

Higher Nationals - Summative Assignment Feedback Form

Student Name/ID	MOHAMMED MAHROOF MOHAMMED AASHIK/E230667		
Unit Title	System Analysis & Design		
Assignment Number	1	Assessor	
Submission Date	17.05.2025	Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	

Assessor Feedback:**Grade:****Assessor Signature:****Date:****Resubmission Feedback:**

*Please note resubmission feedback is focussed only on the resubmitted work

Grade:**Assessor Signature:****Date:****Internal Verifier's Comments:****Signature & Date:**

* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

Important Points:

1. It is strictly prohibited to use textboxes to add texts in the assignments, except for the compulsory information. : Figures, tables of comparison etc. Adding text boxes in the body except for the compulsory information mentioned will result in rejection of your work.
2. Avoid using page borders in your assignment body.
3. Carefully check the hand in date and the instructions given in the assignment. Late submissions will not be accepted.
4. Ensure that you give yourself enough time to complete the assignment by the due date.
5. Excuses of any nature will not be accepted for failure to hand in the work on time.
6. You must take responsibility for managing your own time effectively.
7. If you are unable to hand in your assignment on time and have valid reasons such as illness, you may apply (in writing) for an extension.
8. Failure to achieve at least PASS criteria will result in a REFERRAL grade.
9. Non-submission of work without valid reasons will lead to an automatic RE FERRAL. You will then be asked to complete an alternative assignment.
10. If you use other people's work or ideas in your assignment, reference them properly using HARVARD referencing system to avoid plagiarism. You have to provide both in-text citation and a reference list.
11. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL or at worst you could be expelled from the course.
12. Use word processing application spell check and grammar check function to help editing your assignment.
13. Use **footer function in the word processor to insert Your Name, Subject, Assignment No, and Page Number on each page**. This is useful if individual sheets become detached for any reason.

STUDENT ASSESSMENT SUBMISSION AND DECLARATION

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

Student name: MOHAMMED MAHROOF MOHAMMED AASHIK		Assessor name:
Issue date:	Submission date: 17.05.2025	Submitted on: 17.05.2025
Programme: Pearson BTEC HND in Computing		
Unit: 35 System Analysis and Design		
Assignment number and title: 1 Web based System for “Fresh Grocers”		

Plagiarism

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalized. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

Guidelines for incorporating AI-generated content into assignments:

The use of AI-generated tools to enhance intellectual development is permitted; nevertheless, submitted work must be original. It is not acceptable to pass off AI-generated work as your own.

Student Declaration

Student declaration

I certify that the assignment submission is entirely my own work, and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature: E230667

Date: 20.03.2025

Unit 35: System Analysis & Design

Assignment Brief

Student Name/ID Number	MOHAMMED MAHROOF MOHAMMED AASHIK/E230667
Unit Number and Title	Unit 35: System Analysis & Design
Academic Year	2024/2025
Unit Tutor	
Assignment Title	Web based System for “Fresh Grocers”
Issue Date	
Submission Date	

Submission Format

The submission should be in the form of **an individual written report** written in a concise, formal business style using single spacing and font size 12. You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced. Please provide in-text citations, reference list and bibliography using Harvard referencing system. Please also provide a bibliography using the Harvard referencing system.

The recommended word limit is not less than 5000 words, although you will not be penalised for exceeding the total word limit.

Unit Learning Outcomes

LO1 Evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies.

LO2 Produce a feasibility study for a system for system to be developed to solve a business-related problem.

LO3 Assess systems analysis methodologies to effectively solve business-related problems

LO4 Design the system to meet user and system requirements.

Transferable skills and competencies developed

Computing-related cognitive skills

- **Knowledge of essential computing concepts:** Understanding traditional and agile methodologies, as well as web-based system requirements for a given scenario, helps build a foundation of computing principles and practices.
- **Application of knowledge in system design:** Using diagrams (e.g., data flow, ERD) and prototypes strengthens skills in modeling, understanding, and communicating the system's functionalities.
- **Problem analysis and strategy development:** Evaluating methodologies to determine the best fit for a practical scenario encourages strategic thinking in problem-solving.
- **System criteria evaluation:** Conducting feasibility studies enhances skills in assessing whether a system meets current needs and supports future scalability.
- **Deployment of practices and tools:** Leveraging suitable tools, methodologies, and design practices enables effective and structured system analysis and development.

Computing-related practical skills

- **System evaluation and quality trade-offs:** Comparing agile and traditional approaches enhances the ability to assess system trade-offs like flexibility vs. stability.
- **Project planning and management:** Planning and documenting the system's design helps develop project management skills, ensuring requirements are met within time and budget.
- **Risk assessment:** Recognizing data privacy and security risks associated with online orders enhances skills in risk evaluation and management.
- **Effective tool deployment:** Using visual aids (e.g., flowcharts, prototypes) strengthens the ability to communicate system designs clearly.
- **Complex problem-solving:** Addressing dynamic encourages adaptability in solving complex and evolving issues.

Generic skills for employability

- **Intellectual skills:** Critical thinking and analytical skills are strengthened through evaluating methodologies and designing the system.

- **Self-management:** Skills in goal setting, action planning, and adaptability are developed by adhering to project timelines and adjusting to requirements.
- **Contextual awareness:** Understanding client's needs and adapting the system demands improves awareness of how computing solutions impact businesses and communities.

Vocational scenario

Case Study

"Fresh Grocers" is a grocery delivery service in Sri Lanka, operating for over fifteen years, primarily catering to customers in city and suburban regions. With the increased usage of mobile phones and internet services, "Fresh Grocers" decided to implement a web-based platform to make grocery ordering more accessible for customers. This move is expected to enhance customer convenience and provide a competitive advantage in the growing online grocery market.

The key features required for the "Fresh Grocers" web-based system to manage grocery orders and deliveries are as follows:

- a. Customers and delivery personnel can register on the online system. Once registered, they receive an email with their unique username and password.
- b. Customers can place grocery orders through the website. After an order is confirmed, customers receive an SMS with the delivery agent's contact information and estimated delivery time.
- c. Customers can rate delivery personnel based on their experience with each delivery.
- d. The company employs a customer service representative who can manually enter grocery orders on behalf of customers. This is similar to a customer placing an order but without a logged-in customer in the system.
- e. When the customer service representative places an order, they enter the customer's phone number, and the customer receives an SMS confirmation.
- f. The system must assign the delivery agent closest to the customer's location. When a customer or the customer service representative places an order, the system displays a list of available delivery agents nearby, allowing the customer to choose the preferred agent.
- g. Delivery agents manually update their location within the system or can provide it to the customer service representative for input. Similarly, when placing an order, the customer specifies their location directly or through the customer service representative.

Assignment activity and guidance

Assume you are the newly appointed System Analyst for “Fresh Grocers” and are assigned to work on this project. Produce a professional report addressing the following tasks.

Activity 01

Compare and evaluate traditional and agile system analysis methodologies including,

- Discuss and critically evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies.
- Compare the strengths and weaknesses of the two methodologies by referring to the proposed web system for “Fresh Grocers”.

Activity 02

Conduct a feasibility study for the web-based system for “Fresh Grocers” and produce a ‘Feasibility Report’. Further evaluate the importance of the feasibility criteria used to investigate the feasibility of the proposed system.

Activity 03

Review the user and system requirements for the web-based system suggested in the scenario

- Select a suitable software development methodology and asses the effectiveness of the methodology selected.
- Assess how the system and user requirements are met.
- Justify the chosen methodology for the web-based system with relevant examples.

Activity 04

Design a system specification for the web-based system to meet end-user system requirements and assess the effectiveness of your design and the methodology used with reference to how it meets the user requirements.

Your system design specification should include,

- Data Flow Diagrams (Context Diagram/Level 0, Level 1, Level 2 DFDs) and Use Case Diagram
- Entity Relationship Diagram (ERD)
- Flow Charts (For major system functions)
- Prototypes to illustrate system interfaces and functions
- Release plan and tools

Recommended Resources

Please note that the resources listed are examples for you to use as a starting point in your report – the list is not definitive.

<https://hnglobal.highternationals.com/login>

<https://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages/>

<https://asana.com/resources/feasibility-study>

<https://sites.radford.edu/~softeng06/SoftwareRequirements.doc>

Learning Outcomes and Assessment Criteria

Pass	Merit	Distinction
	<p>LO1 Evaluate the strengths and weaknesses of the traditional and agile systems analysis methodologies.</p>	<p>LO1 & LO2</p> <p>D1 Critically evaluate the strengths and weaknesses of the traditional and agile systems methodologies and feasibility study in solving a specific business-related problem.</p>
<p>P1 Discuss the strengths and weaknesses of the traditional and agile systems analysis methodologies.</p>	<p>M1 Compare and contrast the strengths and weaknesses of the traditional and agile systems analysis methodologies with reference to a specific business-related problem.</p>	
	<p>LO2 Produce a feasibility study for a system to be developed to solve a business-related problem.</p>	
<p>P2 Produce a feasibility study for a system for a business related problem.</p>	<p>M2 Evaluate the importance of the feasibility criteria in the systems investigation for the business related problem.</p>	

Pass	Merit	Distinction
	LO3 Assess systems analysis methodologies to effectively solve business-related problems	LO3 & 4 D2 Justify the choice of the methodology used in the context of the business problem.
P3 Review a system using a suitable methodology for a business-related problem.	M3 Asses the effectiveness of the methodology used in providing a solution for a given business context.	
LO4 Design a system to meet user and system Requirements		
P4 Design a fully functional system to meet user and system requirements for the business-related problem.	M4 Assess the effectiveness of the system design, with reference to the methodology used and how the design meets user and system requirements.	

Acknowledgement

I am deeply grateful for the assistance and guidance I received from numerous esteemed individuals, which was instrumental in the successful completion of my task. I would like to express my sincere appreciation to ESOFT for providing a conducive workspace that facilitated the completion of my task. I am delighted to announce the successful completion of the assignment. I am particularly indebted to **Ms. Kausalya** for her invaluable guidance throughout my third semester assignments. Lastly, I extend my heartfelt gratitude to my family members and classmates whose unwavering support greatly contributed to the timely completion of this project. Thank you all for your immense contribution!

Content

Table of Contents

Software	14
System	17
Computer system	17
System Analysis	17
System Analyst	18
Software Development Methodologies	24
Agile Methodology	40
Types of Agile Methodology	43
Agile VS Traditional methodology	53
Activity 02	60
Feasibility Study	60
Feasibility Report	61
Scope of the System	64
Objectives of the System	65
List of Key Stakeholders	66
Technical Feasibility	69
Legal Feasibility	73
Cultural Feasibility	74
Operational Feasibility	75
Alternative Solutions	76
Feasibility criteria types	78
Activity 3 – Requirements	90
Requirements	90
Types of requirements	90
Feasibility techniques used to identify the requirement	93
Activity 4 – Design documentataion	110
Design specification for Fresh Grocers Ordering System	110
Design Specification	110
System Architecture	111

Software architecture	111
Hardware architecture	112
Design Specification	113
Unified Modelling Language (UML).....	115
Design Documentation for Fresh Grocers	120
gantt chart.....	152
References	152

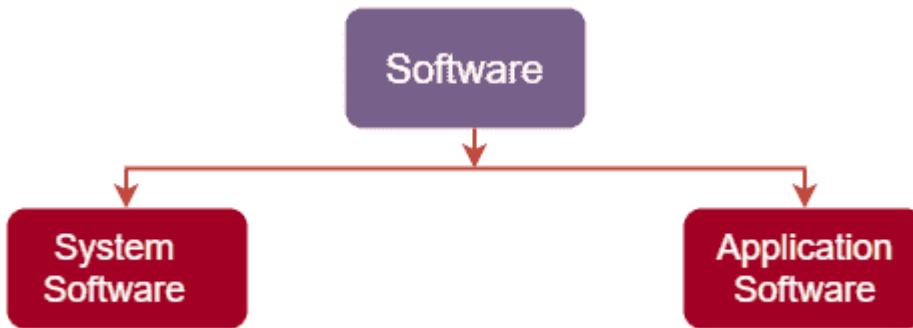
Activity 01

Software

The software is best characterized as a series of instructions, formally known as programs, that conduct operations and specified tasks based on the user's inputs. Every task that a user desires to undertake is governed by software. There is a range of software for various jobs made with binary language (ones and zeroes). (simplilearn, 2025)

Types of Software

Based on its functionalities, the software may be divided into two groups.



1. System Software

System software enables the user to operate computer software or hardware and is in charge of controlling their interaction with one another. It operates in the background all the time to keep the computer's core functions and hardware running. It simply functions as a bridge between the computer and the user, facilitating processes.

There are four types of system software, which are as follows:

☒ Operating Systems

The operating system is a collection of software that aids in program execution and provides general functions to computer applications.

Example: Android, windows 10, Linux and IOS

Device drivers

The actions of hardware devices attached to a computer are handled by device drivers. They serve as a software interface for hardware devices, allowing programs and a computer's operating system to conduct hardware operations without knowing the particular details of the hardware.

Firmware

Firmware is a sort of permanent software that provides low-level control for specific device hardware and is contained in a system's read-only memory.

Utility

Utility software is used to configure, maintain, and assist with computer analysis and optimization.

Example: Antivirus Software, WinZip And Backup Software.

2. Application Software

Application software is a type of end-user program that provides particular functionality to assist users in completing specified activities. Developers continue to create software programs depending on consumer requirements.

