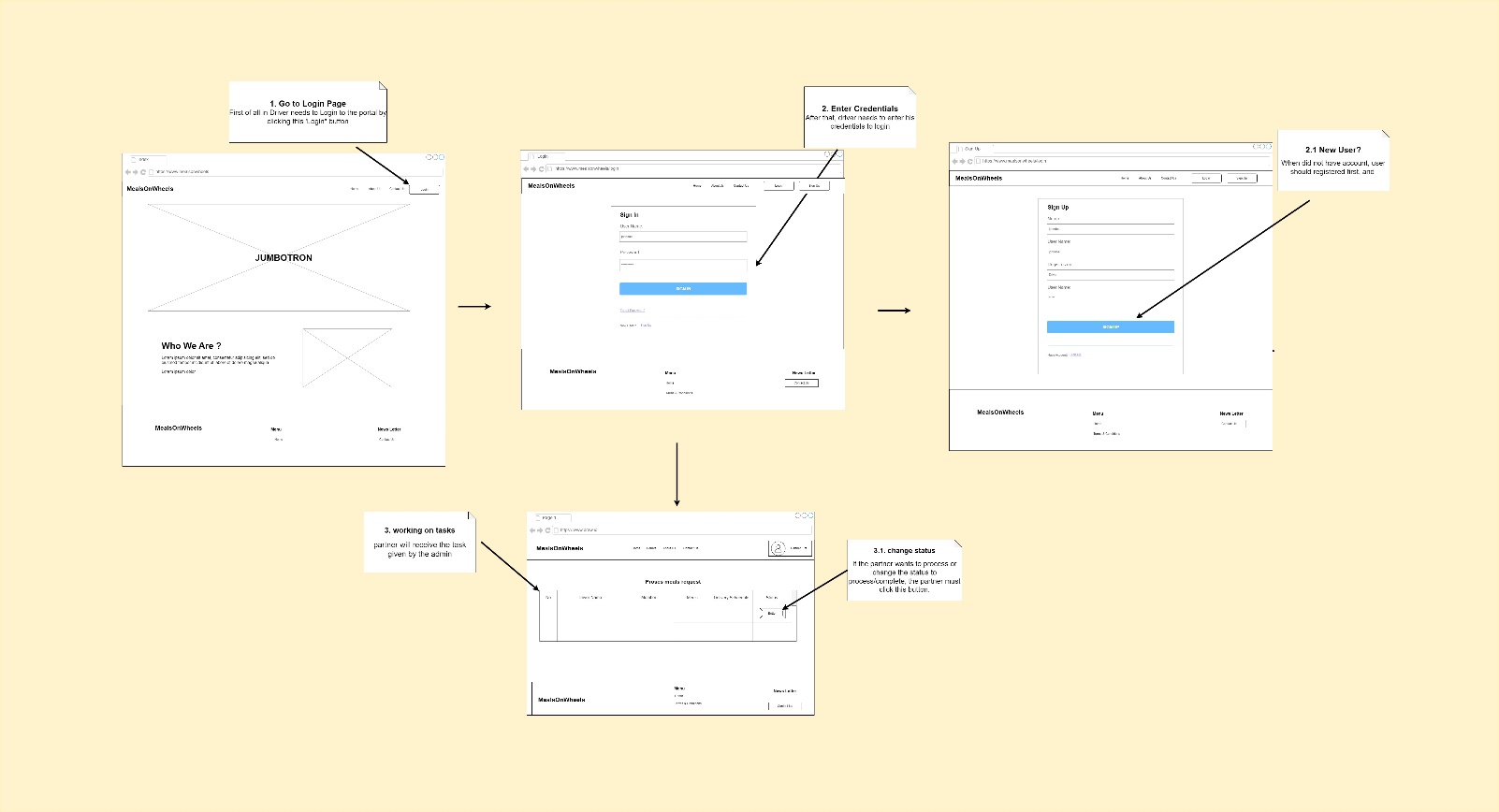
Storyboard

****

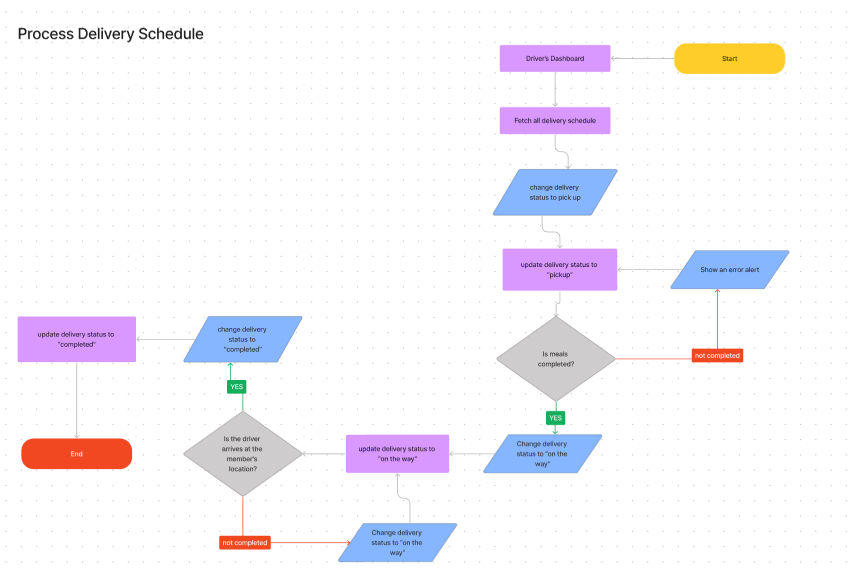
**Partner flowchart**

****

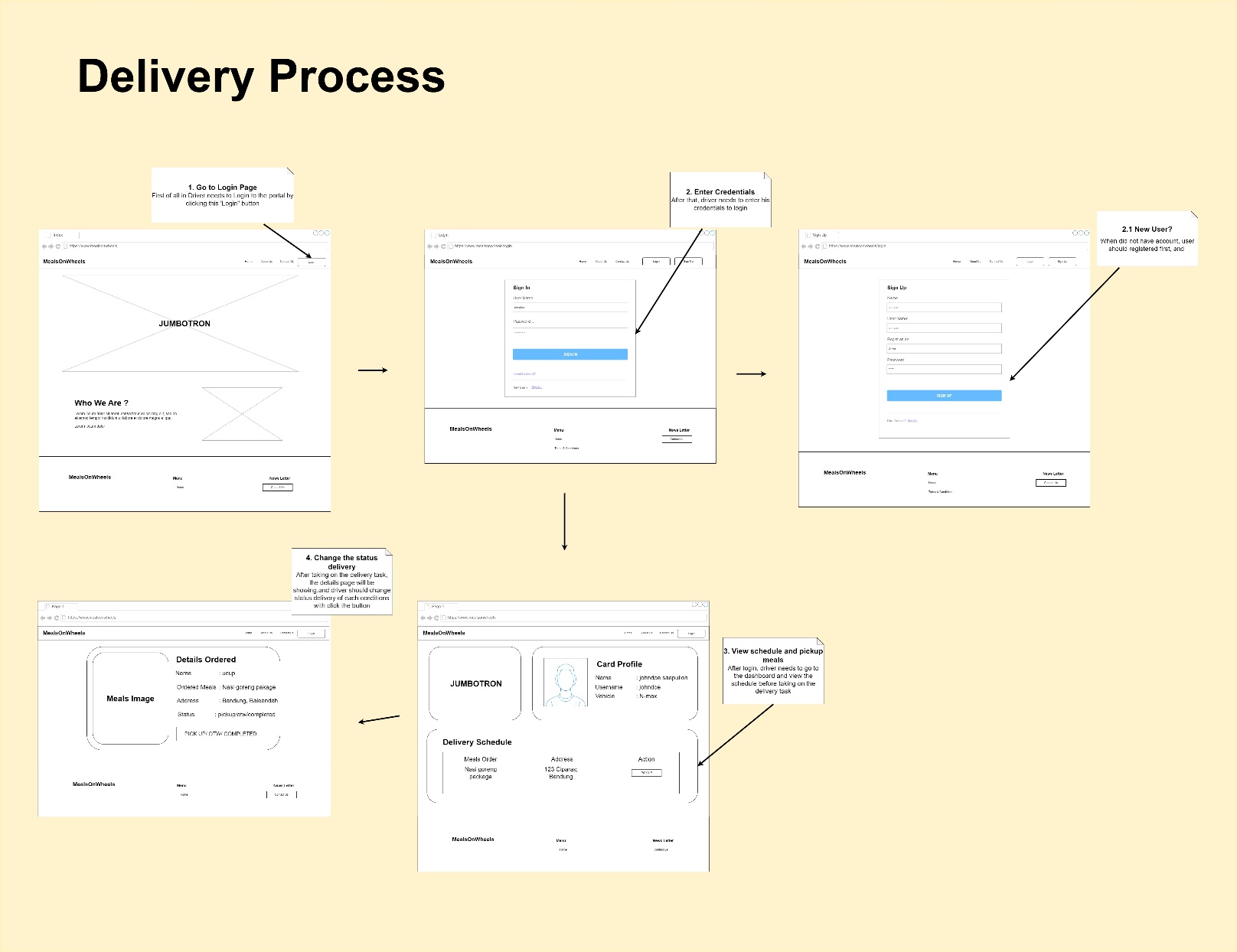
**Storyboard**



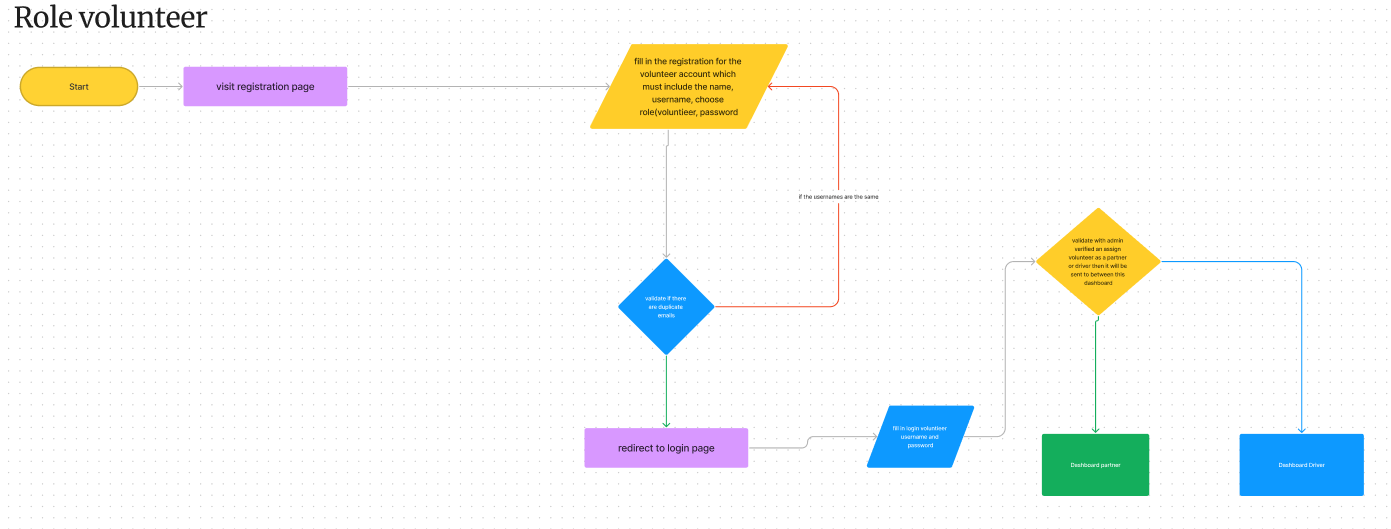
**Driver flowchart**

****

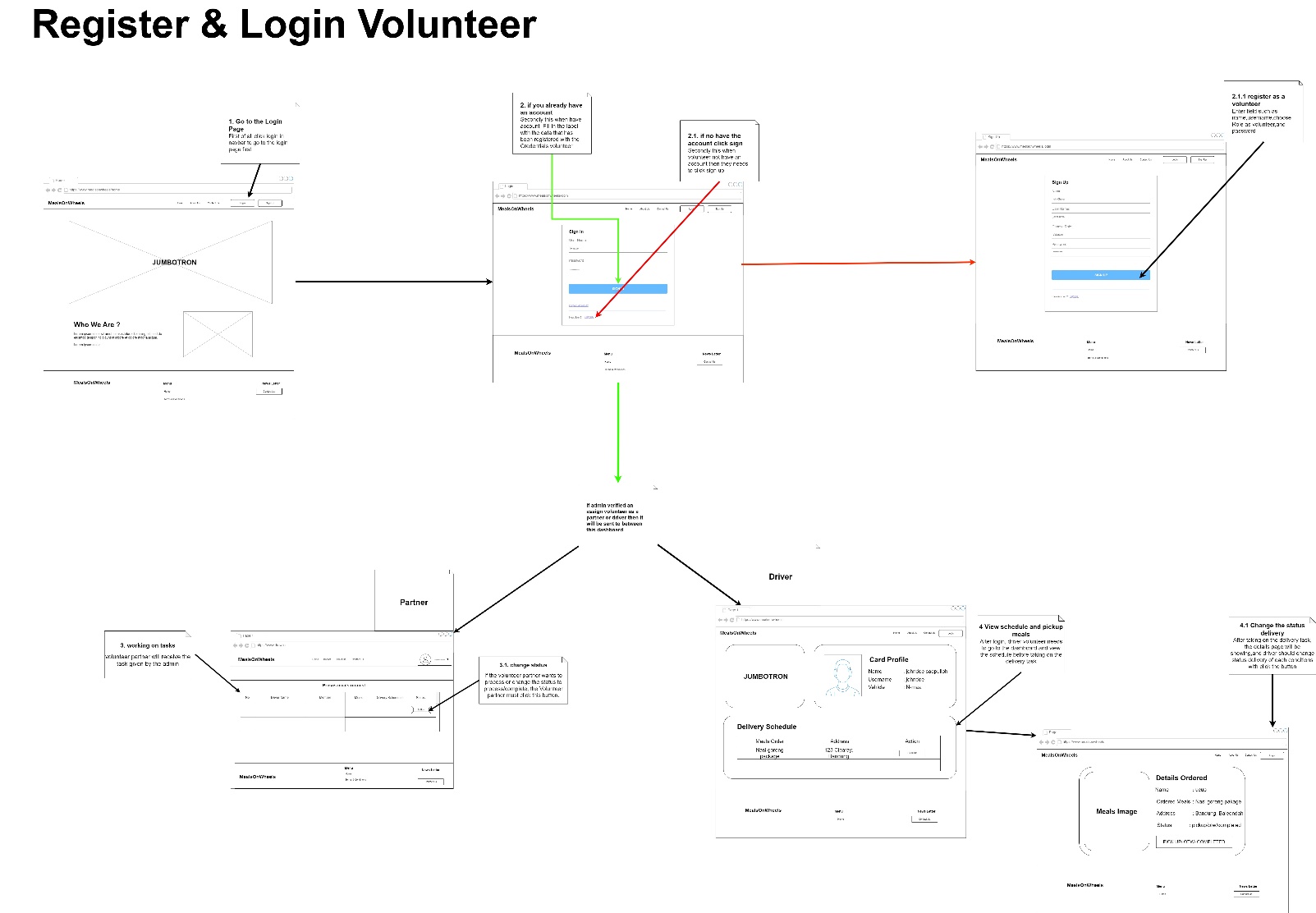
**Storyboard**

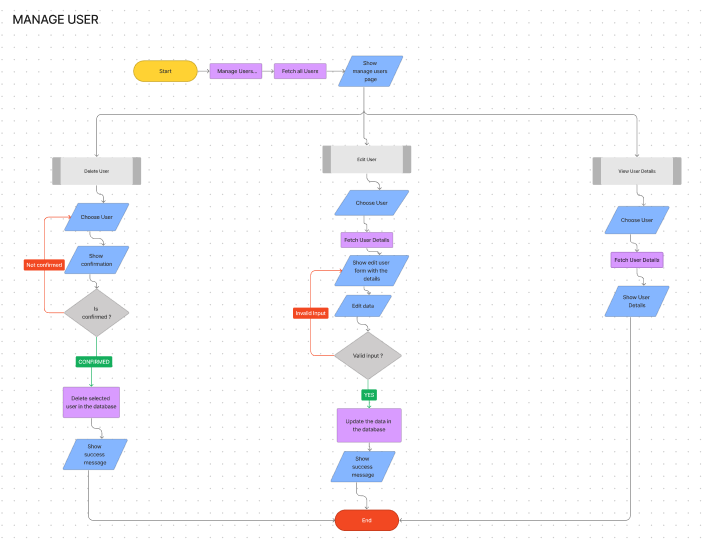


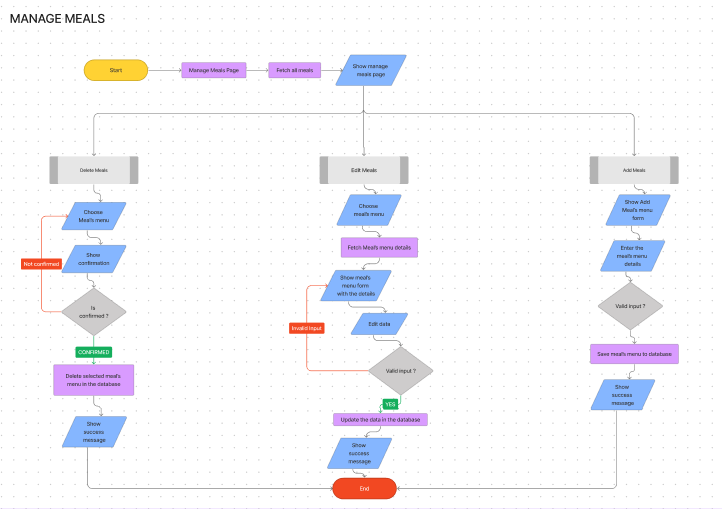
**Volunteer flowchart**

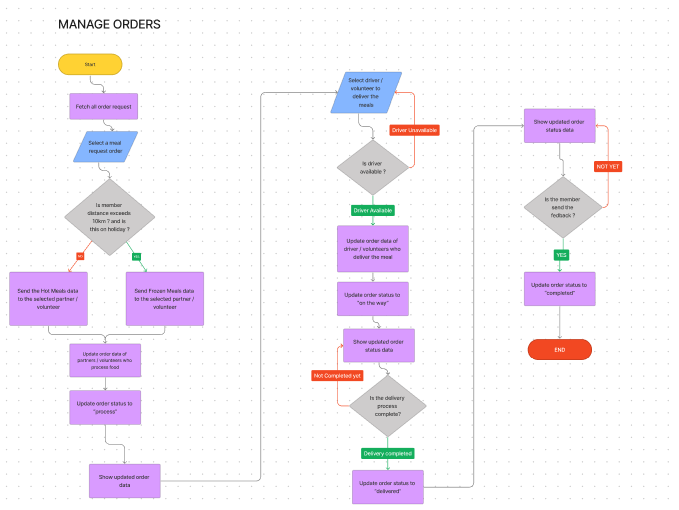
****

**Storyboard**

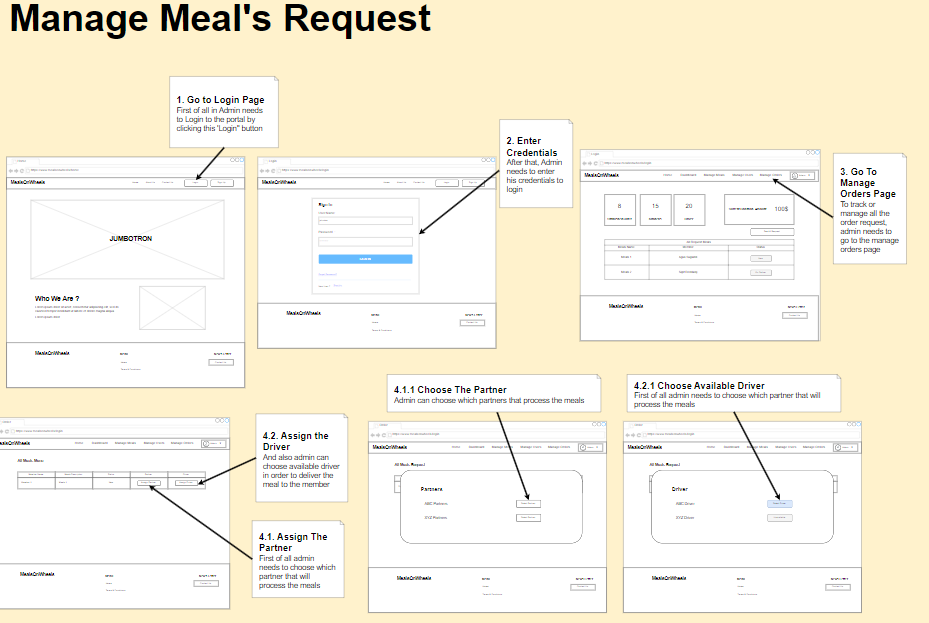


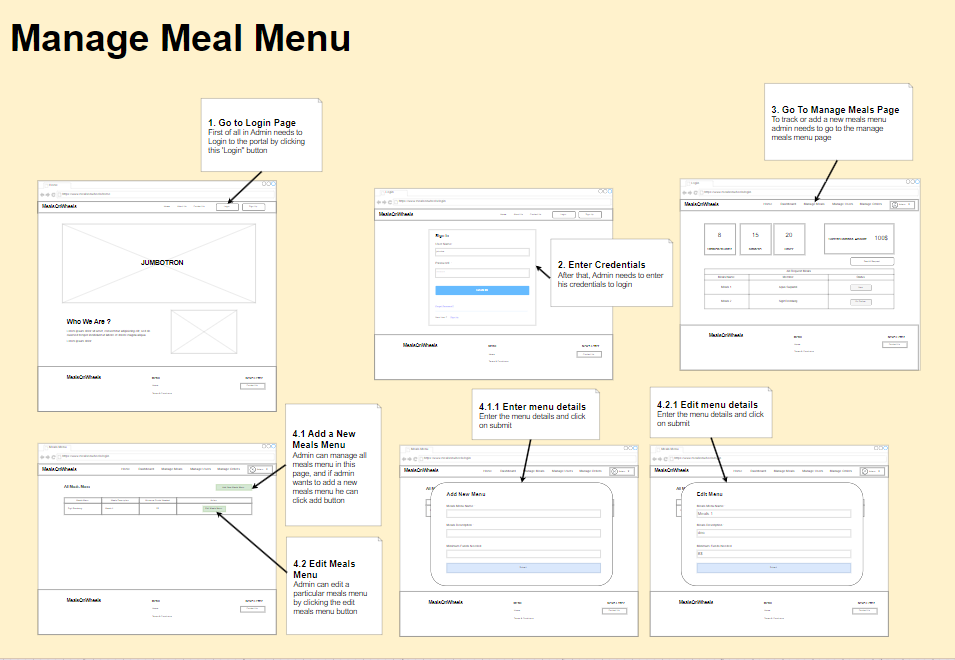
**Admin flowchart  
**

****

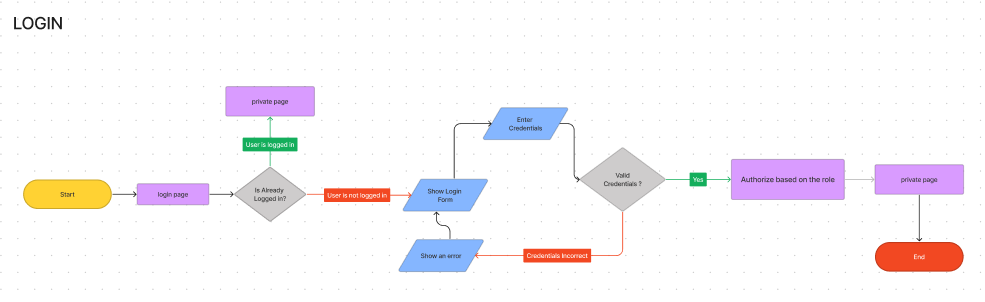
****

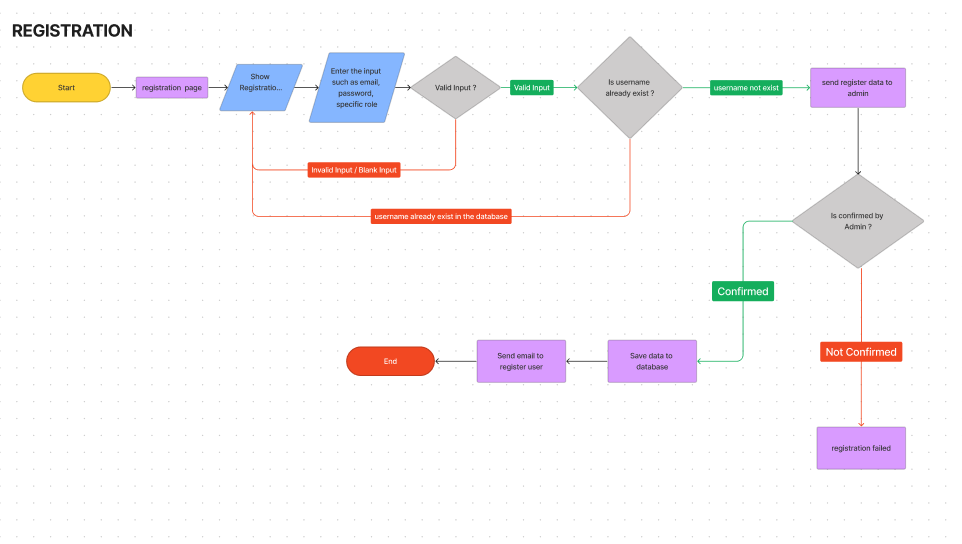
**Storyboard**

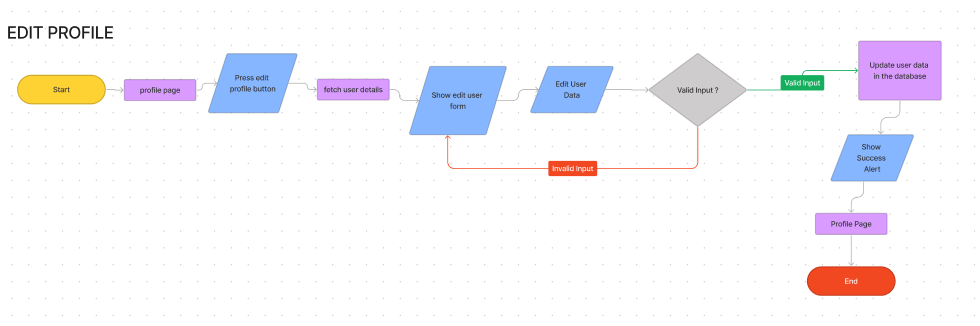
****

****

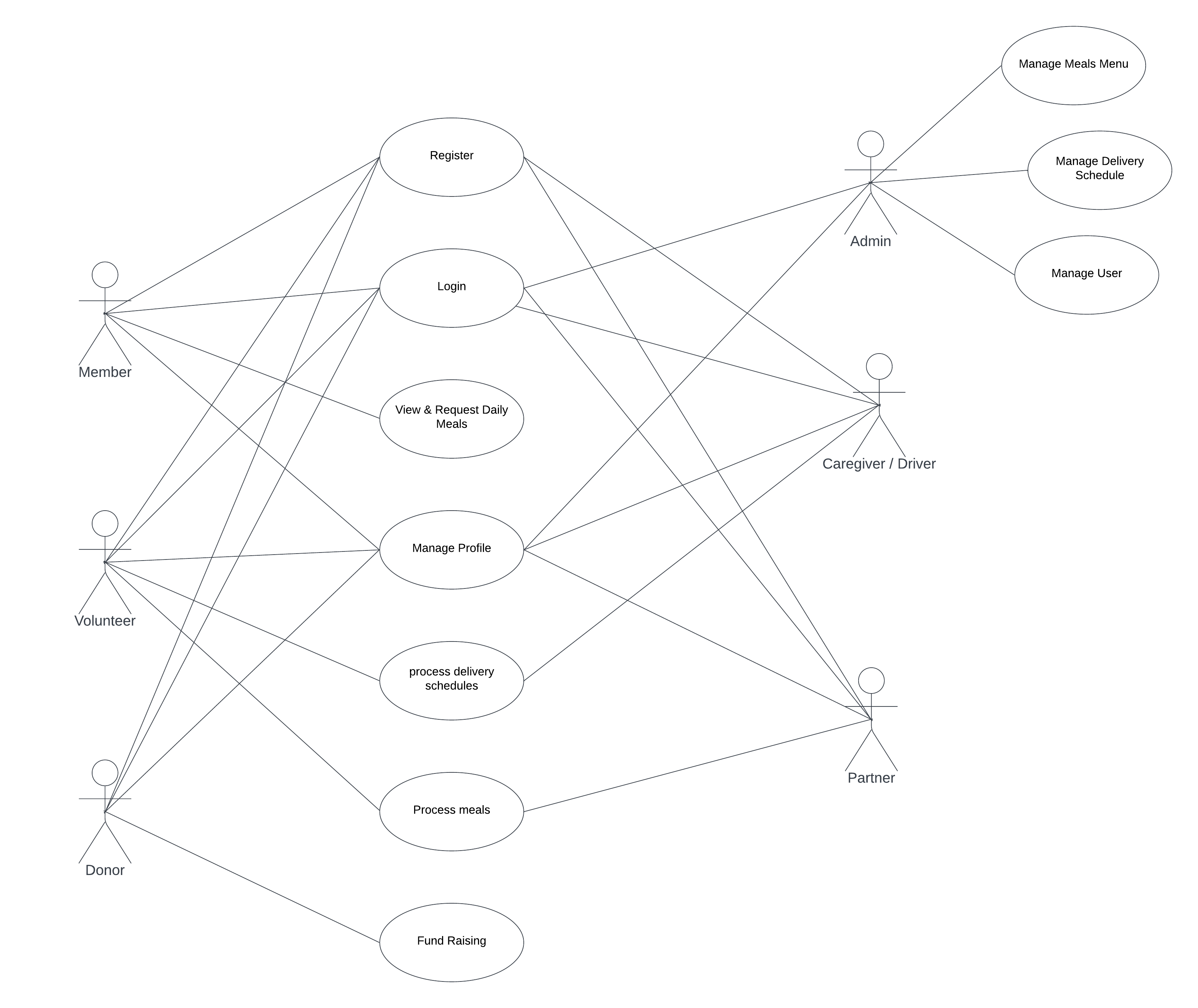
**General function flowchart**

****

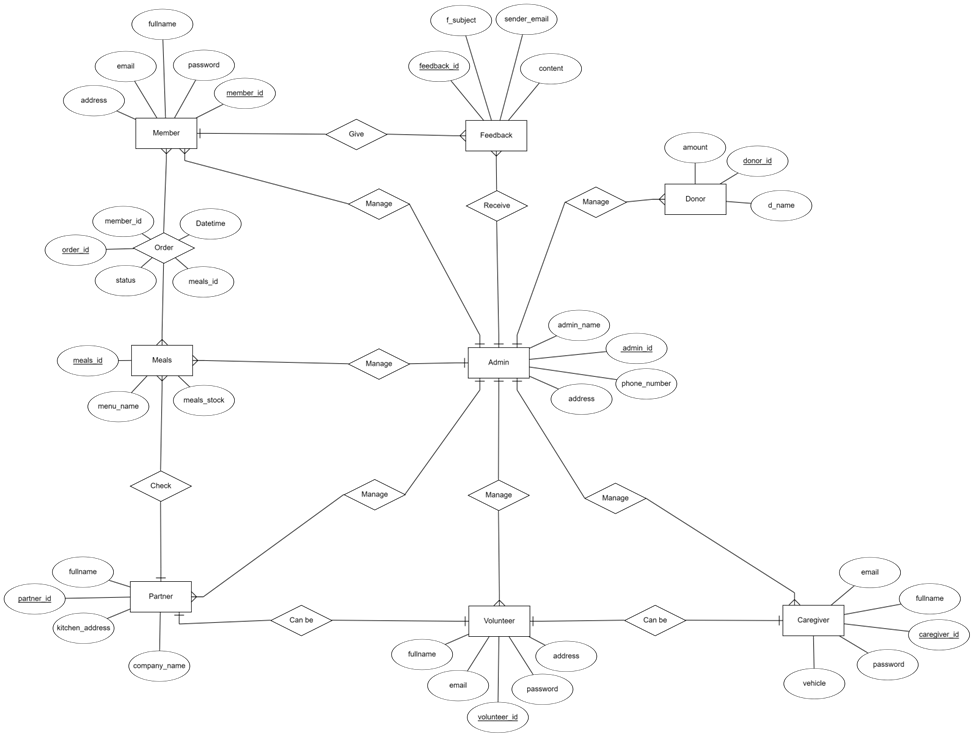
****

****

**UML use case**

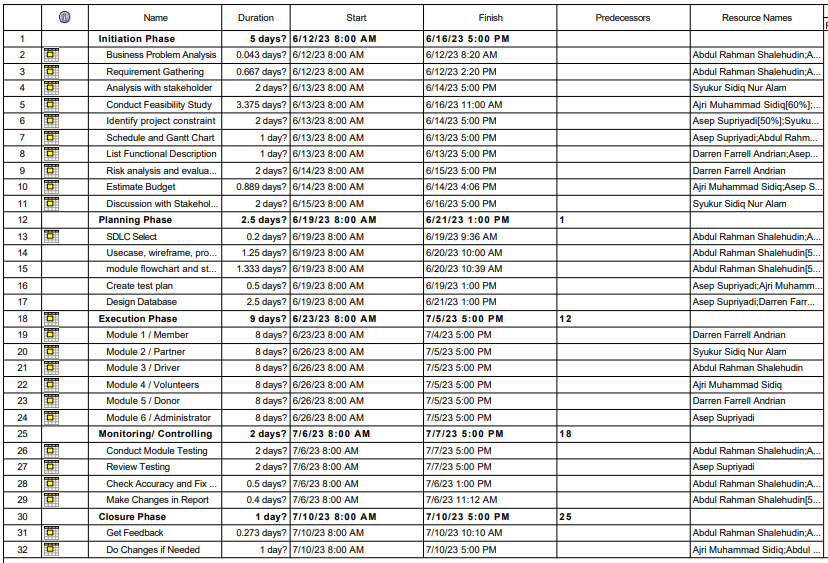


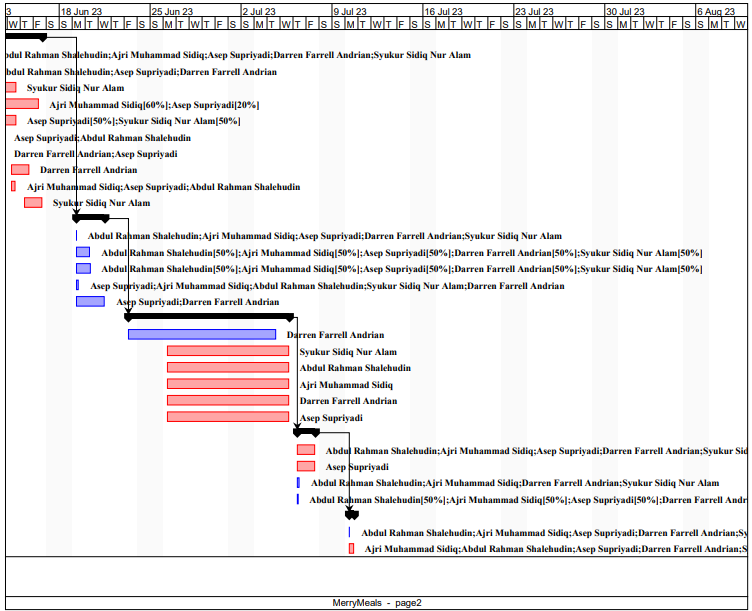
**ERD EERD**

****

****

**Schedule plan: WBS, Gantt chart**





**Communication Phase**: prepared and deliver the stakeholder PPt

**Execution phase**: module implementation

**Monitoring/ controlling:** Testing: Unit/functional testing, UAT, Performance testing

**Closure:** deploy project, release resources, finish project documentation

5.0.2 Budget analysis

|  |  |  |
| --- | --- | --- |
| List of cost needed | Description of activity | Total cost in month |
| Requirement gathering | Activities to gather and document project requirements, including meetings with stakeholders, interviews, research, and analysis. | $400 |