- Word Processors
- Multimedia Software
- Web Browsers
- Freeware
- Shareware
- Open-Source
- Database Software

How Software is distributed?

Different types of software are available based on their distribution technique.

Commercial

Commercial software distribution entails giving customers with a license to use such program. Users cannot modify or add features to the program because they do not fully own it and are simply given with a license.

Open-Source

Open-source software differs from commercial software in that it may be downloaded from the internet together with its source code. Users may work on the programs and make modifications to the software since they can get the source code.

Example : VLC Media player, GIMP and Appache web server.

Freeware

Freeware software may be downloaded from the internet and is absolutely free of charge. Adware is typically found in freeware, which implies it has integrated advertising.

Example : Free Studio, Adobe Reader and Skype.

Shareware

Shareware is a type of software that is free to download for a short time on a trial basis. Its goal is to provide customers with a genuine experience so that they will decide to purchase the full version of the program for an indefinite period of time based on their experience with the shareware.

Example : WinZip, Adobe Acrobat 8 Professional and GetRight.

System

A system is a group of items that function together as an interconnected network to accomplish a certain aim. The collection of objects might include hardware, software, personnel and much more. Systems are always around us such as computer systems which have both hardware and software to conduct particular operations. Every system functions within specific restrictions called constraints. These restrictions determine the bounds within which the system may function. (sanyajagy, 2025)

Computer system

A computer system combines hardware and software components that work together to accept input, store data, process information, and output the results. The hardware components include the physical parts of the computer such as the motherboard, CPU, RAM, hard disk, monitor, keyboard, mouse, etc. Software components include programs or sets of instructions that run on the computer, with the operating system being the most important software component. A computer system can perform various tasks such as word processing, internet browsing, gaming, and multimedia editing, among others, by accepting input in various forms, processing the data, and outputting the results in various forms.

System Analysis

System analysis is an evaluation of a technical system, such a software package, for troubleshooting, development or enhancement objectives. Through in-depth examination, analysts might identify mistakes in code, accessibility concerns for end-users or design incompatibilities. (indeed, 2025)

Systems analysis is the process of examining and evaluating computer-based systems for troubleshooting or development purposes. The process involves collecting information about the system, evaluating its performance, and developing recommendations for improvement. It is a critical step in ensuring the effective development and operation of computer-based systems, as it provides a thorough understanding of the system's makeup and design, identifies opportunities for improvement, and implements solutions to enhance performance and value to users.

System Analyst

A Systems Analyst evaluates and enhances an organization's systems and procedures for improved efficiency. They analyze existing processes, gather requirements from stakeholders, and design solutions that align with business objectives. Communication skills are crucial as they bridge technical concepts to non-technical stakeholders. They oversee testing, ensuring systems meet specifications, and provide training and support to users. Continuous monitoring allows them to identify opportunities for improvement and recommend updates to keep systems current. Systems Analysts play a pivotal role in leveraging technology to streamline operations and achieve organizational goals through a blend of technical expertise and analytical prowess.

Software Development Life Cycle (SDLC) Models

A software life cycle model (also known as a process model) is a diagrammatic and graphical depiction of the software life cycle. A life cycle model represents all of the procedures necessary to move a software product through the stages of its life cycle. It also captures the structure in which these procedures will be carried out.

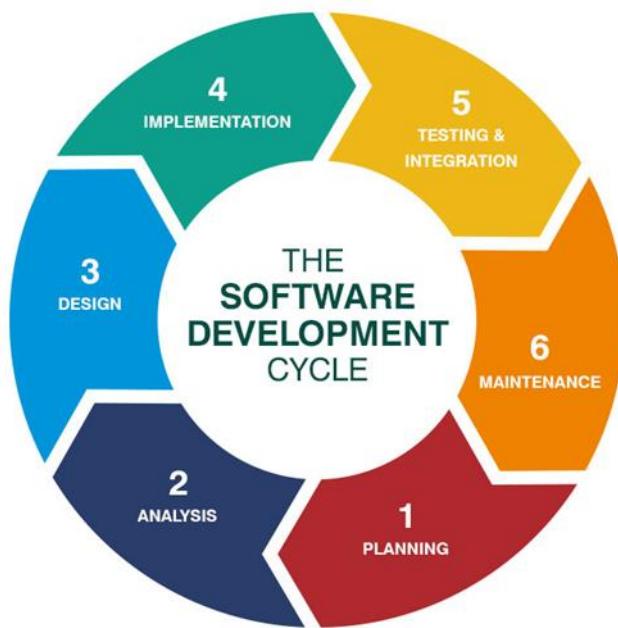
A life cycle model is a diagram that depicts the numerous actions that occur on a software product from its creation through its retirement. Different life cycle models may assign different development tasks to different periods. Thus, regardless of whatever life cycle model is used, the key activities are included in all life cycle models, even though the actions are carried out in various orders in different life cycle models. Multiple activities may be carried out at any point of the life cycle. (Anon., n.d.)

The Software Development Life Cycle (SDLC) provides a comprehensive framework that outlines the entire process of software creation from inception to deployment. It encompasses various stages, including planning, coding, testing, deployment, and maintenance, among others. By defining custom SDLCs, companies establish structured and iterative methodologies tailored to their specific needs and project requirements. These customized frameworks enable teams to follow a predictable sequence of steps, ensuring consistency, efficiency, and quality throughout the development process. Moreover, SDLCs facilitate collaboration among team members, stakeholders, and clients, promoting clear communication and alignment of objectives. The SDLC serves as a guiding roadmap that helps organizations navigate the

complexities of software development, manage resources effectively, and deliver successful outcomes.

The Software Development Life Cycle (SDLC) is a systematic procedure that allows the development of high-quality, low-cost software, in the lowest feasible production period. The purpose of the SDLC is to deliver great software that meets and surpasses all customer expectations and desires. The SDLC creates and describes a thorough plan with stages, or phases, that each cover their own process and deliverables. Adherence to the SDLC boosts development speed and decreases project risks and costs associated with other techniques of production. (synopsys, 2025)

Stages of SDLC



The Systems Development Life Cycle (SDLC) is a framework that defines the phases involved in developing and delivering an information system. It provides a structured approach to planning, designing, developing, testing, and deploying information systems. The SDLC can help organizations to:

- ❖ Develop high-quality, cost-effective information systems
- ❖ Reduce the risk of project failure

- ❖ Improve communication and collaboration among project stakeholders
- ❖ Manage project scope and requirements
- ❖ Deliver information systems on time and within budget

Stages of SDLC

1. Planning and Requirement Analysis

At this point, the software's needs are determined. Discussions are undertaken among the many stakeholders, managers, and users to determine how the program will be utilized. Who is going to utilize it, and how will they use it? During this stage, information on the type of input required and the desired outcome is gathered. Once the information has been gathered, it is analyzed to see whether the requirements can be included into the program that is being created.

2. Defining Requirements

After requirement analysis, the next step is to clearly state and document the product requirements. These are approved by the customer or market analysts and documented in a Software Requirement Specification (SRS) document, which includes all product requirements to be designed and developed during the project.

3. Designing the Product Architecture

The SRS serves as a reference for product architects to develop the best product architecture. Based on the SRS, multiple design approaches for the product architecture are proposed and documented in a Design Document Specification (DDS). The DDS is reviewed by all key stakeholders, and the best design approach is selected based on risk assessment, product robustness, design modularity, budget, and time constraints.

4. Building or Developing the Product

This stage involves the actual development and building of the product. The programming code is generated according to the DDS. If the design is detailed and organized, code generation can

be done smoothly. Developers follow the organization's coding guidelines and use programming tools like compilers, interpreters, debuggers, etc. The programming language used depends on the type of software being developed.

5. Testing the Product

This stage is often a part of all the stages in modern SDLC models. However, this stage specifically refers to the testing phase of the product, where product defects are identified, tracked, fixed, and retested until the product meets the quality standards defined in the SRS.

6. Deployment in the Market and Maintenance

This is the stage at which the completed software is provided to the customer. Once the user starts using the product, the genuine issues become apparent. These issues are handled on a regular basis as they arise. (Anon., n.d.)

Once the product is tested and ready, it is formally released in the appropriate market. Sometimes, the product is released in stages according to the organization's business strategy. The product may first be released in a limited segment and tested in the real business environment (User Acceptance Testing - UAT). Based on the feedback, the product may be released as is or with suggested enhancements in the targeted market segment. After the product's market release, its maintenance is carried out for the existing customer base. (BWC, 2019)

Why do we use SDLC?

Let us now examine how these SDLC methodologies contribute to the success of a software product.

High level of control

It gives you some influence over the development process. This guarantees that the final product is consistent with the client's initial requirements, and that the design and testing process preceding software release is well-managed and stable.

Repeatable

If you develop something using a specific model and a similar project comes along, it will be much easier to use the same process with a higher level of confidence that it will succeed. Planning, design, build, and test phases can be repeated, allowing developers to make gradual improvements at each pass.

Increases efficiency

SDLC techniques are significant because they divide the development process into smaller pieces. This makes evaluating each level considerably easier and provides software developers more freedom to work on each stage concurrently.

Flexibility for teams

Because SDLC techniques break down the whole software development process, teams with varied expertise may be formed to handle each step. Companies will be able to respond to market pressure considerably more quickly.

Fewer delays

When implementing SDLC techniques, most stakeholders and developers realize that they have more alternatives and time. Having alternatives helps them to postpone critical decisions until better data or comprehensive hosting packages become available. This implies that the project may continue without risk of coming to a sudden and unexpected halt.

Advantage and Disadvantage of SDLC

Advantages of SDLC

Structured Approach

SDLC provides a systematic method for software development, enhancing efficiency in planning and task organization for developers. This structure reduces errors and boosts productivity, ensuring the delivery of high-quality software on time.

Risk Management

SDLC excels in identifying and managing risks inherent in the software development process. By detecting potential risks early, developers can proactively address and mitigate them, reducing the overall risk associated with software development.

Consistency

SDLC lays the groundwork for consistency in software development through a standardized framework and methodology. This consistency is key in improving the quality of the software, ensuring that the final product aligns perfectly with client expectations.

Collaboration

SDLC promotes a collaborative environment among team members by providing a common framework and language for communication. This collaborative synergy not only enhances the overall quality of the software but also ensures that the end product accurately meets the client's requirements.

Cost-Effective

SDLC is a cost-effective approach as it identifies potential issues early in the development process through the use of prototyping tools like Figma and others. Early detection of issues allows developers to take proactive measures, significantly reducing overall development costs. This cost-effectiveness makes SDLC a strategic choice in the field of software development.

Disadvantages of SDLC

Time-Consuming

A significant disadvantage is the time required to navigate through SDLC, especially when dealing with complex development processes. This time investment can lead to frustrating delays in software delivery, impacting both developers and clients.

Rigid Framework

SDLC exhibits a degree of rigidity, particularly when faced with changing project requirements during development. This lack of adaptability can result in a final product that fails to meet the client's evolving needs.

High Upfront Cost

Embarking on an SDLC journey requires a substantial upfront investment in terms of time, finances, and resources. This upfront cost can pose a significant challenge for smaller businesses or start-ups that may lack the necessary resources to commit to the demands of SDLC.

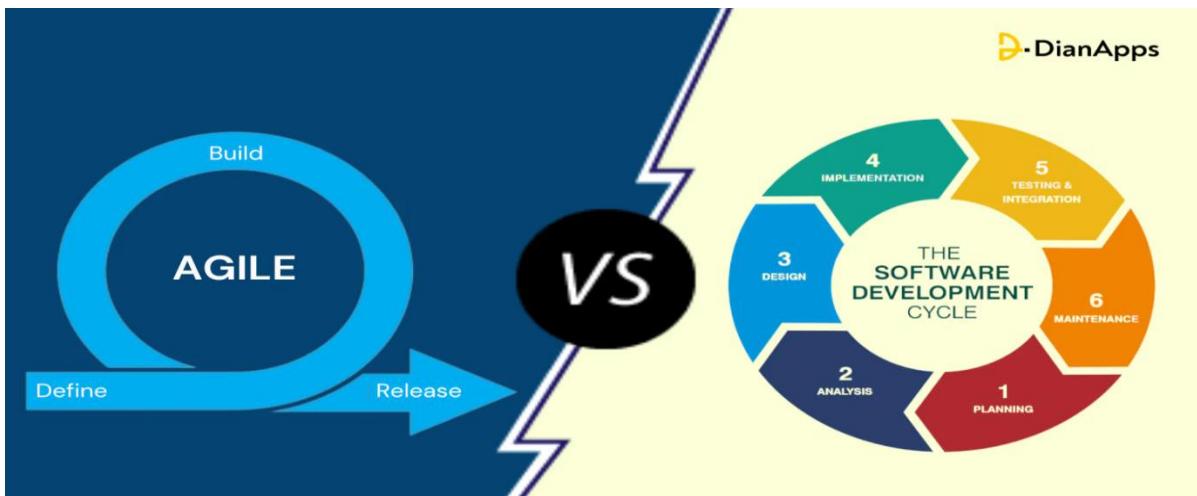
Overemphasis on Process

A potential drawback of SDLC is its tendency to place excessive emphasis on the development process itself, potentially overshadowing the end product. This overemphasis may inadvertently suppress innovation and creativity, resulting in a final product that lacks originality and ingenuity. (Ghosh, n.d.)

Software Development Methodologies

When commencing on a software development project journey, it is highly vital to adopt the correct methodology. Two common approaches that corporations typically examine are SDLC (Software Development Life Cycle) and Agile. While both approaches strive to lead the development process, they vary in their core concepts, procedures, and results. (Quick, 2025)

A software development life cycle (SDLC) model is a conceptual framework that describes all processes in a software development project, from planning through maintenance. This process is linked to various models, each of which includes a number of duties and activities. A set of models streamlines the development process. The development team chooses the appropriate model for the job. (Anon., 2025)



- **Traditional Methodology**

SDLC, or Software Development Life Cycle, is known as traditional linear method to software development. It follows a sequential procedure where each step is completed before going to the next. Each phase has its own objectives and deliverables. (Quick, 2025)

- **Agile**

Agile is an iterative and incremental method to software development. Agile projects are broken down into small sprints, and each sprint focuses on producing a functioning product increment. It emphasizes on cooperation, flexibility, and producing value in quick iterations. (Quick, 2025)

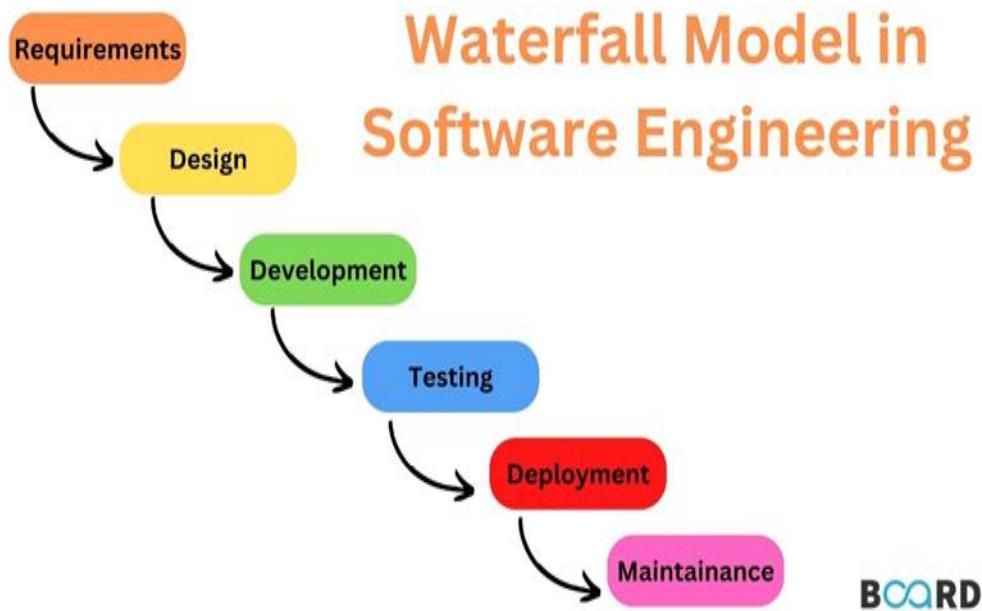
Types of Traditional Methodologies

Traditional methodology is a linear sequential approach to software development that follows a specified set of phases in a sequential manner. Traditional methodologies presume that needs are fully known and that the project can be executed in a single, linear path from beginning to end. Each step of the development process is finished before proceeding to the next, with little or no overlap between phases. The traditional approach is distinguished by a concentration on planning, documentation, and control, and it is frequently employed in big, complicated projects where predictability and control are essential.

Importance of traditional methodology.

1. Structured Approach
2. Well-defined processes
3. Emphasis on documentation
4. Clear project objectives
5. Minimize risks

Waterfall model



The Waterfall Model was the first Process Model to be introduced. It is also known as a linear-sequential life cycle model. It is incredibly simple to grasp and apply. In a waterfall model, each step must be finished before the next phase can begin, and the stages must not overlap. The entire software development process is separated into segments. Typically, the conclusion of one step serves as the input for the following phase in this Waterfall approach.

- **Requirement Gathering and analysis**

During this phase, all potential system needs are identified and recorded in a requirement specification document.

- **System Design**

This phase studies the need specifications from the previous phase and prepares the system design. This system design aids in the specification of hardware and system requirements, as well as the definition of the overall system architecture.

- **Implementation**

The system is first built in discrete programs called units, with input from the system design, and then combined in the following step. Unit testing is the process of developing and testing each unit for functioning

- **Integration and Testing**

Following unit testing, all units generated during the implementation phase are integrated into a system. Following integration, the complete system is tested for flaws and failures.

- **Deployment of system**

Following the completion of functional and non-functional testing, the product is deployed in the client environment or launched to the market.

- **Maintenance**

There are a few difficulties that arise in the client environment. Patches are published to address these vulnerabilities. In order to improve the product, newer versions are published. Maintenance is performed in order to implement these modifications in the client environment.

Waterfall Model Suitability

The Waterfall model is a sequential way of software development which involves requirements, design, implementation, testing, deployment, and maintenance phases that flow down through successive stages. The scale is divided into different phases and each phase has to be achieved before it can move onto the next one, making it structured and understandable. In this respect, it can be applied to projects which have a stable set of needs and requirements, that are unlikely to change dramatically. It is suitable for projects that have a well-defined, steady scope such as developing simple software programs or the projects where there are strict regulations or rules.

Advantages and Disadvantages of the Waterfall Model

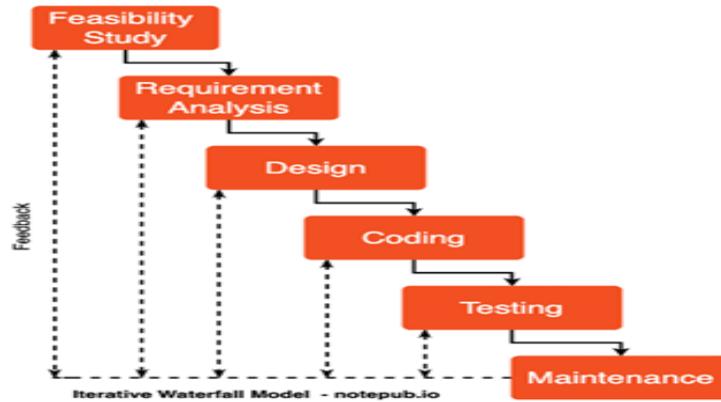
Advantage	Disadvantage
<ul style="list-style-type: none"> Simple and simple to comprehend and apply Because of the model's rigidity, it is simple to manage. Each phase includes its own set of deliverables and a review procedure. One phase at a time is processed and completed. Works effectively for smaller projects with well-defined needs. Stages are well-defined. Milestones that are well understood. Tasks are simple to organize. The process and outcomes are thoroughly documented. 	<ul style="list-style-type: none"> Until late in the life cycle, no functioning software is developed. There is a lot of danger and uncertainty. Not suitable for complex, object-oriented programs. Poor model for long-term projects. Not appropriate for projects with a moderate to high risk of changing needs. As a result, the risk and uncertainty associated with this process model are significant. Measuring development within phases is tough. Cannot meet changing needs. Changing the scope of a project during its life cycle might lead to its demise.

Risk Management

Risk management in the Waterfall model entails identifying possible project hazards early in the process, estimating their effect and likelihood, and implementing ways to minimize or manage these risks. Prior to going on to later phases, risks are often recognized during the initial phase of requirements collection and mitigation techniques are devised. Because the Waterfall methodology is sequential, modifications are difficult to implement once a phase is done, therefore detailed planning and risk analysis are essential to minimizing possible complications down the road. Contingency plans may also be prepared to deal with

unanticipated hazards that appear during the latter stages. Risks are monitored, documented, and shared throughout the project to ensure that the project continues on track and possible concerns are appropriately addressed.

- **Iterative Waterfall Model**



The Iterative Waterfall model closely resembles the traditional Waterfall model but incorporates feedback loops, making it a more realistic approach. It anticipates the likelihood of discovering issues during the testing phase, necessitating a return to either the design or requirement analysis phases for rectification.

The key aim is to identify and address errors within the same phase to minimize the cost of correction. Delaying issue identification leads to increased expenses, which is the fundamental concept behind Phase error containment.

Iterative Waterfall model suitability.

- ⊕ The iterative waterfall model can be appealing for organizations that value a structured and systematic approach to development, testing, and documentation.
- ⊕ If the project's requirements are well-defined and stable.
- ⊕ For smaller to moderately sized projects with limited complexity.

Advantages and Disadvantages of the Iterative Model

Advantage	Disadvantage
<p>Easy to understand, easy to use.</p> <p>Provides a reference for inexperienced staff.</p> <p>Milestones are well understood by the team.</p> <p>It provides requirements stability.</p> <p>Facilitates strong management controls.</p>	<p>All requirements must be known upfront.</p> <p>Deliverables created for each phase are considered frozen.</p> <p>It can give a false impression of progress as there is no output until the project's completion.</p> <p>Integration is one big bang at the end.</p> <p>Little opportunity for the customer to preview the system.</p>

Iterative Waterfall model suitability

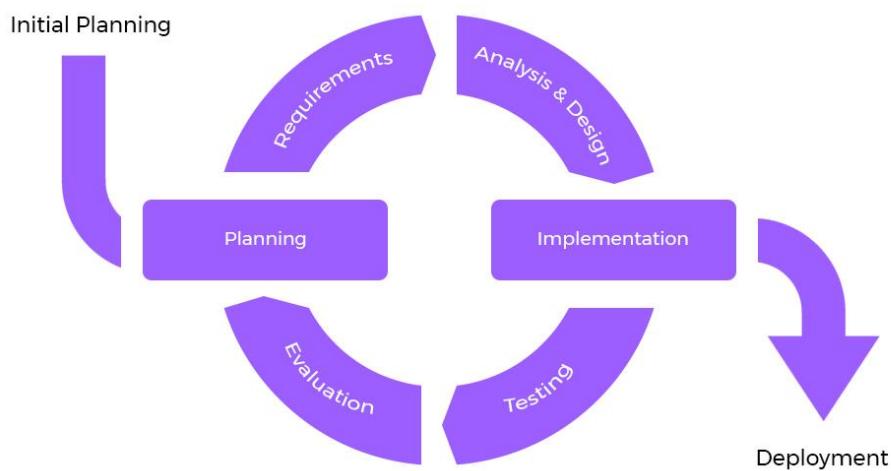
Here we have a model that has the linear structure like Waterfall model and the flexibility of the agile methodology. It divides the project into smaller and less complicated sections, which provides the opportunity for corrections and tweaks at each stage. Its usability is apparent when the project is not strictly confined by requirements and the early feedback from users proves to be beneficial. It is easy to see that this model is practical in situations where the stakeholders prefer a structured way of working but they also want to have the possibility of making changes at any point during the development process.

Risk Management in Iterative Waterfall model

Risk management in the iterative waterfall paradigm entails recognizing possible hazards, analyzing their effect and likelihood, and adopting solutions to handle them during iterative development cycles. Risks are identified at the start of the project and are continuously evaluated and handled during each iteration. The iterative method allows for the modification and tweaking of mitigation techniques depending on input and results from the development

and testing phases of each iteration. This method guarantees that risks are addressed progressively, lowering the possibility of serious difficulties occurring later in the project's life cycle.

Iterative model



An iterative life cycle model does not seek to begin with a complete statement of needs. Instead, development begins by describing and developing only a portion of the program, which is then examined to discover additional needs. This procedure is then repeated, resulting in a new version of the program at the conclusion of each iteration of the model. (Almeida, 2025)

Iterative model design

The development module goes through the requirements, design, implementation, and testing processes with each iteration. Each succeeding module release adds functionality to the prior iteration. The method is repeated until the entire system meets the requirements.

The key to using an iterative software development lifecycle successfully is rigorous requirement validation, as well as verification and testing of each version of the product against

those requirements inside each cycle of the model. Tests must be run and expanded as the software advances through subsequent cycles to validate each version of the software.

Iterative Model Suitability

Waterfall approach emphasizes on single-step development, the Iterative model brings us back into the loop of cycles of development. It is a process by which the design team comes up with a prototype, collects feedback, and they improve and refine what they have made in an iterative manner. This process is repeated until the product you need is developed. It is best used when the projects in question involve uncertainties about what is needed or when the requirements may change over time. In fact, it helps especially for projects that are complex and where stakeholders need to see progress that is visible to them, allowing for necessary modifications from time to time.

Advantages and Disadvantages of the Iterative Model

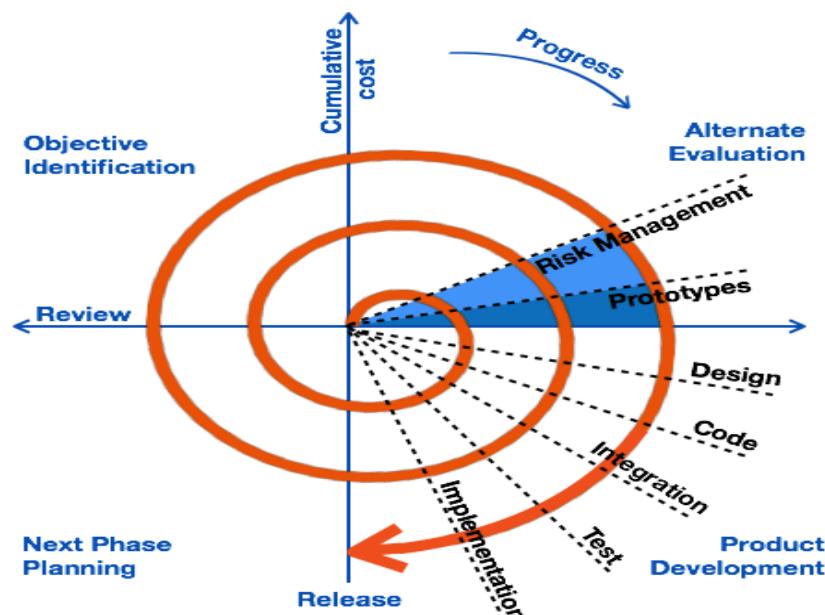
Advantage	Disadvantage
Some functionalities may be created fast and early in the life cycle.	Additional resources may be necessary.
Early and regular results are obtained.	Although the cost of change is lower, it is not well suited to changing requirements.
Parallel development is possible.	More managerial involvement is necessary.
Progress may be quantified.	Because not all requirements are acquired at the start of the complete life cycle, system architecture or design challenges may occur.
It is less expensive to adjust the scope/requirements.	Defining increments may need the specification of the entire system.
It is simple to test and troubleshoot during lesser iterations.	Not appropriate for tiny tasks.
During iteration, risks are recognized and handled, and each iteration is a manageable milestone.	Management complexity is increasing.
Risk management is simplified since the high-risk portion is completed first.	The end of the project may not be known, which poses a danger.
Every increment delivers operational product.	Risk analysis necessitates the use of highly qualified resources.