| Hardware | Procurement and installation of a 4TB hard drive for data storage and processing requirements. | $1.500 |
| Software | Purchase or licensing of necessary software tools and applications required for the project development and execution. | $1.200 |
| Hosting | Subscription or rental fees for hosting services, including web servers or cloud infrastructure to deploy and host the application. | $200 |
| Maintenance | Ongoing support, updates, bug fixes, and enhancements to ensure the system's functionality, security, and performance. | $800 |
| Design Implementation | Implementation of the design elements, including user interface (UI) design, user experience (UX) design, and visual design. | $3.000 |
| Human resource | Costs associated with hiring and retaining personnel for project development and management, including salaries, benefits, and training. | $3.000 |
| Marginal / urgent cost | Additional costs that may arise unexpectedly during the project's lifecycle, such as hardware or software upgrades, unforeseen requirements, or urgent fixes. | $400 |
| Total cost | Total cost The sum of all the costs listed above, representing the total expenditure required for the project. | $21.000 |
| Total cost expected |  | $20.000 |

5.0.3 Communication matrix:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Speaker** | **Audience** | **Purpose** | **Method** | **Frequency** |
| Team Member | Mr. David | Development progress | MS Teams Meeting & Face-to-Face | Twice a week |
| Mr. David | MerryMeal Org | Explain current development process and project alignment with stakeholder expectations | MS Teams Meeting & Face-to-Face | Once a week |