<p>Each increment's issues, problems, and dangers can be used/applied to the following iteration.</p> <p>Risk analysis is superior.</p> <p>It accommodates shifting requirements.</p> <p>The initial operating time is shorter.</p> <p>Larger and more mission-critical projects are better suited.</p> <p>Software is generated early in the life cycle to permit client review and feedback.</p>	<p>The risk analysis step is critical to project development.</p>
--	---

Risk Management

This methodology promotes iterative development and ongoing feedback. Each iteration incorporates risk management, and the process is adaptable to changing objectives and priorities. Identifying and analyzing risks, taking action to minimize or resolve them, and monitoring and changing the risk management plan as the project advances are all part of the risk management strategy. This technique enables the development team to handle hazards as they arise and alter the development process to mitigate their impact.

Spiral model



The spiral model blends the notion of repeated development with the waterfall model's methodical, regulated characteristics. This spiral model combines the iterative development process model and the sequential linear development model. (tutorialspoint, 2025)

Spiral Model Design

The spiral model is divided into four stages. A software project goes through these phases in iterations known as Spirals.

- **Identification**

This step begins with acquiring the baseline spiral's business needs. This phase is used to identify system requirements, subsystem requirements, and unit requirements in future spirals as the product grows. This phase also includes continual contact between the client and the system analyst to understand the system requirements. The product is launched in the specified market at the conclusion of the spiral.

- **Design**

The Design process begins with conceptual design in the baseline spiral and progresses to architectural design, logical module design, physical product design, and final design in successive spirals.

- **Construct or Build**

At each spiral, the Construct phase relates to the construction of the real software product. In the baseline spiral, while the idea is still being thought about and the design is being developed, a POC (Proof of Concept) is created to get user input. Then, in following spirals with greater clarity on requirements and design specifics, a functioning model of the program known as a build with a version number is generated. These prototypes are submitted to the buyer for review.

- **Evaluation and Risk Analysis**

Identifying, assessing, and monitoring technical feasibility and management risks such as schedule slippage and cost overrun are all part of risk analysis. After testing the build, the customer reviews the program and offers feedback at the conclusion of the first iteration. The spiral iteration process continues throughout the software's life.

Spiral Model Suitability

Spiral Model, a risk-driven approach, integrates the principles of Waterfall and agile methodologies. It defines risk analysis and mitigation as a continuous process through the project life-cycle in which these stages are iteratively running from planning to risk assessment, to engineering, and back to evaluation. The unique feature of this technique is its applicability to complex projects characterized by high uncertainty and risk. It opens the door for adaptability and flexibility, able to deal with fluctuating prerequisites and transforming situations.

Advantages and Disadvantages of Spiral Model

Advantage	Disadvantage
<p>Changing needs can be addressed.</p> <p>Prototypes can be used extensively.</p> <p>More precise requirements can be captured.</p> <p>Users first notice the system.</p> <p>Development may be broken into smaller portions, and dangerous aspects can be created early, allowing for better risk management.</p>	<p>Management is becoming increasingly difficult.</p> <p>The project's conclusion may not be known for some time.</p> <p>tiny or low-risk initiatives are not suited, and tiny projects may be costly.</p> <p>The procedure is complicated.</p> <p>The spiral might carry on indefinitely.</p> <p>A large number of intermediate steps need an abundance of documentation.</p>

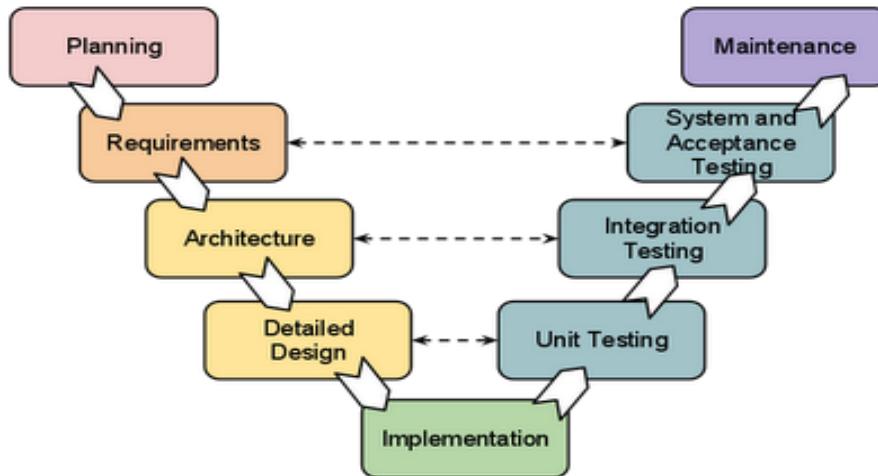
Risk Management

The waterfall and iterative approaches are combined in this model. Each stage of the process incorporates risk management, and the process cycles back on itself to handle risks and uncertainties. Identifying possible risks, analyzing their likelihood and effect, devising and testing mitigation methods, and monitoring those tactics as the project proceeds are all part of the risk management strategy.

V-Shaped Model

Description

It is an extension of the waterfall model. Instead of moving down in a linear way, the process steps are bent upwards after the implementation and coding phase, to form the typical V shape. The major difference between the V-shaped model and the waterfall model is the early test planning in the V-shaped model.



The usage

- Software requirements clearly defined and known
- Software development technologies and tools are well-known

Advantages and Disadvantages

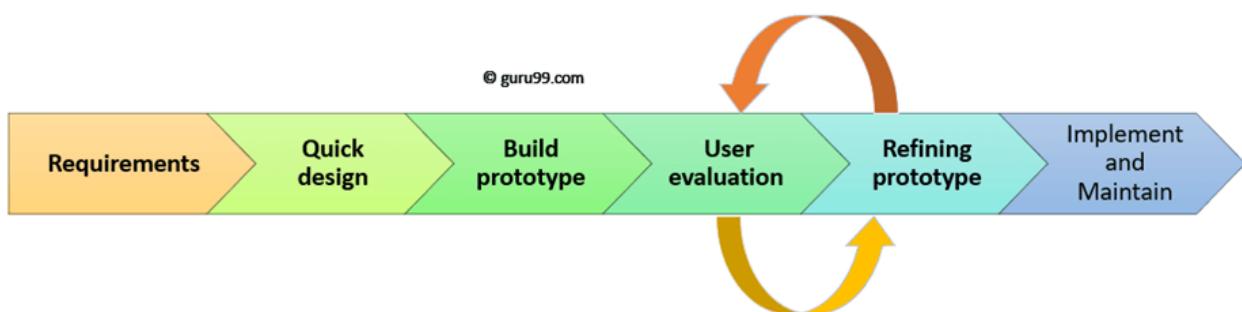
Advantages	Disadvantages
Simple and easy to use Each phase has specific deliverables. Higher chance of success over the waterfall model due to the development of test plans early on during the life cycle.	Very inflexible, like the waterfall model. Adjusting the scope is difficult and expensive. The software is developed during the implementation phase, so no early prototypes of the software are produced. The model doesn't provide a clear path for problems found during the testing phases. Costly and required more time, in addition to a detailed plan

<p>Works well for where requirements are easily understood.</p> <p>Verification and validation of the product in the early stages of product development.</p>	
---	--

Risk Management

Emphasizes verification and validation at each stage to catch risks early. Testing occurs simultaneously with development to detect issues early. Structured approach limits flexibility in addressing unforeseen risks once a phase is complete. During the development of the savings module, testing uncovers a critical error in calculations, allowing early correction before implementation. (Sami, 2012)

Prototype model



A software prototype is a working model with limited functionality. The prototype does not necessarily contain the precise logic used in the final software application and is an additional work that must be factored into the effort calculation. Prototyping is used to allow consumers to review and test developer suggestions before they are implemented. It also aids in understanding user-specific requirements that the developer may not have addressed during product creation. (Martin, 2025)

Stages of Prototyping Model

- **Basic Requirement Identification**

Identification of Fundamental Requirements

Creating the Initial Prototype Reviewing the Prototype Revising and Improving the Prototype. This process entails comprehending the most fundamental product needs, particularly in terms of user interface. At this point, the more detailed features of the internal architecture and exterior factors like as performance and security can be overlooked.

- **Developing the initial Prototype**

In this stage, the first Prototype is created, where the most basic needs are shown and user interfaces are supplied. These functionalities may not function exactly the same way in the final program built. While, in the prototype developed, workarounds are made to provide the same appearance and feel to the consumer.

- **Review of the Prototype**

The prototype is subsequently shown to the client and other key players in the project. The input is collected in a systematic manner and utilized to improve the product under development.

- **Revise and Enhance the Prototype**

The prototype is subsequently shown to the client and other key players in the project. The input is collected in a systematic manner and utilized to improve the product under development.

Advantage and Disadvantage of Prototype Model

Advantage	Disadvantage
Enhanced user participation in the product even before its installation.	Risk of poor requirement analysis due to over-reliance on the prototype.

Users have a better grasp of the system being created since a functioning model of the system is shown. Reduces time and expense by detecting faults much sooner. User feedback is provided more quickly, leading to better solutions. Missing functionality is quickly spotted Functions that are perplexing or difficult to understand can be identified.	Users may become confused between prototypes and live systems. In practice, this process may enhance the system's complexity when the scope of the system expands beyond the original goals. Even if it is technically impossible, developers may attempt to utilize previous prototypes to construct the actual system. If prototypes are not carefully monitored, the work involved may be excessive.
---	--

Prototype model suitability

The Prototype model is about the designing of the software prototype which is done at the beginning of the development. This prototype becomes a physical manifestation of the final product that is shown to stakeholders. Such people see and touch it, which enables them to give feedback. It works best in cases where requirements are either unclear or in cases where user interaction with the product is necessary. It can be considered as an advantage especially in projects that user experience is a focus of attention or when there is a need for many iterations and refinements.

Risk Management

This strategy emphasizes the production of prototypes to test and enhance software concepts before moving on to full development. Building and testing prototypes to discover and resolve problems, improve requirements, and demonstrate technical feasibility helps to control risks. Iterative development and testing, assessing input to identify and reduce risks, and updating the design based on lessons gained are all part of the risk management approach.

Agile Methodology

The Agile methodology is a project management strategy that includes splitting the project into stages and promotes constant collaboration and development. Teams follow a cycle of planning, executing, and assessing. (atlassian, 2025)



To provide high-quality software solutions, agile technique is employed in software development. It places a premium on adaptability, teamwork, and responsiveness to change. The process is broken into discrete parts, called sprints, and the code is continuously integrated and tested. It is critical that team members communicate and collaborate often, and testing is included in the development process. In contrast to the classic Waterfall model, Agile is a flexible and collaborative strategy that promotes continuous improvement and high-quality products.

Important of Agile methodology

Flexibility

High-quality products

Collaboration

Faster time to market

Customer satisfaction

Early defect detection

Agile Manifesto

The Agile Manifesto is a statement that focuses on four ideals and 12 principles for agile software development. It was launched in February 2001 by 17 software engineers who wanted an alternative to the more linear product development method. (Laoyan, 2025)

There are four main values of agile project management:

1. Individuals over procedures and tools

Agile teams encourage team communication and teamwork above working individually and doing things "by the book."

2. Working software above detailed documentation

The software that Agile teams produce should work. Additional labor, like documentation, is not as vital as building effective software.

3. Customer cooperation above contract negotiation:

Customers are incredibly essential inside the Agile technique. Agile teams enable clients to influence where the product should go. Therefore, customer participation is more vital than the finer elements of contract negotiation.

4. Responding to change over following a plan

One of the key advantages of Agile project management is that it enables teams to be flexible. This structure enables for teams to easily modify tactics and procedures without derailing a whole project.