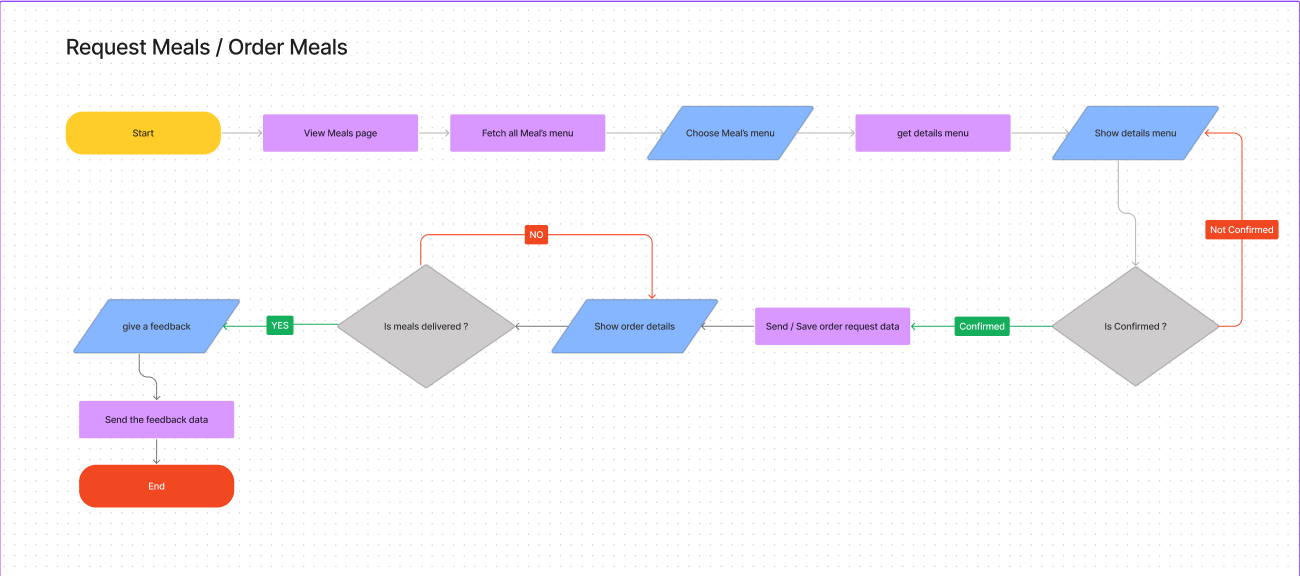
**5.0.4**

**Module Test Plans: :**

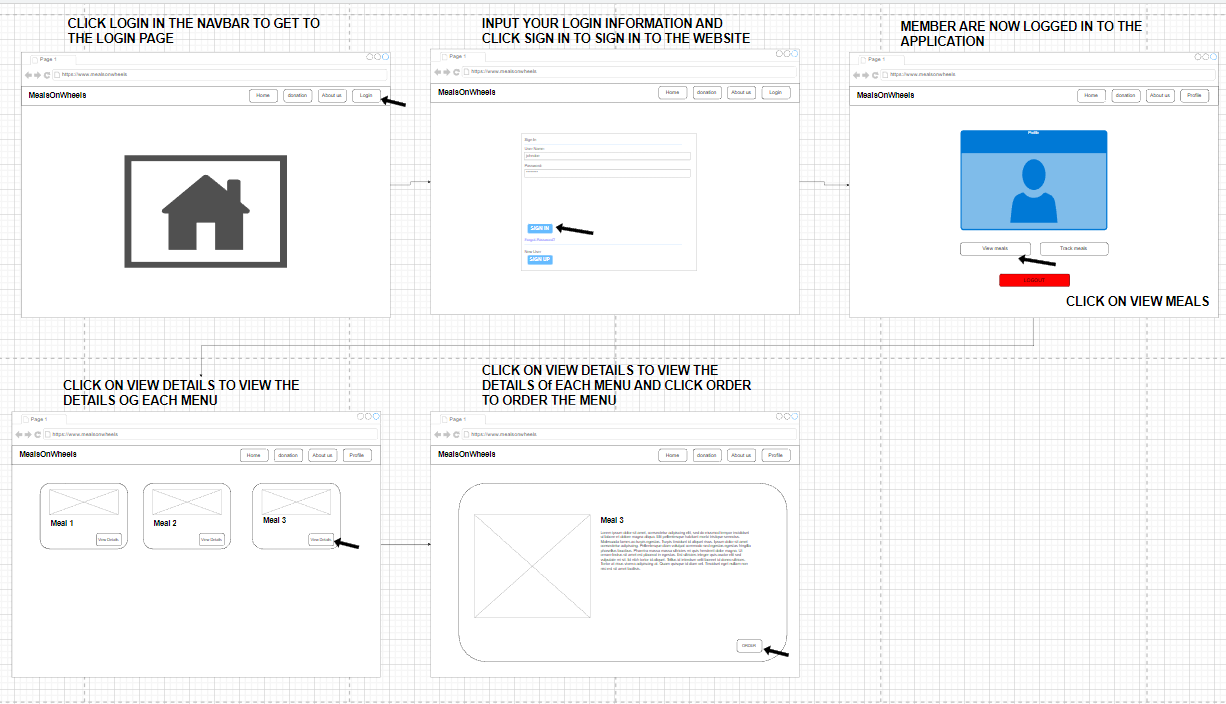
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Risk Id | Test Scenario ID | Test Scenario | Test Objective | Test Priority | Risk | Technique | Expected Result |
| 1 | TS001 | Validating Registration as a Partner funtionality | For Partners to successfully register with their credentials. | High | partner cannot create an account if there is a similarity in the email | Unit Testing | After registering user will see the succes message and their credentials saved in the database |
| 2 | TS002 | Validating Login as a Partner functionality | To check Partner credentials are existing in the system. | High | Partner is not able to log in to the portal | Unit Testing | After login, user will reach to the home page |
| 3 | TS003 | Ensure the functionality of the ordering process | to ensure that orders can be processed. | High | Partner not being able to proceed food | Unit Testing | After checking, junit should return a pass result because the selected food data can be processed.  Ensure Retrieval Menu functionality. |
| 4 | TS004 | Ensure the functionality of the Retrieval Menu | so that partners can see the daily menu | High | partners cannot see the daily menu | Unit Testing | after checking, junit should return a pass result because the daily menu can be seen. |
| 5 | UA001 | Ensure partner registration functionality. | To allow users playing the role of partners to successfully register using their credentials | High | Partners cannot create an account if there is an email similarity | User Acceptance Testing | After conducting the testing, UAT should return the appropriate result if partners can register successfully using their credentials. |
| 6 | UA002 | Ensuring the partner login functionality | To allow users playing the role of partners to log in to the portal using their credentials | High | Partners are not able to log in to the portal | User Acceptance Testing | After conducting the testing, UAT should return the appropriate result if partners can log in to the portal using their correct credentials. |
| 7 | UA003 | Ensuring the ordering process functionality. | To allow partners to successfully place orders. | High | Not being able to order food. | User Acceptance Testing | After conducting the testing, UAT should return the appropriate result if partners can successfully processed the order |
| 8 | UA004 | Ensuring the Retrieval Menu functionality | allow partners to view the daily menu that needs to be prepared. | High | Partners cannot see the daily menu. | User Acceptance Testing | After conducting the testing, UAT should return the appropriate result if partners can view the daily menu as expected. |
| 9 | PM001 | Testing the performance of Partner dashboard page | To utilize the page’s loading speed of the Partner dashboard page using Chrome DevTools | Extreme | Partner will have a difficulty for managing the application process properly | Performance Testing | After testing, The loading time speed should be above 80 – 90% or less than 1s. |
| 10 | CB001 | Ensure Login page maintain a consistent look and feel in different browser (Chrome, Edge, Firefox) | To check a consistent look and feel of the login pageacross different browser | High | The Partner may feel confused with the different look and feel in the login page so partner may have a difficulty for logging in to the portal | Compatibility Testing | After testing, Login page maintain a consistent look and feel across different browser |
| 11 | CB002 | Ensure Partner dashboard page maintain a consistent look and feel in different browser (Chrome, Edge, Firefox) | To check a consistent look and feel of the Partner dashboard page across different browser | High | The Partner may feel confused with the different look and feel in the dashboard page so Partner may have a difficulty tracking the order in the portal | Compatibility Testing | After testing, dashboard page maintain a consistent look and feel across different browser |
| 13 | PB001 | Ensure dashboard page maintain a consistent look and feel in different devices (laptop, iPad, phone) | To ensure all feature in the dashboard page is visible to the Partner in different devices | High | Partner may feel difficulty to chose a different option as the website are not able to adapt to their device of choice | Portability Testing | After testing, all dashboard page is visible to the Partner in different devices (laptop, iPad and phones) |

**6.0: Project Design:**

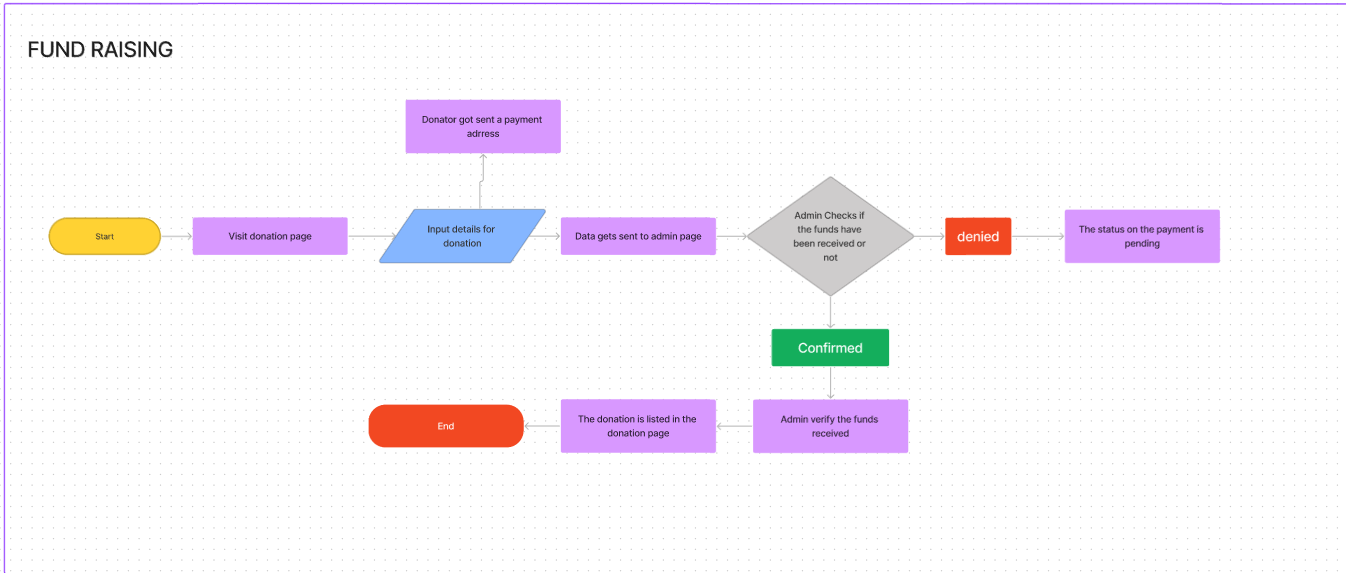
**Member Flowchart**  
 - Ordering food

****

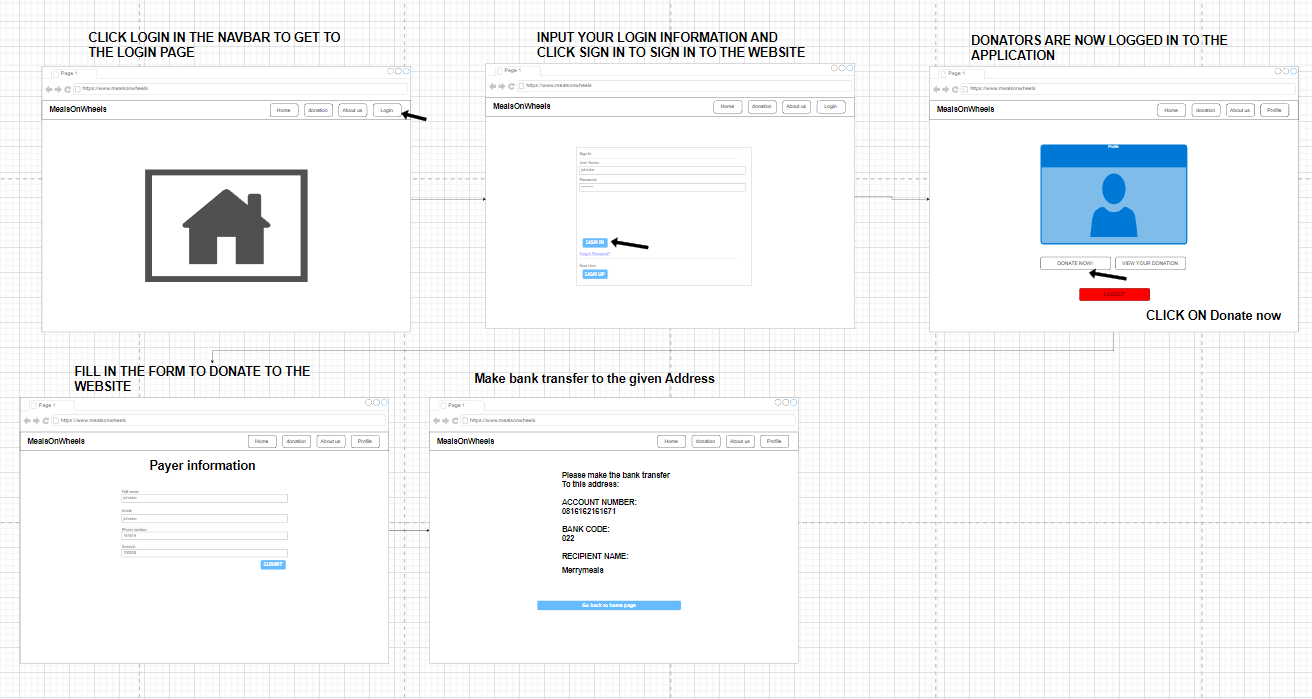
Storyboard

****

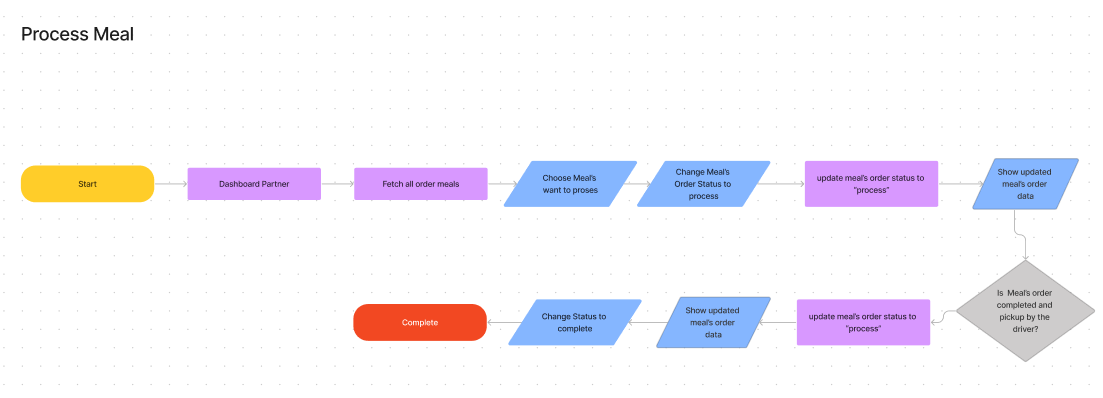
**Donor flowchart**

****

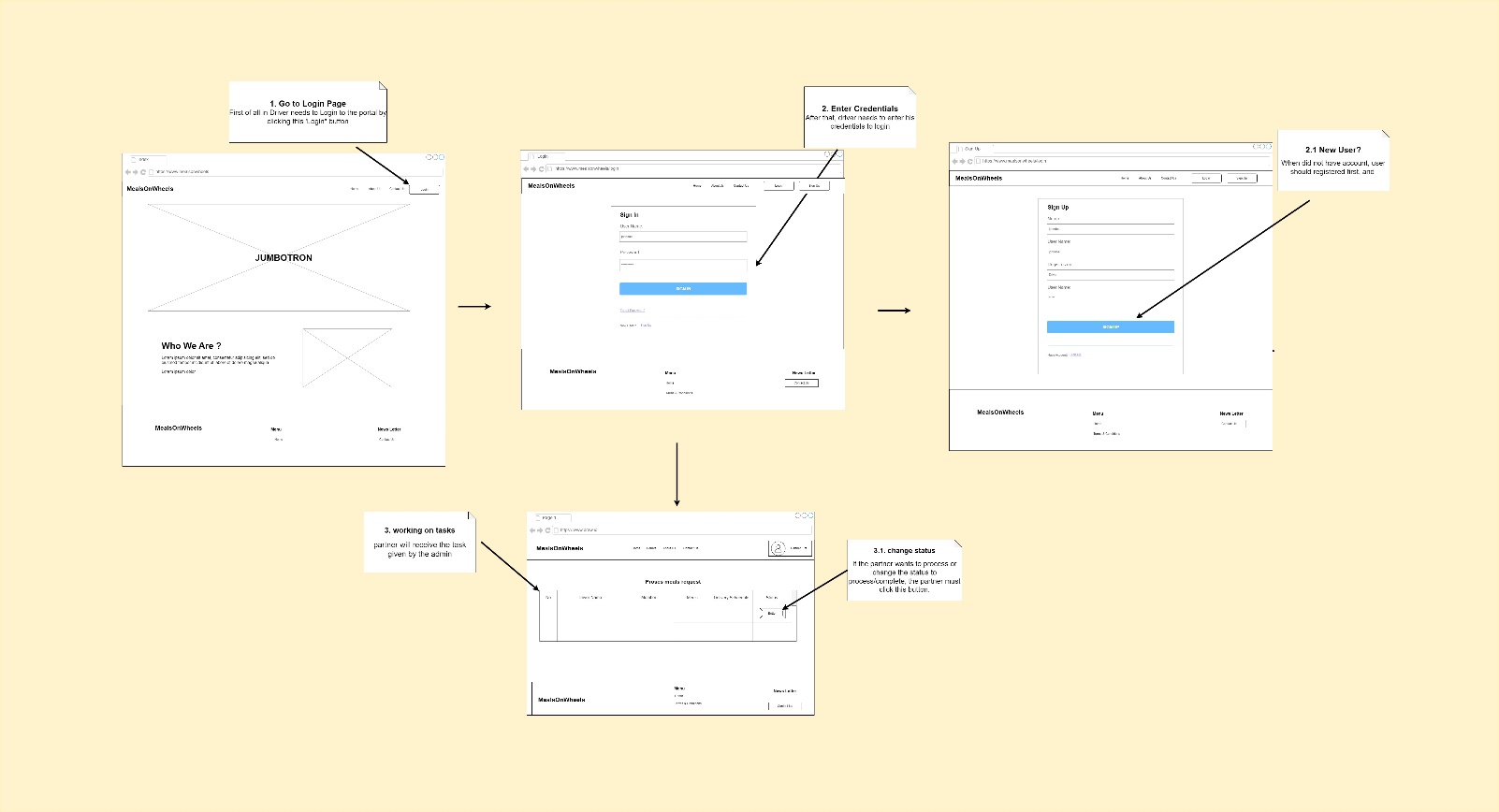
Storyboard

****

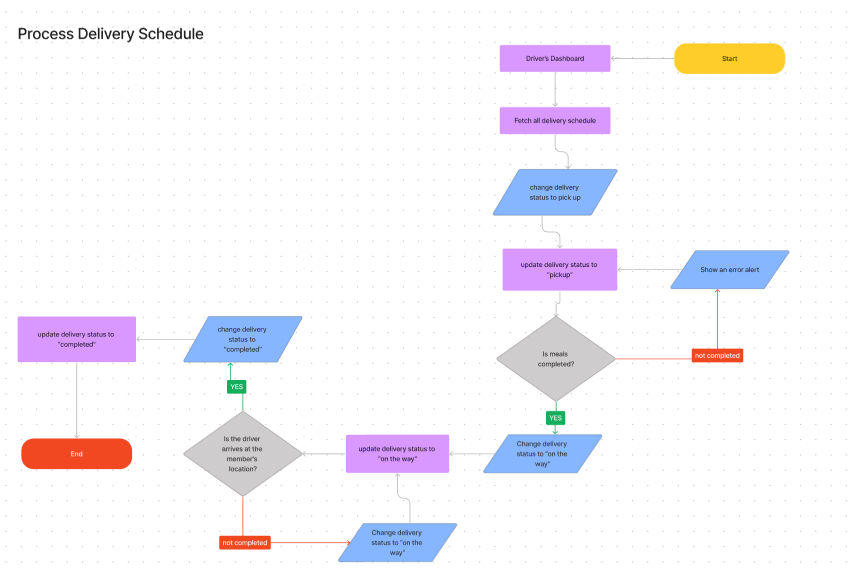
**Partner flowchart**

****

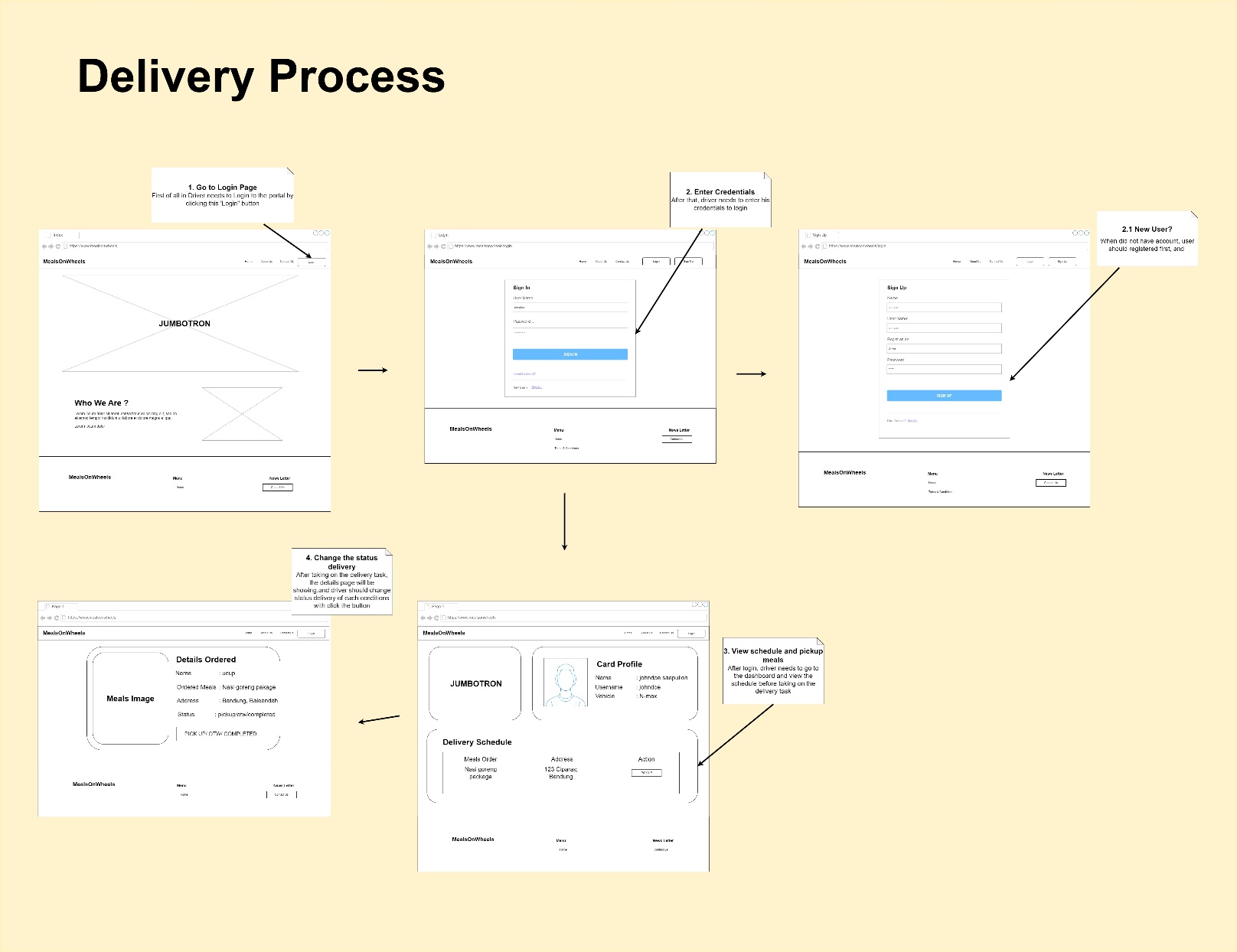
**Storyboard**



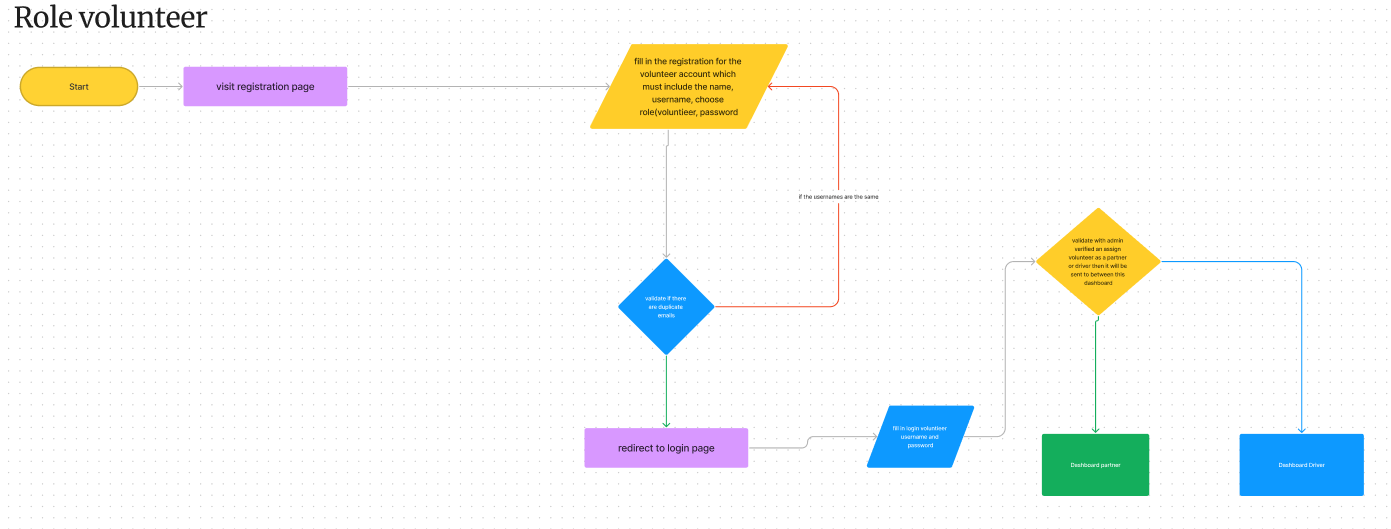
**Driver flowchart**

****

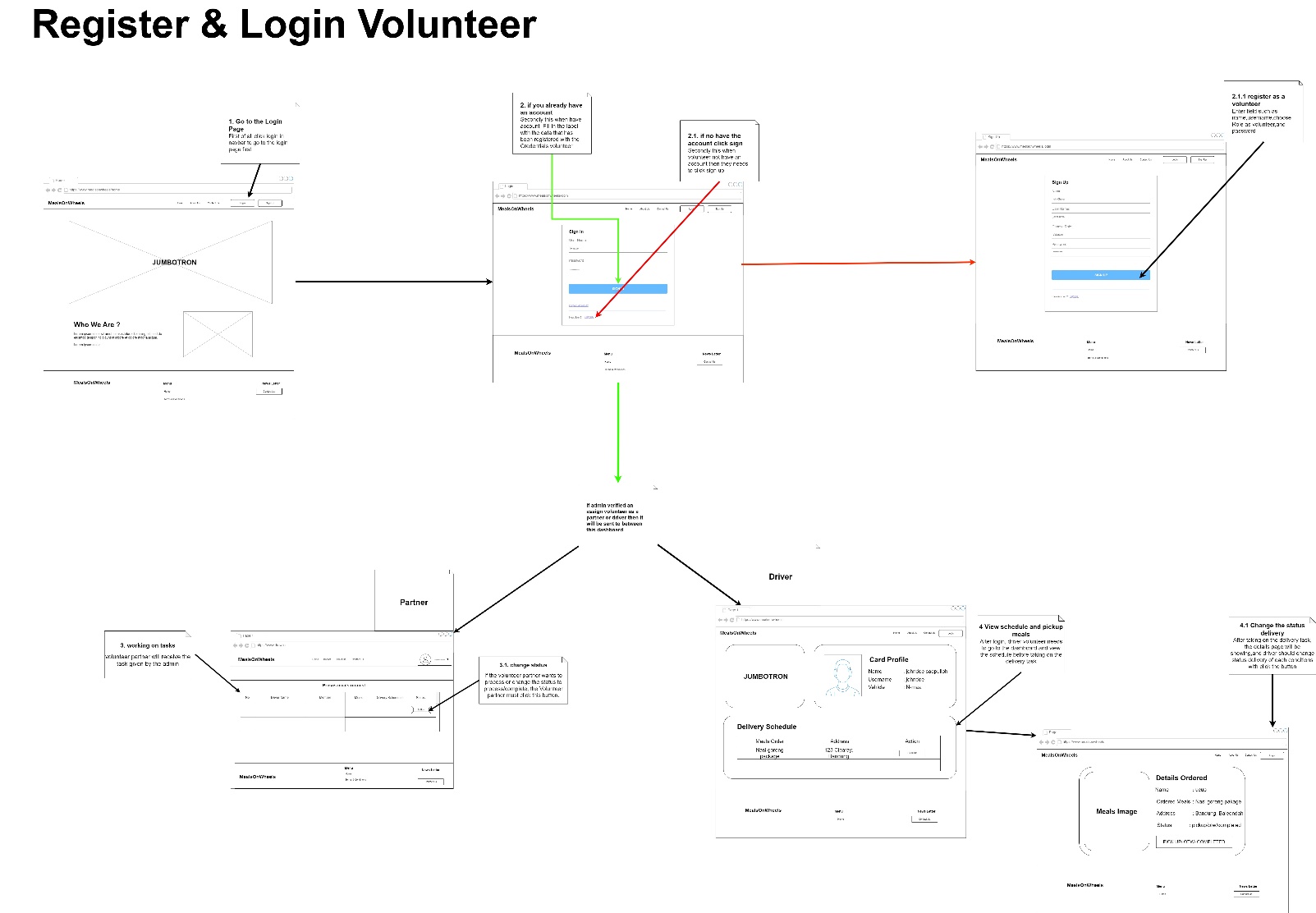