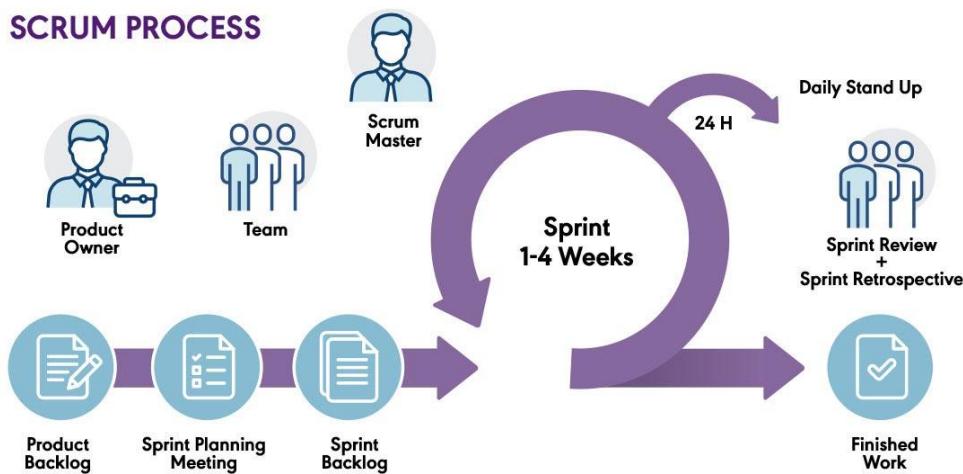
The four values of Agile are the pillars of Agile methodology. From those values, the team developed 12 principles. (Laoyan, 2025)

improvement, equipping teams to identify and resolve problems early, optimize processes and ultimately deliver high quality products fast. With the Agile approach, teams can work closely with stakeholders, and thus, ensure that the end product not only satisfies but also surpasses the customers' expectations. Fundamentally, the Agile model equips organizations with the skills to maneuver the complexity of software development of the 21st century with flexibility, sensitivity, and a never-ending motivation to add value to clients.

Types of Agile Methodology

SCRUM

Scrum is a lightweight agile framework that specifies a collection of ideals, tools, and strategies to help teams organize and manage their work to achieve value. Its formal description can be found in the Scrum Guide (authored by the architects of Scrum), but in a nutshell, it takes an iterative, incremental strategy when delivering products to maximize predictability and manage risk. In reality, this sees a small cross-functional team – the Scrum team – working cooperatively on bite-sized amounts of work at a time. Each work is created within a time-boxed period known as a ‘sprint’, with an emphasis on learning and developing along the way. (pm-partners, 2025)



Advantages and Disadvantages of Scrum

Advantages	Disadvantages
Scrum can help teams complete project deliverables quickly and efficiently	• Scrum often leads to scope creep, due to the lack of a definite end-date
Scrum ensures effective use of time and money	• The chances of project failure are high if individuals aren't very committed or cooperative
Large projects are divided into easily manageable sprints	• Adopting the Scrum framework in large teams is challenging
Developments are coded and tested during the sprint review	• The framework can be successful only with experienced team members
Works well for fast-moving development projects	• Daily meetings sometimes frustrate team members
The team gets clear visibility through scrum meetings	• If any team member leaves in the middle of a project, it can have a huge negative impact on the project
Scrum, being agile, adopts feedback from customers and stakeholders	• Quality is hard to implement until the team goes through an aggressive testing process
Short sprints enable changes based on feedback a lot more easily	

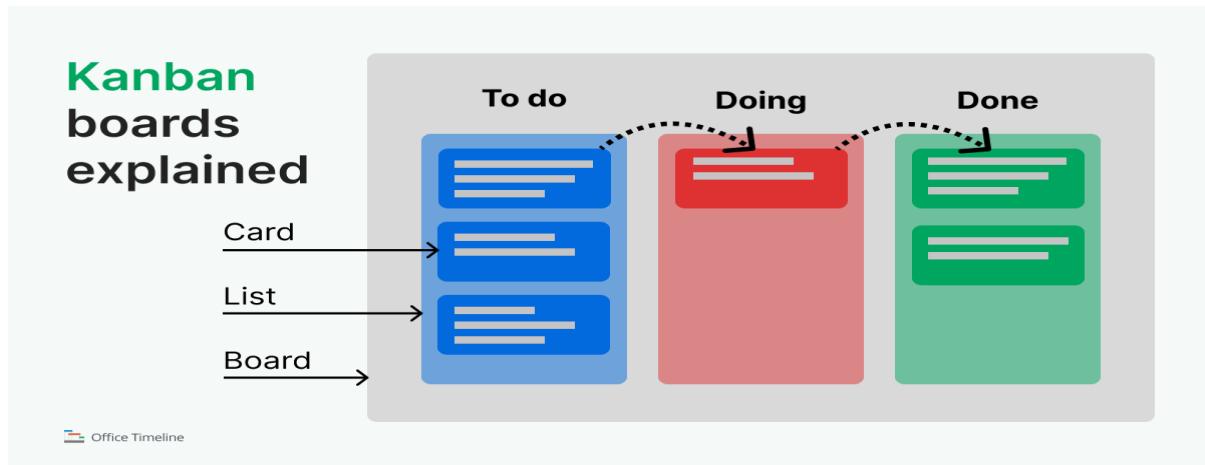
Scrum Master

The Scrum Master is a key role in the Scrum framework, responsible for facilitating the Scrum process and ensuring that the team adheres to Scrum principles and practices. The Scrum Master serves as a servant-leader to the team, helping them understand and implement Scrum practices effectively.

1. Guiding the Team: The Scrum Master coaches the development team on the Scrum framework, Agile principles, and best practices. They help the team understand their roles and responsibilities and guide them in self-organizing to accomplish their goals.
2. Removing Obstacles: The Scrum Master identifies and removes any impediments or obstacles that may hinder the team's progress. This could involve addressing issues related to resources, communication, or organizational constraints.
3. Facilitating Meetings: The Scrum Master facilitates various Scrum events, such as sprint planning, daily stand-ups, sprint reviews, and sprint retrospectives. They ensure that these meetings are productive, focused, and time-boxed.
4. Promoting Collaboration: The Scrum Master fosters collaboration and communication within the team and with stakeholders. They help facilitate transparency and ensure that information flows smoothly between team members and stakeholders.
5. Shielding the Team: The Scrum Master protects the team from external distractions and interruptions, allowing them to focus on their work and meet their commitments. They also shield the team from undue pressure or interference from outside sources.
6. Continuous Improvement: The Scrum Master encourages a culture of continuous improvement within the team, promoting reflection and adaptation. They facilitate discussions during sprint retrospectives to identify areas for improvement and help the team implement changes to enhance their processes.

Kanban

The name Kanban may literally be translated from the Japanese language as ‘visual sign’, ‘signboard’ or ‘card’. In recent years, Kanban boards have become a popular tool to visualize the flow of work. In its simplest arrangement, a Kanban board indicates ‘To Do’ (the work to be done), ‘In Progress’ (what we are working on today) and ‘Done’ (finished). (pm-partners, 2025)



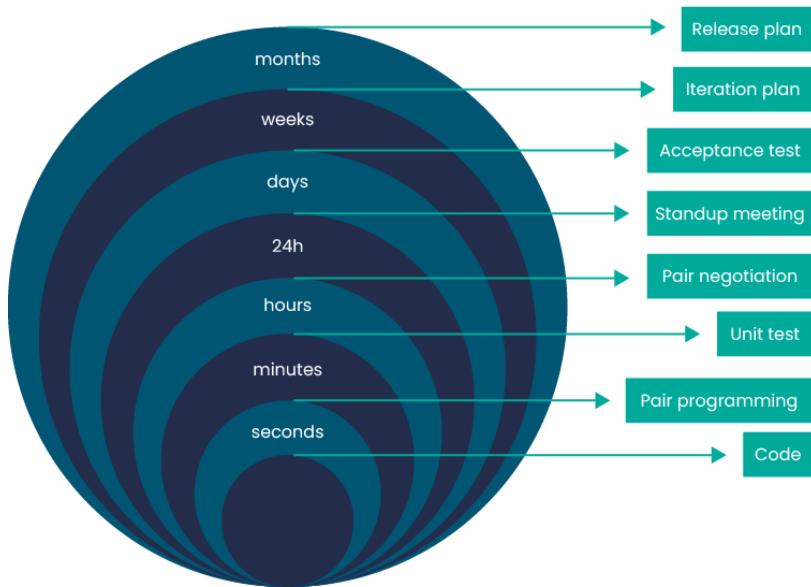
Advantage and Disadvantage of Kanban

Advantages	Disadvantages
<p>Kanban boards provide clear visibility of project tasks and details, enabling team members to understand their responsibilities and track progress efficiently.</p> <p>Easily visualize team progress and task statuses, reducing the need for constant status updates and ensuring alignment among team members.</p> <p>Visualizing workflow, setting priorities, and identifying bottlenecks enhances team effectiveness and directs energies to the right tasks at the right time.</p> <p>High task visibility reduces concerns about missing important tasks, allowing team members to focus on their work and preventing burnout.</p>	<p>Lack of detailed time frames and schedule information can make it challenging to predict task completion and overall project timelines, particularly for estimating.</p> <p>Kanban boards may become overwhelming for larger projects, requiring additional organization methods such as swimlanes to manage complexity effectively.</p> <p>Maintaining up-to-date boards is crucial for the effectiveness of the Kanban methodology, requiring discipline from team members and utilizing project management software for ease of updating.</p>

Extreme Programming

Extreme Programming (XP) is an agile software development approach that seeks to deliver better quality software and improved quality of life for the development team. XP is the most detailed of the agile frameworks addressing suitable engineering techniques for software development. (agilealliance, 2020)

XP Feedback Loops



Advantage and Disadvantage of Extreme Programming

Advantages	Disadvantages
<p>Readability and Updateability: XP promotes straightforward documentation, making it easy to read and update in the future if required.</p> <p>XP replaces lengthy requirement files with user stories, simplifying the development process.</p> <p>Customer involvement in development and testing ensures that business assumptions</p>	<p>Customers may not have a clear idea of the final product, making it difficult to estimate effort, cost, and time accurately.</p> <p>Insufficient documentation may lead to unclear requirements and project scope extensions.</p> <p>Pair programming requires more time and may not always yield the intended outcomes, sometimes due to personality clashes.</p>

and criteria are met, leading to a product that fulfills expectations.	Structural Adjustments for Adoption: Adopting XP requires significant structural adjustments from conventional software development approaches.
Short iterations minimize the need for overtime by focusing on specific needs and delivering what the client truly wants.	XP's reliance on face-to-face meetings limits its effectiveness in distant dispersed teams.
Pair programming improves software quality and fosters collaboration among developers, enhancing understanding of assumptions and needs.	Regular client meetings may consume a lot of time, potentially affecting project implementation.
XP promotes continuous communication, fostering openness and accountability within the team and keeping everyone updated on project progress and changes.	Clients may lack the time, desire, or expertise to assist in product development, particularly under tight deadlines or when oversight is inadequate
The iterative approach of XP allows for the rapid delivery of usable components, focusing only on essential functionality.	

(nearshore-it, 2025)

Lean Software Development

Lean Software Development (LSD) is an agile approach centered on minimizing development time and resources, avoiding waste, and finally providing just what the product requires. The Lean technique is also sometimes referred to as the Minimum Viable Product (MVP) strategy, in which a team exposes a bare-minimum version of its product to the market, learns from consumers what they like, don't like and want to be included, and then iterates based on this input. (productplan, 2025)



Advantage and Disadvantage of LEAN

Advantages	Disadvantages
<ul style="list-style-type: none"> Removal of waste improves productivity and accelerates the software development life cycle, reducing project effort and financial resources. Lean practices enable timely project delivery, allowing teams to create more features in less time, impressing both financial departments and clients. Empowering project teams fosters decision-making abilities, creating a positive environment and a motivated workforce. Agile engineering, quick iteration evaluation, and improved efficiency between development phases ensure quality in the final product. Iterations, code analysis, pair programming, and other practices accelerate the learning experience for team members. 	<ul style="list-style-type: none"> Lean requires meticulous documentation at each stage, and failure to do so can lead to developmental problems and increased resource requirements. While flexibility is beneficial, excessive flexibility may lead to innovation that deviates from initial goals and fails to succeed. Lean methodologies rely heavily on team cohesion and individual contributions, making it challenging to scale at a faster rate than traditional software development methodologies.