**Storyboard**

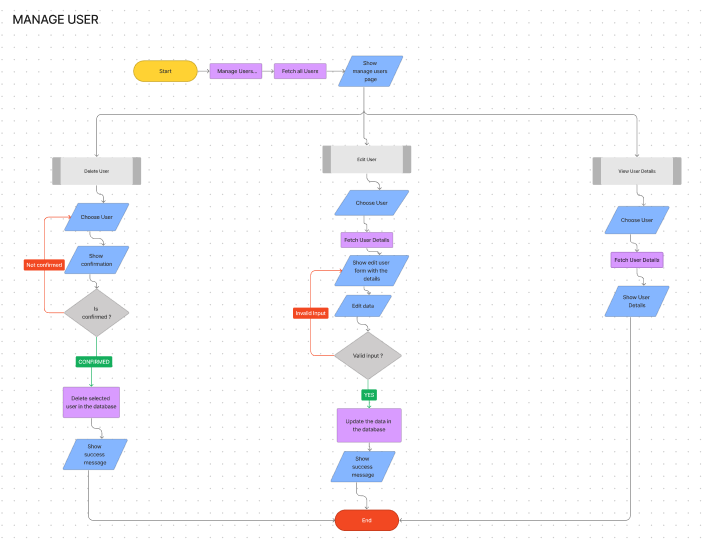


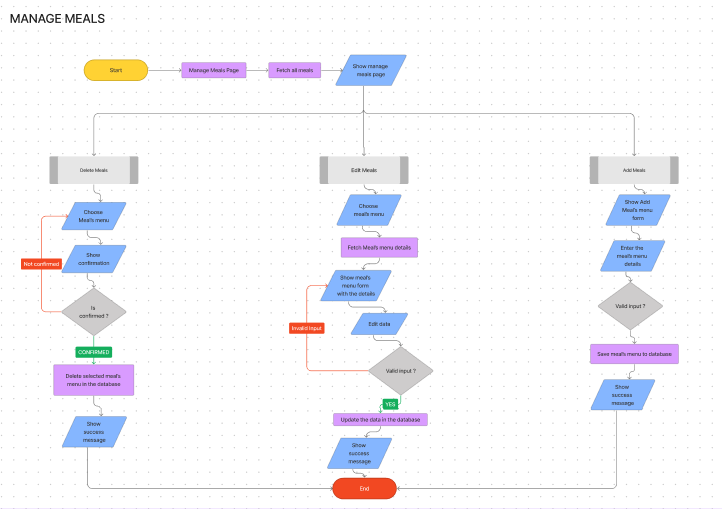
**Volunteer flowchart**

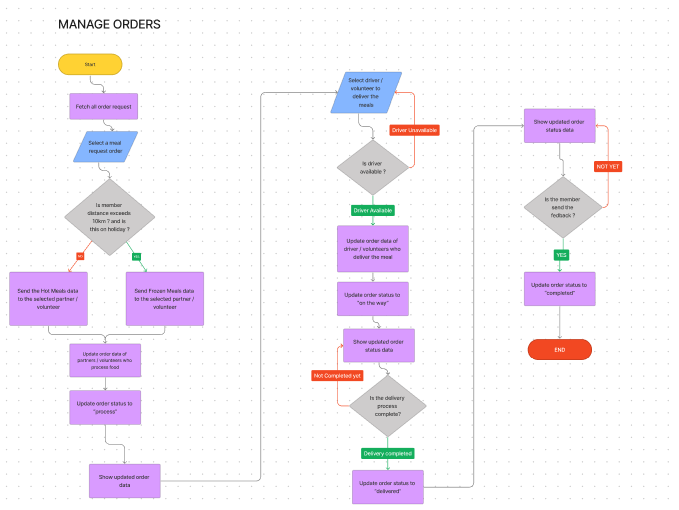
****

**Storyboard**

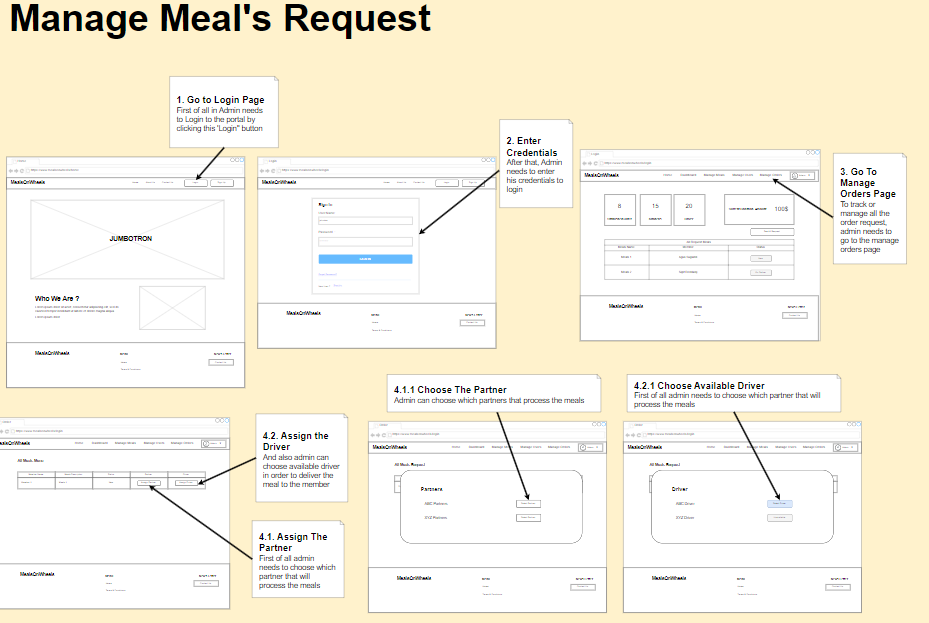


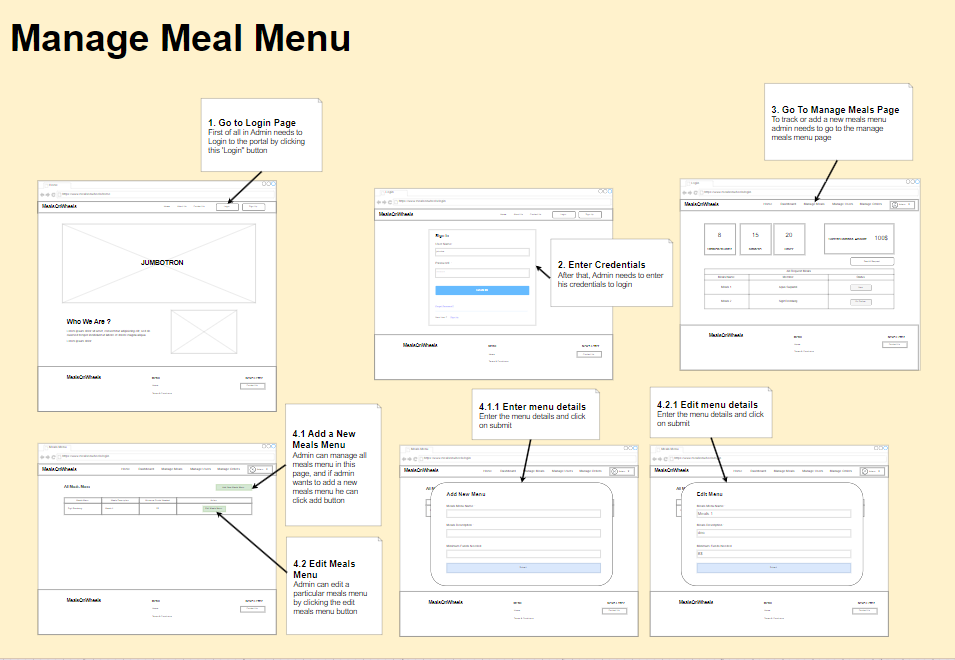
**Admin flowchart  
**

****

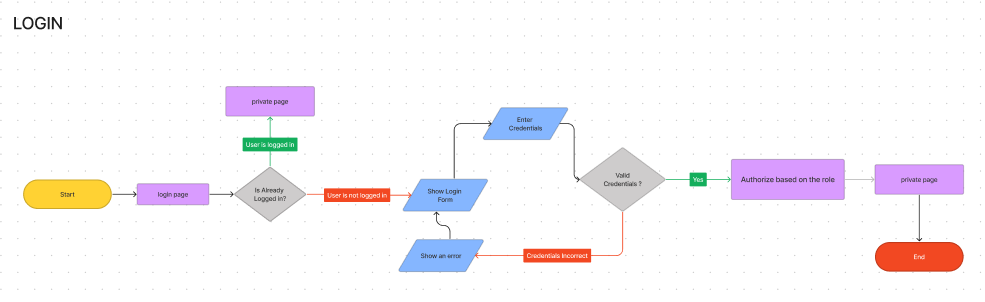
****

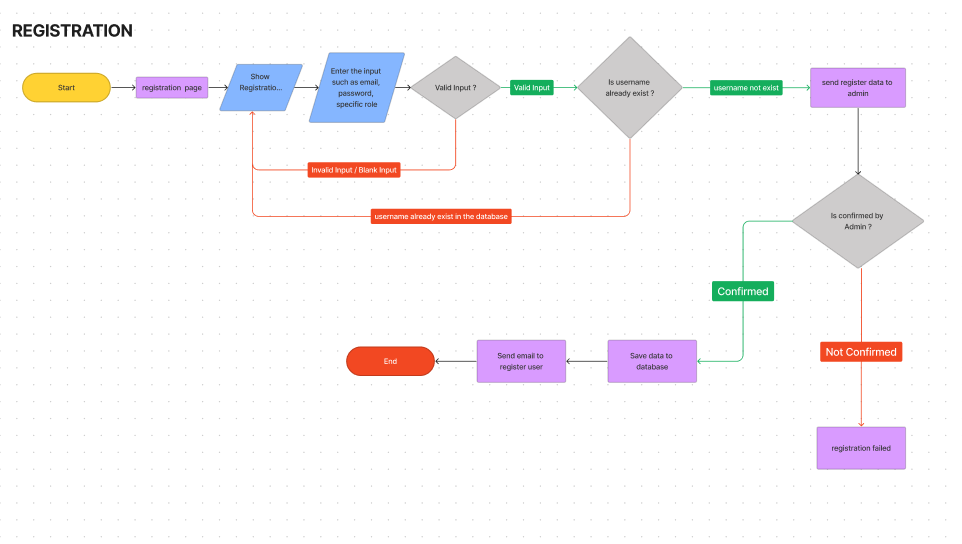
**Storyboard**

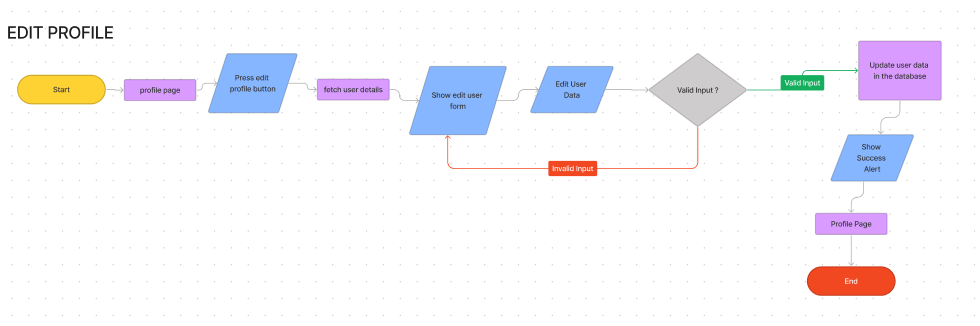
****

****

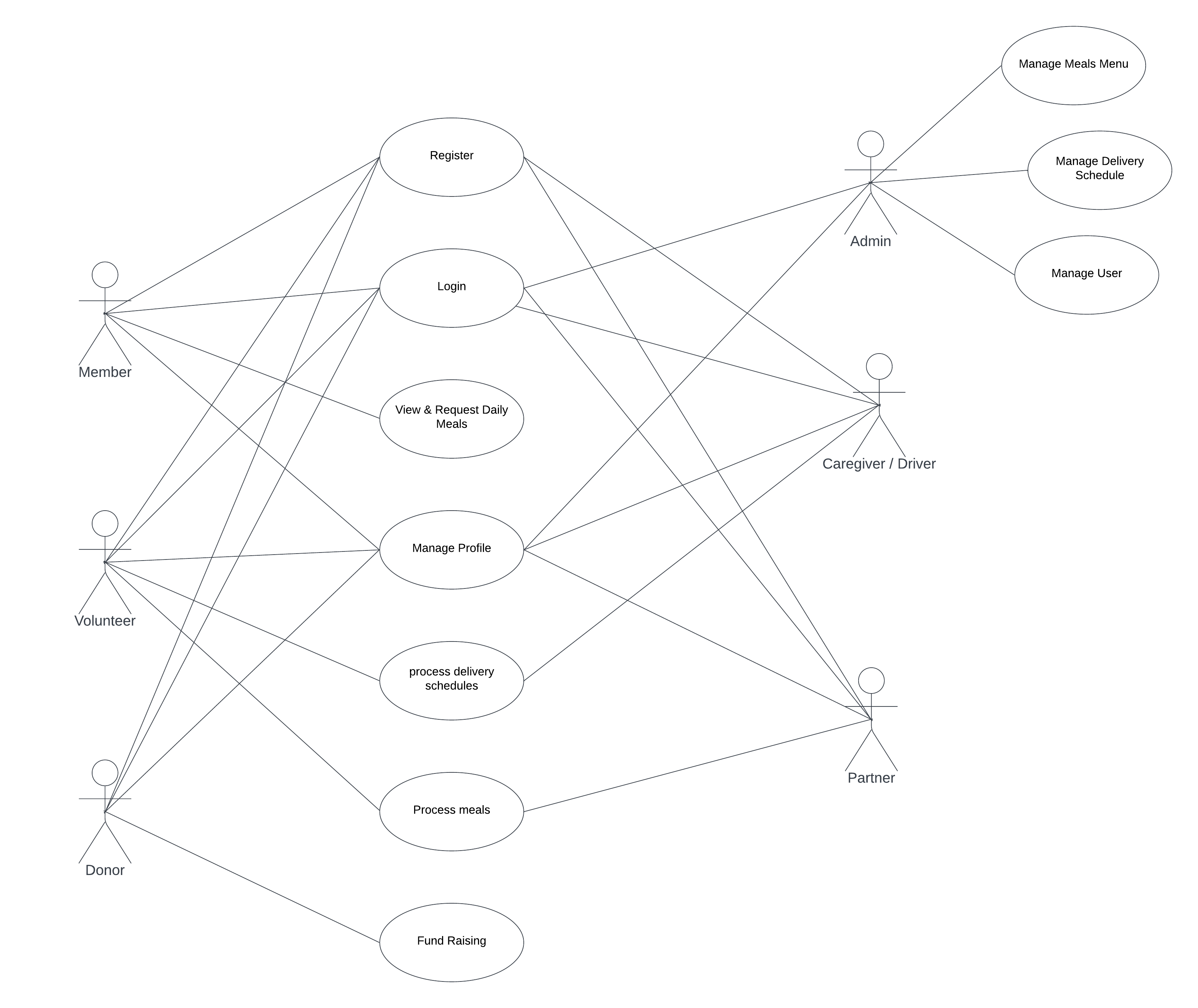
**General function flowchart**

****

****

****

**UML use case**



**7.0 Project Stakeholder presentation**

ppt slides

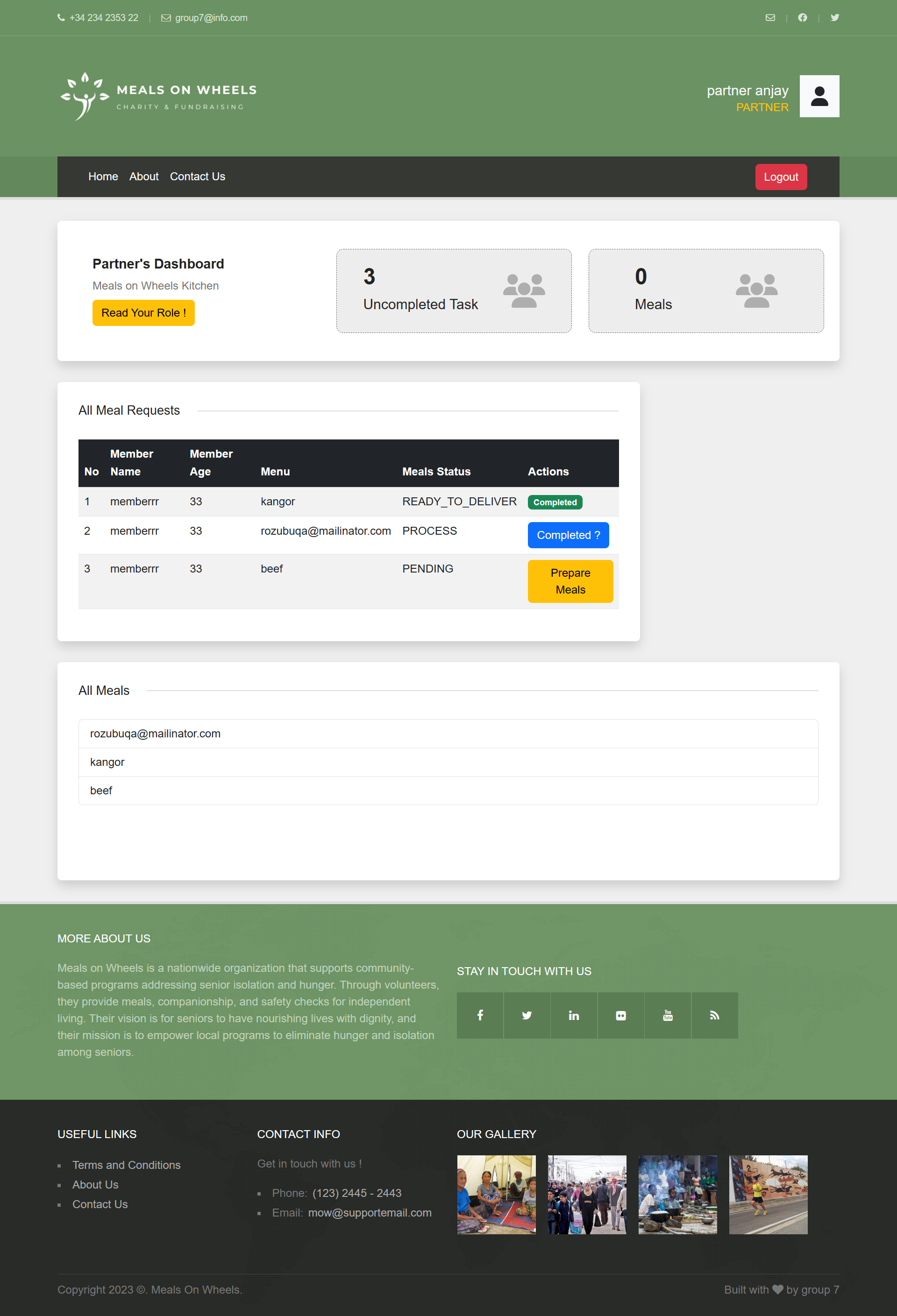
(Stakeholders review statements and document their feedback)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stakeholder | Specific Areas to be improved | Actions to be taken to improve performance | Progress Status (Achieved and not achieved) | Screen capture ID (improvised design page) |
| Sr. Manager from Unity One solution, Merry Meal representatives |  |  | achieved |  |
| Sr. Manager from Unity One solution, Merry Meal representatives |  |  |  |  |