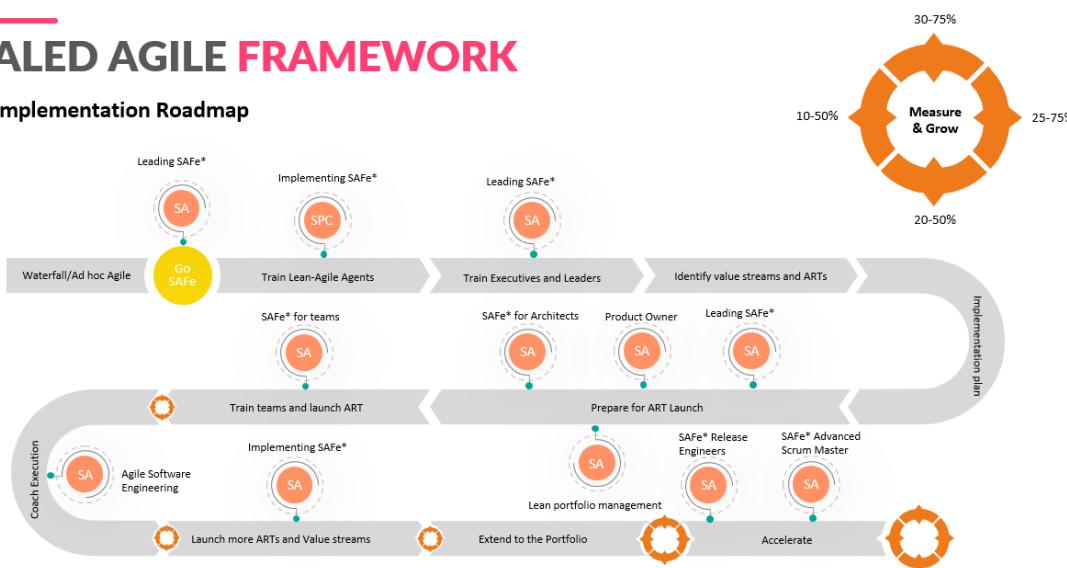
(Raghavan, 2025)

Scale Agile Framework (SAF)

The Scaled Agile Framework, or SAFe, is an agile framework built for development teams. Most notably, SAFe's basis consists of three metaphorical pillars: Team, Program, and Portfolio. Furthermore, SAFe allows a product team freedom. Moreover, it helps handle some of the issues bigger enterprises experience while using Agile. SAFe comprises of a vast knowledge foundation of established best practices. Likewise, product teams utilize SAFe to produce successful software products. (productplan, 2025)

SCALED AGILE FRAMEWORK

SAFe® Implementation Roadmap



Advantage and Disadvantage of scaled agile framework

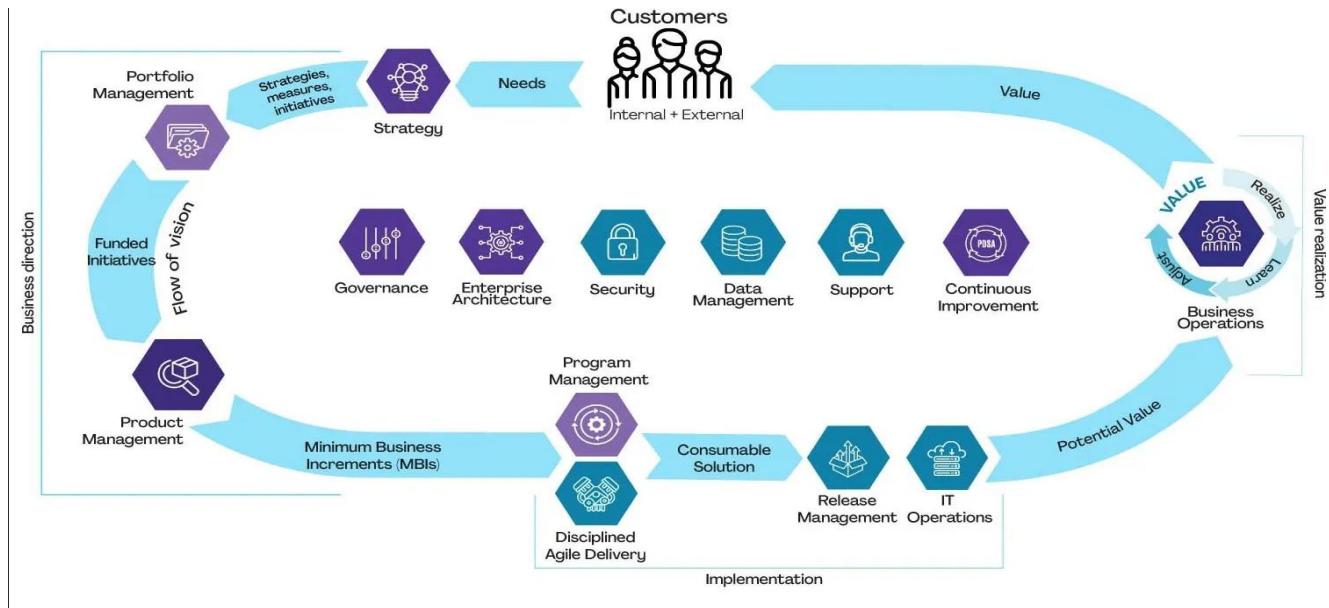
Advantages	Disadvantages
<p>SAFe serves to all organizational levels, delivering significant scaling insights for executives to face enterprise-level concerns.</p> <p>PI planning supports clear vision, handles cross-team dependencies, and encourages cultural sustainability within release trains.</p>	<p>SAFe employs several jargons and terminologies, possibly hindering comprehension and implementation.</p> <p>SAFe may look more directive than collaborative, with numerous levels of</p>

<p>SAFe allows unified planning and execution throughout the enterprise, producing value quicker by integrating rapid input at all levels.</p> <p>SAFe stresses empathy-driven product design, concentrating on understanding end user impact via research and empathy mapping.</p> <p>SAFe dissolves divisions between business and technology, integrating stakeholders' goals with IT delivery via activities like PI planning and system demonstrations.</p> <p>SAFe combines business and IT strategy, fostering hierarchical structure with entrepreneurial networks to survive in the digital age.</p>	<p>administration restricting team flexibility and experimentation.</p> <p>The Innovation or IP sprint is sometimes exploited as a hardening phase, skipping progressive product development and possibly neglecting agile principles.</p> <p>SAFe's concentration on big corporate agility may not be ideal for startups, restricting flexibility in responding to market developments.</p> <p>Some detractors say that SAFe's entire architecture and licensing strategy may limit the genuine ideals and principles of Agile, thereby constraining people and relationships.</p>
---	---

(Dileep, 2025)

Discipline Agile Delivery

Disciplined Agile (DA), is a process choice framework that puts persons first and gives only lightweight assistance to enable teams adapt their processes according to the unique demands of each single project. As a people-first agile framework, DA is in some respects comparable to the Crystal technique. In reality, DA is supposed to be a hybrid technique integrating components of XP, Scrum, Kanban, and other approaches. (productplan, 2025)

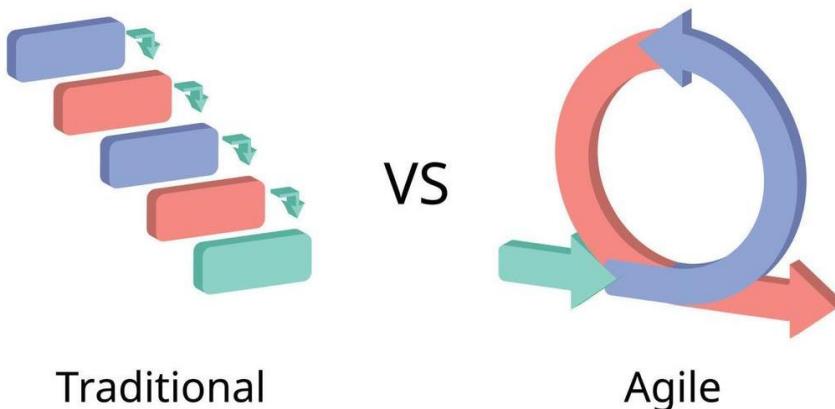


Advantage and Disadvantage of Disciplined Agile

Advantages	Disadvantages
<ul style="list-style-type: none"> • Disciplined Agile provides a flexible framework that enables businesses to adjust their approach depending on their particular environment, objectives, and goals. • It gives a holistic picture of the complete company, combining numerous agile and lean methods to handle business agility from end to end. • Disciplined Agile scales successfully to suit the demands of small teams as well as big companies, delivering guidelines for expanding agile methods throughout the company. • It gives pragmatic advise on different elements of agile and lean approaches, helping 	<p>The framework may look difficult to novices, requiring a steep learning curve to grasp and apply efficiently.</p> <p>Disciplined Agile may create extra overhead in terms of paperwork, responsibilities, and rituals, especially for smaller teams or organizations.</p> <p>Tailoring the framework to unique organizational settings may provide obstacles, needing careful analysis and skill to apply effectively.</p> <p>The flexibility of Disciplined Agile may lead to variance in implementation across teams or organizations, making it tough to standardize processes.</p>

<p>businesses traverse complicated decision-making processes efficiently.</p> <ul style="list-style-type: none"> Disciplined Agile promotes a hybrid approach, allowing businesses to combine agile techniques with current frameworks and processes easily. 	<p>Adopting Disciplined Agile may need major cultural transformations inside the business to align with agile principles and ideals.</p>
---	--

Agile VS Traditional methodology



Comparing Traditional and Agile System Development Methodologies Using Fresh Grocers Web-Based Grocery Delivery System as Case

Software development is very important for modern businesses, and picking the right system analysis method can make a big difference. Two main methods used are traditional and agile. Both have their own strengths and weaknesses depending on the situation. Traditional methods like Waterfall, Spiral, or Prototyping are more structured and follow a fixed step-by-step plan. Everything is well documented before the development even starts. These methods are good when requirements are clear and unlikely to change. For example, in Fresh Grocers' case, if

they already know exactly what features they want for their web-based grocery delivery system, traditional methods could work fine.

On the other side, agile methods like Scrum or Kanban focus more on flexibility, teamwork, and regular customer feedback. Development is done in short cycles called sprints, and after each sprint, feedback is collected and used for improvements. This is great for Fresh Grocers if they want to test features like SMS alerts or agent tracking early and keep updating based on user input. Agile allows the team to adapt to changes easily, unlike traditional models where changing anything later can be very costly and time-consuming.

Overall, traditional methods give better control and are good for long-term planning, while agile offers faster delivery and better customer satisfaction. For Fresh Grocers, if the goal is to launch quickly and adjust based on user needs, agile might be the better choice. But if everything is fixed and well understood, traditional could be more safe and predictable.

Background of the Business Problem – Fresh Grocers

Fresh Grocers is a grocery delivery service that is serving urban and sub-urban areas in Sri Lanka for over 15 years. Due to the increased use of mobile phones and internet in the country, the business decided to launch a web-based system for their customers to make grocery ordering easier and more convenient. The web system includes customer and delivery personnel registration, placing orders, assigning nearest delivery agent, SMS alerts, agent ratings, and manual order entry by a customer service rep.

While starting the project, the company had to choose the right system development approach. Should they go with a structured plan like Waterfall which gives control and proper documentation or should they choose an Agile method which allows more flexibility and customer feedback?

Requirement and Scope Stability

Traditional methodologies like Waterfall depend heavily on fixed and clearly defined requirements. All requirements are collected before development starts. In the case of Fresh Grocers, if the management knows exactly what features they need, like SMS notifications, agent tracking, customer ratings etc., then Waterfall would be suitable. The scope is stable, so

documentation and project planning can go smoothly. This also helps when working with external development companies, as the signed document would ensure both parties are on same page.

However, one weakness of the traditional approach is that once development begins, it becomes hard to make changes. Imagine if during development, Fresh Grocers realize that customers also want to pay using online payment methods, or request a feature to reorder from past purchases, it would be very difficult and costly to go back and add these in a Waterfall model. This can result in delays and increased costs.

In comparison, Agile is much better suited to changes. In Agile, development is done in short iterations or sprints (maybe 1–2 weeks each), and new requirements can be added along the way. If Fresh Grocers uses Scrum and later decides to include a live delivery tracking map or improve customer feedback forms, it can be added to the next sprint. Agile handles uncertainty and evolving features much better than traditional ways.

Customer Involvement and Feedback

One of the biggest advantages of Agile methods is how they involve the customer throughout the whole development process. In Scrum, for example, someone from the Fresh Grocers team would take the role of Product Owner. This person works closely with the development team and makes sure the most important features are built first. After every sprint (which usually lasts two weeks), there is a demo where the team shows what they've built. Stakeholders like Fresh Grocers' managers or staff can give their feedback right away. If users think the delivery agent selection page is not easy to use, the developers can take that feedback and improve it in the next sprint. This kind of constant interaction helps make the system more user-friendly and closely aligned with the business goals.

On the other hand, traditional methods like Waterfall and Iterative Waterfall don't allow this much back-and-forth with customers. Once the initial planning and requirement gathering are done, the customer usually doesn't get involved again until the final product is delivered. For a business like Fresh Grocers, this could be risky. If something important is misunderstood or missed during the planning stage, they might not find out until the end when the full system is almost done. Fixing it at that point could take a lot of time and money. So, while traditional

methods give clear structure, they are not great when customer needs might change or if feedback is needed during development. Agile solves this with its focus on collaboration and flexibility.

Risk Management

The Spiral model, which is part of traditional system development approaches, is very good when it comes to handling risks. It works by combining elements of both design and prototyping in a repetitive process. This allows risk assessment at each stage before moving forward. For a business like Fresh Grocers, which might be implementing new features such as SMS alerts for order updates or real-time tracking of delivery agents, the Spiral model offers a way to carefully study if those features are technically possible and acceptable to customers. If there's a chance that customers won't like a certain function or the system might not handle it well, these concerns can be spotted early. However, this model is not easy to manage. It needs experienced project leaders who understand both planning and technical analysis, and it often takes more time and resources than simpler models.

In comparison, Agile takes a different approach to managing risk. Instead of doing deep risk analysis upfront, Agile teams deal with risk by launching small, working pieces of software early and often. This way, if a feature like agent tracking doesn't work properly or is not liked by customers, it can be fixed or changed in the next release. This method helps avoid wasting time building something that users won't want or need. It also fits better for fast-moving industries like online grocery delivery, where companies like Fresh Grocers must quickly adapt to customer feedback and market changes. Agile's fast response helps reduce risk in a more flexible way.

Team Communication and Management

Traditional models are more suitable when team structure is hierarchical and each team member has a clearly defined role. Managers can easily track progress since everything is planned from beginning. Fresh Grocers would find it easier to outsource work to external IT vendors using Waterfall as all contracts are based on fixed scope, timelines and documentation.

In contrast, Agile requires close teamwork, continuous communication, and cross-functional teams. If Fresh Grocers have an in-house development team or closely collaborating with a vendor, Agile will work better. Daily stand-up meetings, sprint reviews and retrospectives will help the team stay on track and fix issues faster. However, if the company lacks an agile mindset or doesn't have experience with iterative methods, it might face challenges in using Agile effectively.

Documentation vs Working Software

In traditional software development methods, documentation is a very important part of the process. Before the actual development begins, teams prepare detailed documents like Software Requirement Specifications (SRS), design diagrams, test plans, and user manuals. Everything is clearly planned and recorded. For a company like Fresh Grocers, this level of documentation can be very helpful, especially if they want to keep a record for legal purposes, to avoid misunderstandings with third-party vendors, or to train new developers in the future. Having full documentation means there is a clear picture of how the system should work and who is responsible for what.

Agile, on the other hand, takes a different approach. It still uses documentation, but not to the same extent. Agile teams prefer to focus more on delivering working software rather than spending too much time on paperwork. Documentation is only created if it adds value and helps the team or the customer. For Fresh Grocers, if the goal is to get the system live quickly and adjust it later based on real user feedback, then Agile is a better fit. It allows the team to build and improve features step by step, without being slowed down by too much planning. In fast-moving markets like online grocery delivery, being able to respond quickly is more useful than having detailed documents. So, while traditional methods are good for stability and control, Agile is better when flexibility and speed are more important for business success.

Time to Market

Agile has a clear edge when it comes to delivering software fast. With each sprint, some part of the system is delivered and can be used. For instance, the registration and login system could

be completed in the first two weeks, order placement in next, and SMS feature in the third sprint. This way, Fresh Grocers can start testing the system early or even allow beta users to try it.

Traditional methods may take months before the full system is ready. All design and coding must be completed before testing begins. So, Fresh Grocers may lose the competitive edge in a fast-changing market.

Scalability and Complexity

When dealing with large-scale and complex systems, traditional methodologies like Waterfall or Iterative Waterfall usually work better because they offer more control, structure, and documentation. In projects where there are hundreds of dependencies and integration with multiple systems like an ERP that handles inventory, accounting, customer service, and logistics all together it's important to plan everything carefully from the start. For example, if Fresh Grocers was developing a full ERP platform, using a traditional method would help ensure that each department's needs are documented and addressed properly. Detailed planning reduces confusion and helps manage risks that come from system dependencies.

However, the current system Fresh Grocers is working on is a web-based grocery ordering and delivery platform for customers. This kind of project is not as large or complex as an ERP and needs more flexibility, especially because customer needs might change during the development. Agile methodologies like Scrum or Kanban would be more useful here because they allow fast updates and regular feedback from users. If users say they want to add a live chat or a loyalty points system, the team can adjust in the next sprint without needing to redo the whole plan. Also, the simpler structure of Agile helps small teams work faster and focus on delivering working features in short cycles. So, while traditional methods are best for complex enterprise-level systems, Agile is a better choice for smaller, customer-facing apps like Fresh Grocers' delivery platform.

Change Management

One of the biggest downsides of traditional system development methods like Waterfall is how poorly they handle changes. Once the initial requirement gathering is finished and the documents are signed off, making any changes becomes very difficult, time-consuming, and expensive. For example, in the Fresh Grocers project, imagine the web platform is halfway through development and suddenly customers start asking for a mobile app version with push notifications. In a traditional setup, this would cause major delays. The team would have to go back to the requirement phase, make new design documents, change timelines, and possibly rewrite large parts of the code. In some cases, the company might even decide not to include the change at all because it's just too complicated.

Agile approaches, however, treat change as something natural and expected. In Scrum or Kanban, new features or user demands like push notifications for a mobile app can be added to the product backlog at any time. The team can then choose to prioritize them in future sprints. This means Fresh Grocers can quickly respond to user trends and market conditions without going through a long and formal change process. Developers, testers, and business people work closely together, and feedback is given regularly, which helps avoid wasting time and resources. In today's fast-moving digital world, this ability to adapt quickly gives Agile a big advantage over traditional methods.

Activity 02

Feasibility Study



(senwork.com, 2024)

A feasibility study is a component of any project's or plan's early design stage. It is carried out in order to discover the strengths and weaknesses of a prospective project or an existing firm objectively. It may aid in identifying and assessing the natural environment's potential and risks, as well as the resources needed for the project and its chances of success. (CFI Team, 2025)

Feasibility studies are important because,

Feasibility studies are the key step that enables the project planner to have a clear vision of how their project can be completed. These investigations offer talent for detection of probable threats, problems, and potentials for the project. Thus, using these information, stakeholders are able to make data based decisions on how resources are allocated and strategic direction of the company. For instance, through cost-benefit analysis and feasibility study, economic consequences of the project are being addressed in order to make sure that there is a match with the goals and targets of the organization. Besides that, these studies can foster stakeholder involvement in the process, offering the chance for officials to provide relevant information and work towards a common purpose. Compliance issues and regulatory risks can be scrutinized by legal and regulatory requirements in a feasibility study. This way, the organization can mitigate the risks and follow relevant regulations. Eventually, feasibility

studies are a very significant survey for weighing up project feasibility, handling risks and achieving the project objectives.

Feasibility Report

A feasibility study is a report that assesses the viability of a collection of potential project pathways or solutions. The person who writes a feasibility report assesses the feasibility of several ideas and then recommends the best alternative. They then give the feasibility study and offer their proposal to their company.

Sections of a feasibility report

- Problem definition
- Scope and objectives of the new system
- List of key stakeholders and their roles
- Models of the new system (e.g. Use Case Diagram, DFD etc.)
- Cost/benefit analysis of the proposed system
- Risk analysis of the new system
- Budget and funding plan
- Schedule (time duration)
- Alternative solutions with cost/benefit analysis
- SWOT Analysis of the proposed system (Strengths, Weaknesses, Opportunities, Threats)

Feasibility Study Template

- 1) Introduction
- 2) Background
- 3) Outline of the system
- 4) Models of the system
- 5) Overview of the alternatives
- 6) Conclusion
- 7) Recommendation

Feasibility Report for Fresh Grocers

Feasibility Report

Fresh Grocers web based system

Prepared by: [Junior System Analyst, Fresh Grocers]

Date: April 04 , 2025

Introduction

Fresh Grocers is a well-established grocery delivery service in Sri Lanka, serving customers in urban and suburban areas for over fifteen years. Over the years, the company has built a strong reputation for providing fresh and high-quality groceries with efficient home delivery services. However, with the increasing demand for online shopping and the widespread use of smartphones and internet services, the company recognizes the need to modernize its operations. To remain competitive in the growing e-commerce sector, Fresh Grocers plans to develop a web-based platform that will allow customers to conveniently place grocery orders online. This system is expected to enhance customer satisfaction, streamline the order management process, and improve delivery efficiency.

The feasibility study is conducted to determine whether this proposed system is practical, cost-effective, and beneficial for the company. It evaluates several critical aspects, including technical feasibility, which assesses whether the existing IT infrastructure, internet connectivity, and software requirements can support the platform. The study also considers economic feasibility to analyze whether the investment in system development is financially justifiable, ensuring that the company achieves long-term profitability. Operational feasibility is another crucial factor, as it examines how well the new system will integrate with current business operations, including order processing, inventory management, and delivery coordination. Additionally, legal feasibility ensures that the platform adheres to e-commerce regulations, data privacy laws, and consumer protection guidelines, guaranteeing a secure and compliant online shopping experience for customers.

One of the key features of the proposed system is an automated delivery agent assignment process, which will optimize the allocation of delivery personnel based on customer locations.

This feature will help minimize delivery time, reduce operational costs, and enhance service quality. Customers will receive real-time updates on their order status, including estimated delivery time and assigned delivery personnel details, improving transparency and trust. Furthermore, the system will include a customer feedback mechanism, allowing users to rate their delivery experience, which will help Fresh Grocers maintain high service standards.

By carefully analyzing all these feasibility factors, this study provides a comprehensive assessment of the potential benefits and challenges associated with implementing the web-based grocery ordering system. The findings of this report will help stakeholders, including company management and investors, make well-informed decisions regarding the system's development. Ultimately, the goal is to create an efficient, user-friendly, and reliable online platform that meets customer expectations, enhances operational efficiency, and ensures long-term business growth for Fresh Grocers.

Problem Definition

Fresh Grocers has been delivering groceries in Sri Lanka for over fifteen years, primarily serving urban and suburban customers. However, with increasing demand for online shopping, the company faces challenges in maintaining efficiency and meeting customer expectations. Currently, orders are taken manually via phone calls, leading to errors, delays, and inconvenience for customers. There is no real-time product availability, order tracking, or automated confirmation, making the process inefficient.

Another major issue is inefficient delivery management. Delivery agents are assigned manually, often resulting in delays and poor resource utilization. Without an automated system, deliveries can be slow, and tracking their progress becomes difficult. Additionally, agents have no structured way to update their availability, affecting the overall efficiency of the delivery process.

Customer feedback is another concern, as there is no system in place for customers to rate delivery personnel or provide reviews. This lack of structured feedback makes it difficult for the company to assess service quality and improve performance.

To address these challenges, Fresh Grocers needs a web-based ordering system with automated delivery assignments and real-time tracking. This will streamline the ordering process, reduce

delays, and enhance customer experience. Additionally, a feedback system will help improve service quality. Implementing these solutions will modernize operations, improve efficiency, and allow Fresh Grocers to stay competitive in the online grocery market.

Scope of the System

The scope of the web-based system for Fresh Grocers is to provide a comprehensive solution that enhances the customer experience, streamlines operations, and improves delivery management. The system will cover the following key functionalities:

1. Customer Registration and Order Placement

The system will allow customers to register and create accounts to access the platform. Once registered, they can browse products, add items to their cart, and place orders directly through the website. Customers will be able to track their orders in real-time, receive updates, and access their order history.

2. Order Confirmation and Notifications

Once an order is placed, the system will send an SMS confirmation to the customer, along with delivery details such as the estimated delivery time and the contact information of the assigned delivery agent. This feature ensures transparency and keeps customers informed about their order status.

3. Delivery Agent Assignment and Management

The system will automatically assign the nearest available delivery agent to each order based on customer location. Delivery agents will have the ability to update their status and location in the system, which will help optimize the delivery process and reduce delays.

4. Manual Order Entry by Customer Service Representatives

The system will allow customer service representatives to manually enter orders on behalf of customers who may not have access to the online platform. These orders will be processed just like regular online orders, with automatic SMS confirmations sent to customers.

5. Customer Feedback System

The platform will feature a feedback mechanism that allows customers to rate their delivery experience. This feedback will help Fresh Grocers assess service quality, identify areas for improvement, and maintain customer satisfaction.

6. Inventory and Order Management

The system will include tools for managing inventory and orders. It will help track product availability, alerting the company when stock levels are low and preventing out-of-stock issues. Additionally, it will streamline the order fulfillment process, improving operational efficiency.

The scope of this system does not include features like mobile app development or integration with third-party services. However, these could be considered in future phases based on the system's success and user demand. The main focus is on improving the online ordering process, enhancing customer service, and optimizing delivery management.

Objectives of the System

The primary objectives of the web-based system for Fresh Grocers are as follows

1. Enhance Customer Experience

Provide customers with a seamless online ordering platform that allows them to easily browse products, place orders, and track deliveries in real-time. The system should ensure convenience, quick order processing, and reliable delivery.

2. Improve Operational Efficiency

Automate the ordering, delivery assignment, and inventory management processes to reduce human errors, minimize delays, and improve resource utilization. This will streamline operations and reduce manual intervention.

3. Optimize Delivery Management

Automatically assign the nearest available delivery agent to each order based on customer location, ensuring faster deliveries and improved service. The system will also allow delivery agents to update their status and location, improving tracking and operational control.

4. Provide Real-Time Notifications and Updates

Send real-time updates to customers regarding their order status, estimated delivery time, and assigned delivery agent. This will improve customer transparency and satisfaction.

5. Facilitate Customer Feedback

Implement a feedback system that allows customers to rate their delivery experience. This will help the company assess service quality, identify areas for improvement, and maintain high customer satisfaction levels.

6. Support Manual Order Entry

Allow customer service representatives to manually enter orders for customers who may not have access to the online platform, ensuring that all customers can benefit from the services provided by Fresh Grocers.

7. Improve Inventory Management

Provide tools to efficiently manage inventory, track product availability, and prevent stock shortages, ensuring that the company can fulfill orders promptly.

List of Key Stakeholders

1. Customers

The end users who will interact with the platform, placing orders and receiving deliveries. They are the primary stakeholders, as the system's success depends on providing a positive user experience.

2. Delivery Agents

Individuals responsible for delivering groceries to customers. They will use the system to receive delivery assignments, update their status, and track their deliveries.

3. Customer Service Representatives

Employees who assist customers with placing orders, addressing queries, and handling orders manually for customers who don't have access to the online system.

4. Fresh Grocers Management Team

The company's leadership and decision-makers who will oversee the development and implementation of the system, ensuring it aligns with business objectives.

5. IT and Development Team

The technical team responsible for designing, developing, testing, and maintaining the web-based system. They will ensure the system is functional, secure, and scalable.

6. Suppliers

Vendors who supply the grocery items. They are crucial to ensuring inventory management is accurate and that the system can track product availability in real-time.

7. Regulatory Authorities

Government agencies responsible for e-commerce regulations, data protection, and consumer rights. They will ensure that the system complies with relevant laws and guidelines.

8. Investors/Stakeholders

Financial backers or other key stakeholders who have a vested interest in the company's

performance and the success of the new system. They will assess the financial viability of the project.