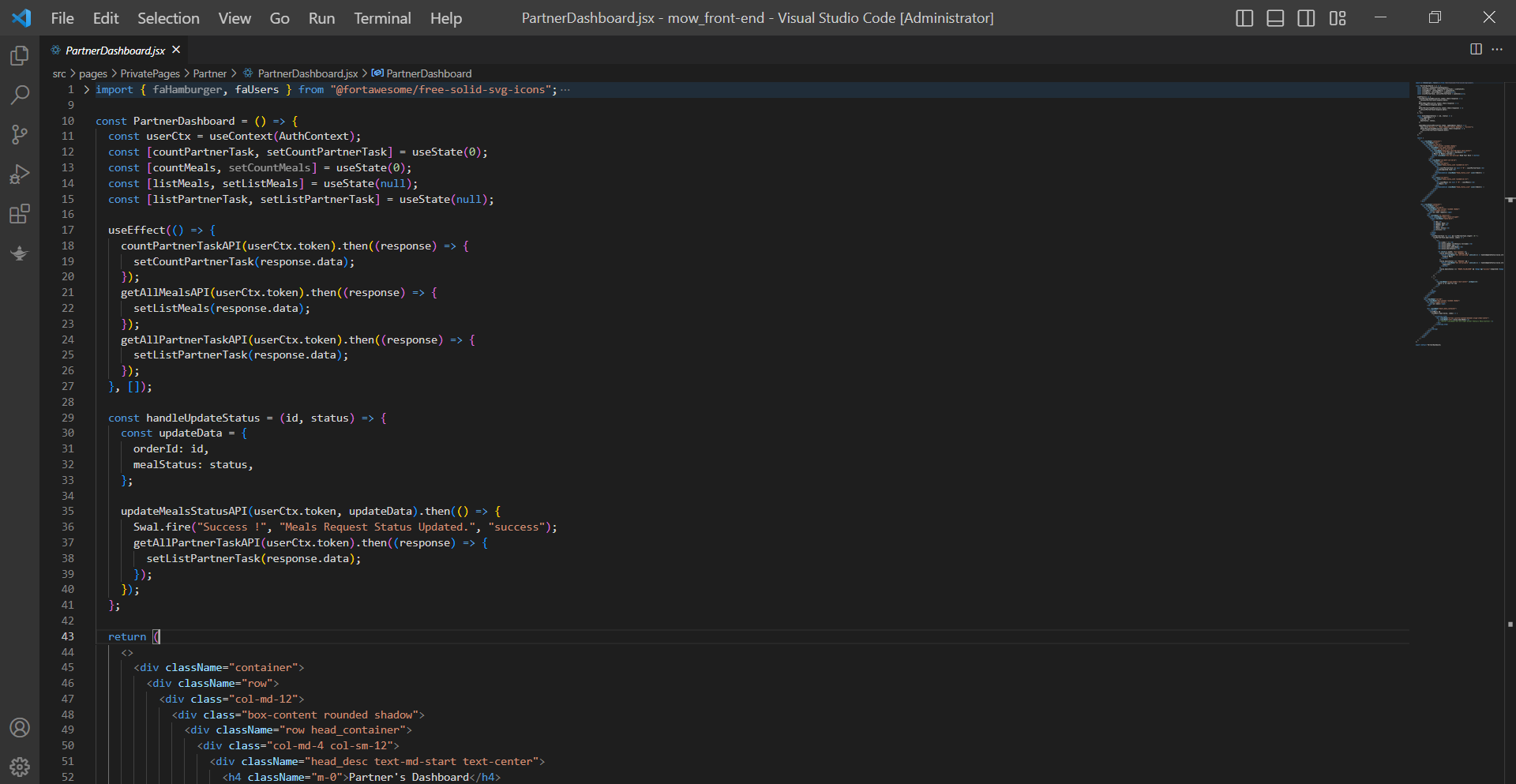
**Observation And Witness Records:**

**8.0 Project execution module-wise: evidence**

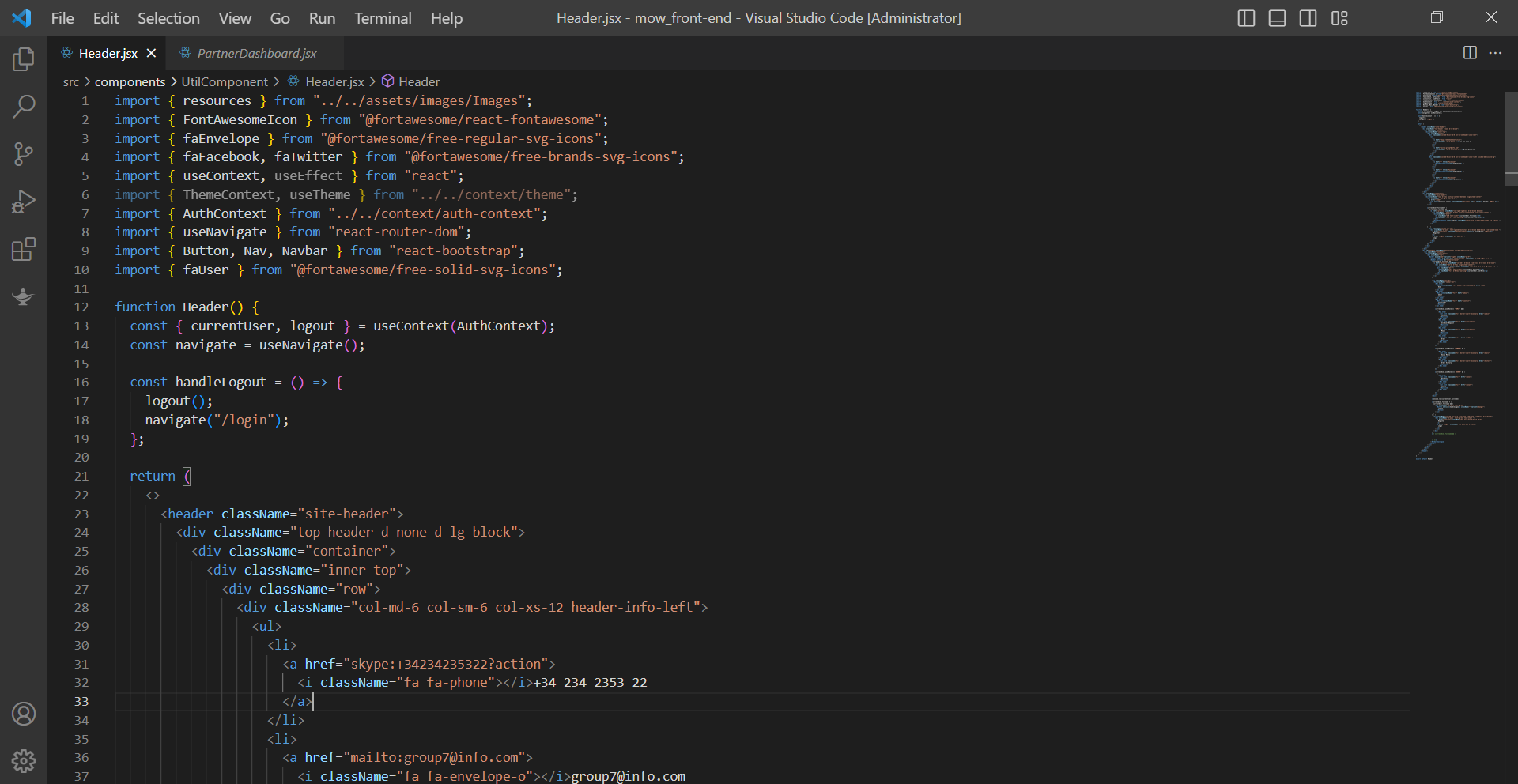
**-Dashboard Partner**

****

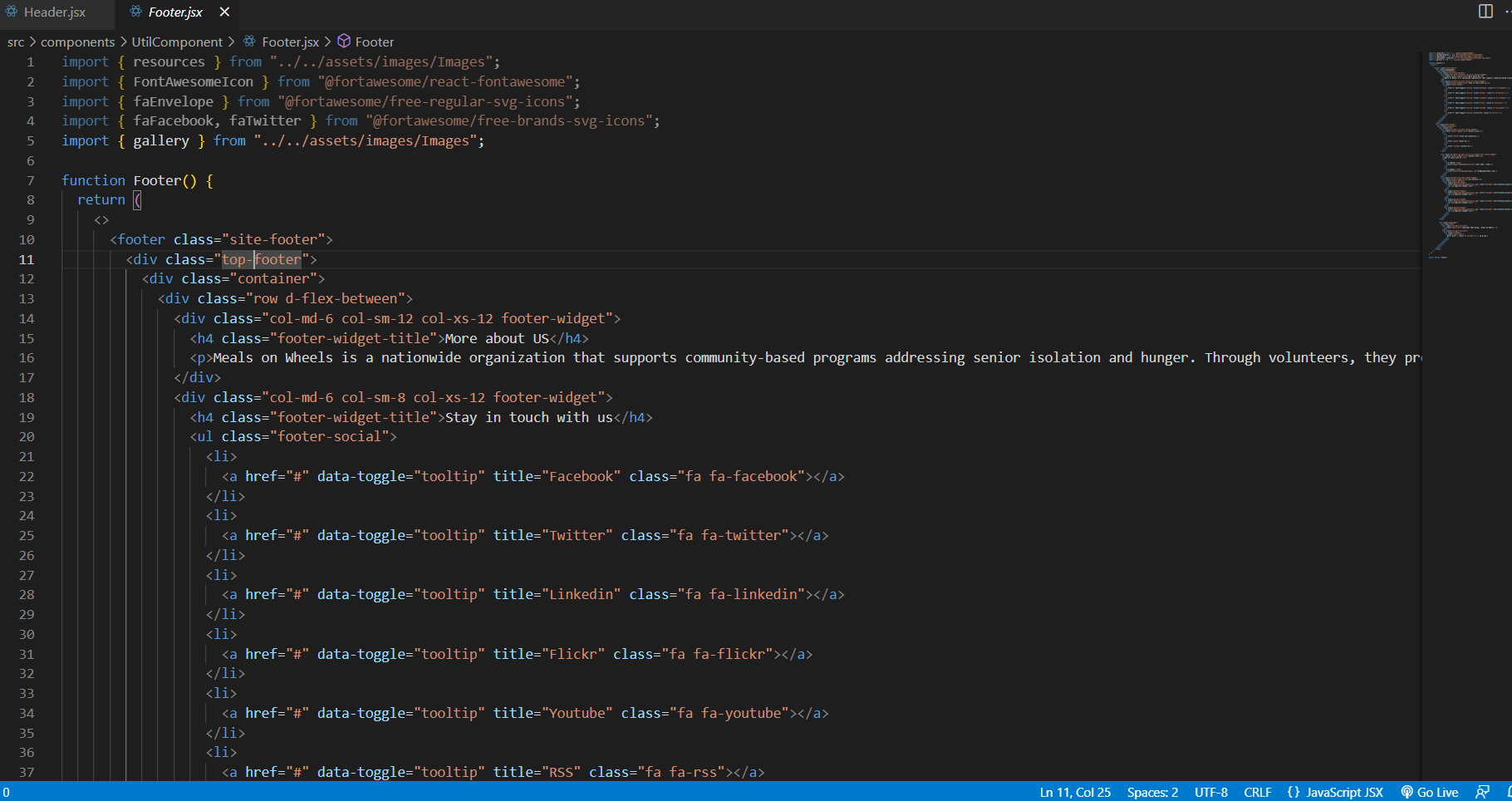
**- Partner Dashboard**



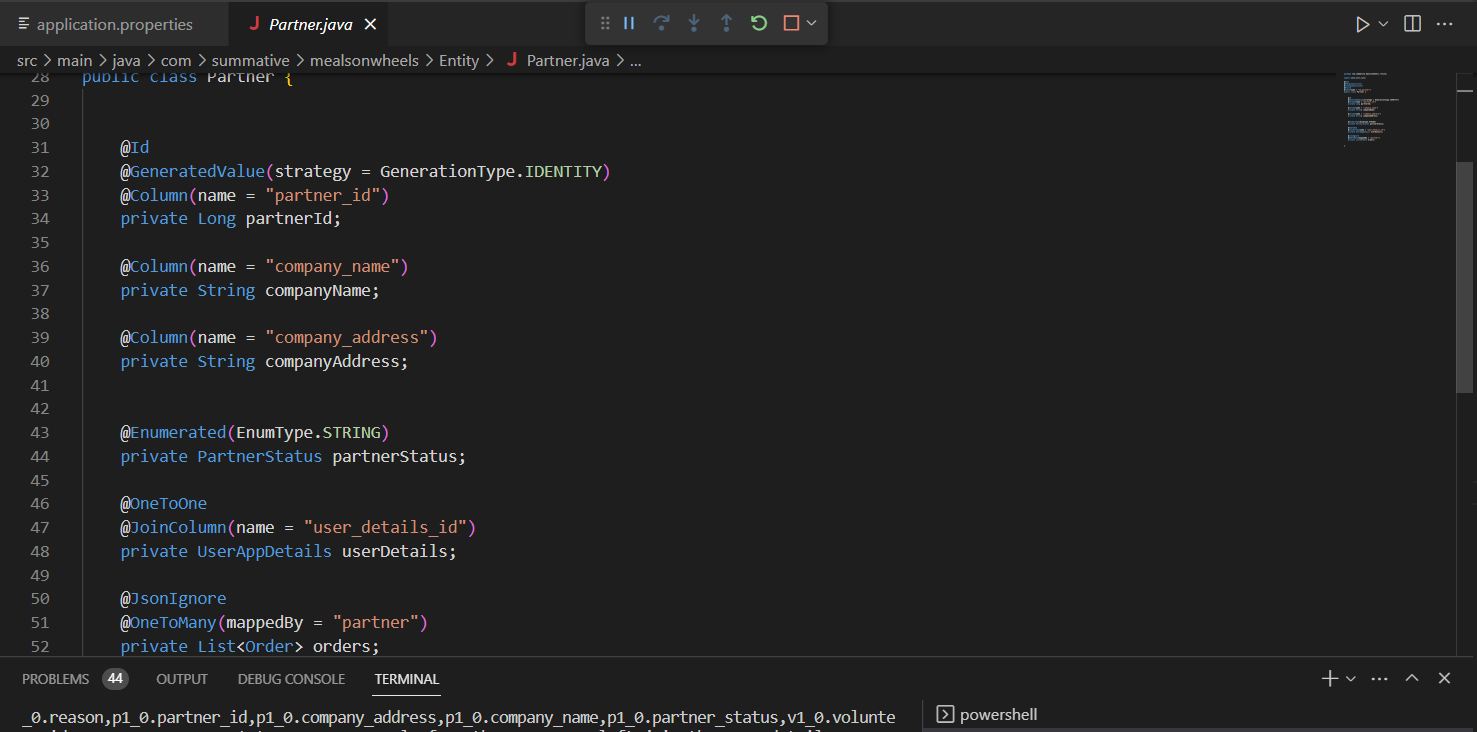
**-Navbar**



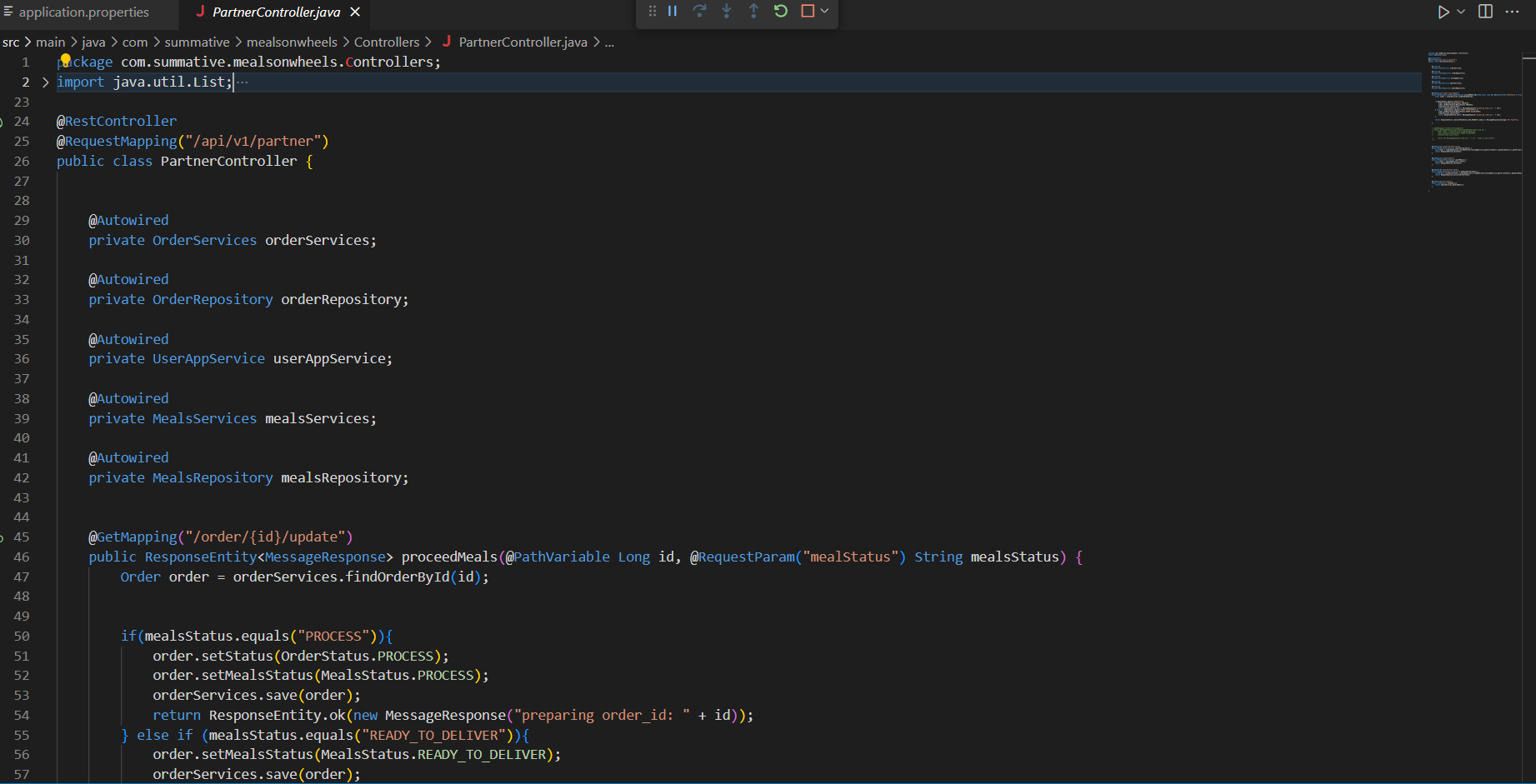
**-Footer**



**-Partner backend**



**-partner controller**



**9.0 Unit Testing :**

(min 5 to 7 unit testing for each module)

1. Integrations Testing

|  |  |
| --- | --- |
| **Test Scenario** | Ensure account authorization with JWT Token |
| **TS001** |
| **Test Cases** | Integration Testing |
| **IT001** | Check if the user can access the website if there is no bearer token. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | Test Data | Expected Result | Actual Result | Final Result |
| TS001 | IT001 | 5 | * Open Postman API * Login to get Bearer token | Open Postman Api, change method to post and type URL http://localhost:8080/auth/login  {      "email": "admin2@gmail2534ssaas.com",      "password": "admin2"  } | The user must have their bearer token authorization to be able to access the website. | After we login, we can get Bearer Token in Postman | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **IT001** | A screenshot of a computer  Description automatically generated |

1. Unit Testing:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| TS001 | Validating Login as a partner functionality | Email : partner@gmail.com  Password: password | partner must successfully log in to the portal | Successfully logged in | Pass |
| TS002 | Validating Registration as a partner functionality | fullname : Partner MOw  Email : partner@gmail.com  Password: password  Company Name : ABC PARTNER | partner must successfully Create account | Successfully Registration | Pass |
| TS003 | check the functionality of the food process | mealStatus : "PROSES" | status should be changed to successful | function Successfully | Pass |
| TS004 | check the functionality of the Meals ready to deliver | mealStatus : “READY\_TO\_DELIVER” | status should be changed to successful | function Successfully | Pass |
| TS005 | check the functionality of the Partner task | Meals Name : batagor | must appear a cooking task with the food name batagor | As expected | Pass |

|  |  |
| --- | --- |
| Test case ID | Test Scenario |
| TS001 | Validating Login as a partner functionality |
|  |  |

|  |  |
| --- | --- |
| Test case ID | Test Scenario |
| TS002 | Validating Registration as a partner functionality |
|  |  |

|  |  |
| --- | --- |
| Test case ID | Test Scenario |
| TS003 | check the functionality of the food process |
|  |  |

|  |  |
| --- | --- |
| Test case ID | Test Scenario |
| TS004 | check the functionality of the Meals ready to deliver |
|  |  |

|  |  |
| --- | --- |
| Test case ID | Test Scenario |
| TS005 | check the functionality of the Partner task |
|  |  |

3 : User Acceptance Testing (UAT)

|  |  |
| --- | --- |
| **Test Scenario** | "Comprehensive Testing of Driver Functionality Suite" |
| **TS003** |
| **Test Cases** | User Acceptance Testing (UAT) |
| **UAT001** | To allow partner to login with their credentials |
| **UAT002** | To Allow partner doing registration process |
| **UAT003** | To allow partner to processing the order |
| **UAT004** | To check the task uncompleted status, is there any task to working or not |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | Expected Result | Actual Result | Final Result |
| TS003 | UAT001 | 5 | * Open the browser * Go to Meals on Wheels Application * Registration and go to login | After login, partner will reach to the dashboard | As expected | Pass |
| UAT002 | 5 | * Open the browser * Go to Meals on Wheels Application * Try to Registration | After registration, the partner can login but should waiting the confirmation from the admin | As expected | Pass |
| UAT003 | 3 | * Open the browser * Go to Meals on Wheels Application * Logged In | partner can processing the orders | As expected | Pass |
| UAT004 | 4 | * Open the browser * Go to Meals on Wheels Application * Logged In | The status uncompleted task will show in dashboard | As expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **UAT001** |  |
| **UAT002** |  |
| **UAT003** |  |
| **UAT004** |  |
|  |

Scenario 4 : Performance Testing

|  |  |
| --- | --- |
| **Test Scenario** | "Optimizing Driver Dashboard Page Loading Speed Using Chrome DevTools" |
| **TS004** |
| **Test Cases** | Performance Testing |
| **PT001** | To utilize the page’s loading speed of the patner dashboard page using Chrome DevTools |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | | Expected Result | Actual Result | Final Result |
|  | PT001 | 5 | * Open the browser * Go to Meals on Wheels Application * Registration and go to login | After login, member will reach to the dashboard | | As expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **PT001** | **A screenshot of a computer  Description automatically generated** |

Scenario 5 : Compatibility – Cross Browser Testing

|  |  |
| --- | --- |
| **Test Scenario** | Cross-Browser Verification of Consistent Display and Functionality |
| **TS005** |
| **Test Cases** | Compatibility – Cross Browser Testing |
| **CBT001** | To check a consistent display and function of the login page across different browser |
| **CBT002** | To check a consistent display and function of the login page across different browser |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | | Expected Result | Actual Result | Final Result |
| TS005 | CBT001 | 5 | Downloaded required browser | After testing, Login page maintain a consistent display and function across different browser | | As expected | Pass |
| CBT002 | 4 | After testing, driver’s dashboard page maintain a consistent display and function across different browser | | As expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **CBT001** |  |

Scenario 6 : Portability Testing

|  |  |
| --- | --- |
| **Test Scenario** | "Ensuring Feature Visibility: Consistent Display Across Various Devices for Drivers" |
| **TS006** |
| **Test Cases** | Portability Testing |
| **PB001** | To ensure all feature in the dashboard page is visible to the partner in different devices |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | | Expected Result | Actual Result | Final Result |
| TS006 | PB001 | 5 | prepare the various devices needed for the test | After testing, all dashboard page is visible to the driver in different devices (laptop, iPad and phones) | | As expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **PB001** |  |

this table can be swiped right and left