Each stakeholder group plays an essential role in ensuring the success of the system, from its development to its usage and ongoing maintenance.

Purpose of a Feasibility Study

The first step for entrepreneurs and business owners who might want to start a new business venture, is a feasibility study. The main aim of running a business idea assessment is to identify how viable the offered business idea is at making such a business a reality by analyzing different factors which might affect its success. The feasibility study offers a bird's eye view of the business landscape and aids owners to decide whether or not to continue, rewrite or cancel the venture

- Who are the competitors?

You have to analyze the competitive landscape to know how the market dynamics are. This feasibility study discusses existing competitors, their strengths and weaknesses, and position in the market. It aids in identifying differentiating opportunities and informs the ways to have a competitive advantage through the differentiation.

- How much is it going to cost to set up and run this business?

A cost analysis in detail is a necessary part of the feasibility study. The first shows the starting costs (equipment, inventory and licensing) and the last show ongoing operational costs (rent, utilities, salaries, marketing). It is important for business owner to understand these costs so that they can use this to determine how much they can spend on a budget for their business.

- What are the risks and rewards of this business?

There is a risk to every business venture. Potential risks like market fluctuations, regulatory changes, regular operational challenges are identified by the feasibility study. It also determines the possible rewards such as revenue and profitability. This risk-weighs analysis helps business owners to weight the financial benefits with the risk that's involved.

- Just how big a business is this?

Assessing the viability of the business is very dependent on these projections of revenue. The feasibility study provides an estimate of potential sales based on market research, pricing strategies and forecasts of potential demand. Revenue potential important for owners to grasp in order to know if the business can make enough money to keep the doors open and the dream intact.

Key Purposes of a Feasibility Study

1. Evaluate Project Viability

The feasibility study helps assess whether the proposed project is technically, financially, and operationally viable. It answers questions like, “Can this project be done?” and “Is it worth doing?”. For large-scale projects like software systems, it helps to determine if the resources and budget required are justifiable based on the potential benefits.

2. Risk Identification and Management

It identifies potential risks, including technical, financial, and operational risks, and helps in developing mitigation strategies. This is essential to avoid issues such as cost overruns or project delays during the software development lifecycle.

3. Clearer Decision-Making

the study provides valuable insights that assist stakeholders in making informed decisions about the project. Based on the findings, decision-makers can either approve the project, seek more information, or discontinue it altogether.

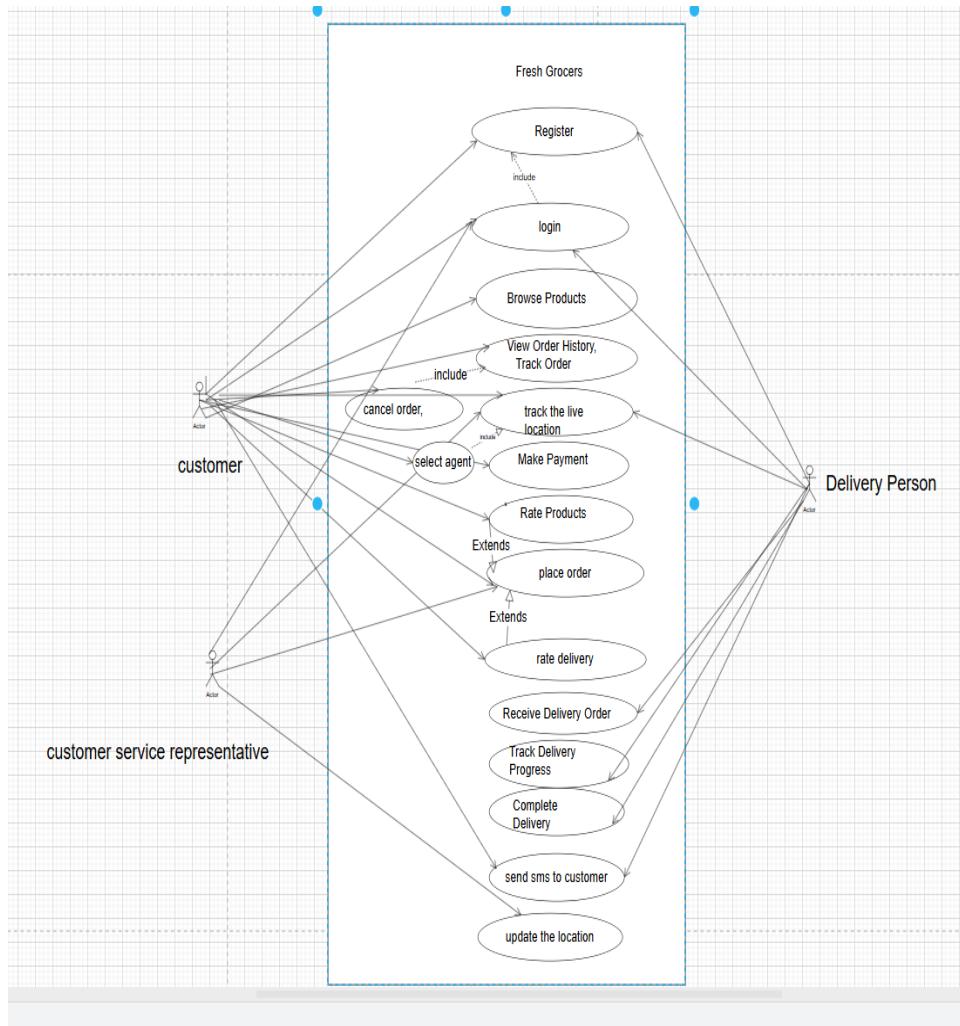
4. Defines Scope and Constraints

The study helps outline the scope of the project and defines constraints, such as budget limits, technical limitations, time constraints, and other operational challenges that need to be considered during project planning.

Determine Legal and Compliance Issues

the study also considers any legal, regulatory, or compliance requirements that might affect the project.

Model of New system



FEASIBILITY CRITERIA

Technical Feasibility

Software	Description
Web development framework	React JS, Laravel, Tailwind CSS
Data base management system	MySQL
SMS Integration Service	Twilio

Geolocation API	Google Maps
Hardware	Description
Web Server	AWS EC2 instances, DigitalOcean droplets
Database Server	AWS RDS, Google Cloud SQL
Devices	Tablets, Laptops
Team	Quantity
Web Developers	3
UI/UX Designers	2
QA testers	2
Project Manager	1
System Administrator	1
Support Personnel	2

From these data it can be seen that the new project is technically feasible.

Economic Feasibility

The economic feasibility given below is describe the cost for this project.

Software	Description	Cost (lkr)
Web development framework	React JS, Laravel, Tailwind CSS	-
Data base management system	MySQL	-
SMS Integration Service	Twilio	-
Geolocation API	Google Maps	-
Hardware	Description	
Web Server	AWS EC2 instances, DigitalOcean droplets	150 000
Database Server	AWS RDS, Google Cloud SQL	200 000
Devices	Tablets, Laptops, Computers	500 000
Team	Quantity	
Web Developers	3	360 000
UI/UX Designers	2	150 000
QA testers	2	120 000

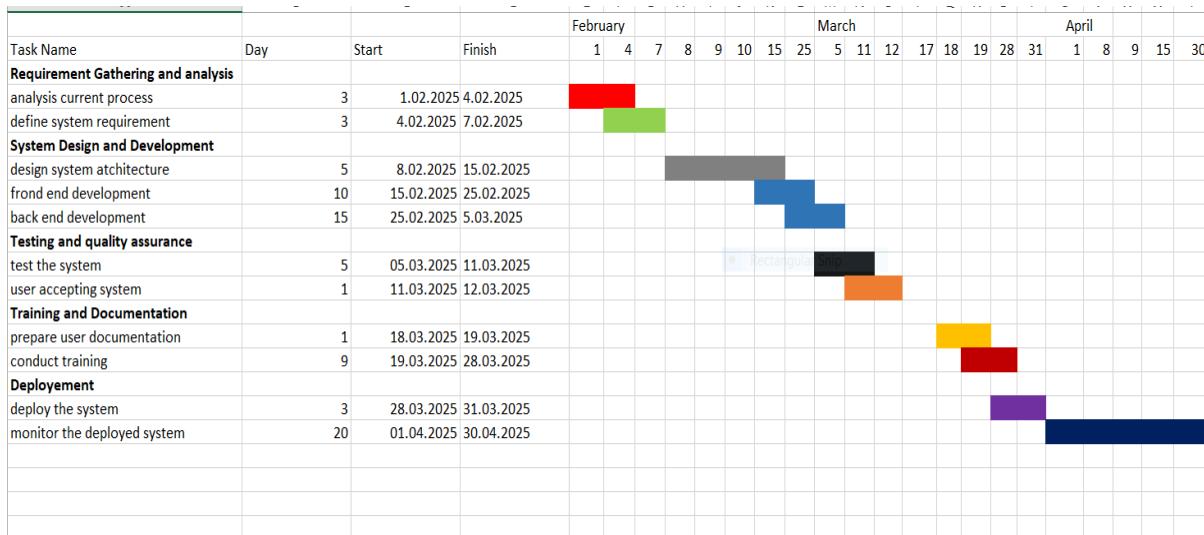
Project Manager	1	145 000
System Administrator	1	85 000
Support Personnel	2	80 000

INITIAL COST	
Development cost	940 000
Hardware cost	850 000
Total	1 790 000
ONGOING COST (Every Year)	
IT Staff Salaries	400 000
Maintenance	100 000
Total	500 000
BENEFITS	
Increased Users	1 800 000
Staff Reduction	300 000
Total	2 100 000

	Year 0	Year 1	Year 2	Year 3	Year 4
Cost	1790000	500000	500000	500000	500000
Benefit	0	2100000	2100000	2100000	2100000
Net Benefit	0	500000	500000	500000	500000
Accumulated Net Benefit	0	1600000	3200000	4800000	6400000

From these data it can be seen that the new project is highly beneficial economically. Moreover, it has a very short Pay Back period of just 1 years,

Schedule Feasibility



Legal Feasibility

Legal feasibility ensures that the web-based system for Fresh Grocers complies with all applicable laws and regulations. This is a crucial step in determining whether the system can legally operate without violating any laws that could lead to penalties or reputational damage. The legal considerations for Fresh Grocers primarily involve data protection laws, which regulate the collection, storage, and processing of personal data. In Sri Lanka, the Personal Data Protection Act (PDPA) mandates that businesses must protect customers' personal and financial information from unauthorized access or misuse. The system must be designed to store customer data securely and provide users with the right to access, update, and delete their personal data. This is essential for building customer trust and ensuring compliance with legal requirements.

The system must also comply with e-commerce regulations, which govern online transactions. These regulations ensure that consumers are provided with clear information about the terms and conditions of service, payment methods, and refund policies. The platform must implement secure payment gateways that protect customers' financial information during transactions. Additionally, it is important for the system to maintain transparency by displaying accurate product descriptions, delivery timelines, and any fees associated with the service. By adhering

to these regulations, Fresh Grocers can minimize the risk of customer disputes and ensure fair business practices.

Moreover, consumer protection laws are a vital consideration. These laws protect consumers from unfair business practices and ensure that their rights are upheld. For example, the system must offer customers an easy way to lodge complaints or report issues with their orders. It must also clearly communicate the terms and conditions of orders, returns, and refunds, ensuring that customers are aware of their rights. Additionally, the legal feasibility of the system includes ensuring compliance with taxation laws, such as collecting the correct VAT or other applicable taxes. By addressing all these legal aspects, the company will not only be compliant with relevant laws but also demonstrate a commitment to ethical business practices, which helps to build long-term customer trust and loyalty.

Cultural Feasibility

Cultural feasibility evaluates how well the web-based system aligns with the local culture, societal norms, and consumer behaviors. In Sri Lanka, many customers are still not fully accustomed to online shopping, especially older generations or those in rural areas. Therefore, the system must be user-friendly and accessible to people with varying levels of digital literacy. The design should prioritize simplicity, with intuitive navigation and clear instructions to ensure that all customers, regardless of their technical skills, can easily place orders.

Another important aspect of cultural feasibility is language support. Sri Lanka is a multilingual country with Sinhala, Tamil, and English being the primary languages spoken. To ensure that the system caters to a broad audience, the web-based platform must offer multilingual support, allowing users to navigate the platform in their preferred language. This will make the system more accessible and appealing to a diverse customer base. Additionally, localization of the content is important. The system should present information in a culturally relevant manner, respecting local customs and preferences.

Furthermore, payment methods must be considered in the context of Sri Lankan cultural norms. While digital payments are becoming more common, many customers, especially those in rural areas, may still prefer to pay via cash on delivery (COD) rather than using online payment methods like credit or debit cards. The platform must, therefore, support multiple payment options, including COD, to ensure that it is inclusive and accessible to all customer segments.

Offering flexible payment methods will increase customer trust in the platform and encourage more people to adopt online grocery shopping.

Delivery preferences also play a role in cultural feasibility. In Sri Lanka, customers often expect personalized service, especially in terms of delivery. Ensuring that the delivery system is flexible and responsive to customer needs, including accommodating specific time slots for delivery, can enhance customer satisfaction. Understanding and adapting to these cultural preferences will significantly contribute to the platform's success and customer retention.

Operational Feasibility

Operational feasibility assesses whether Fresh Grocers can integrate the web-based system into its existing operations effectively. This includes evaluating whether the company has the necessary resources, infrastructure, and workforce to support the new system. The system must seamlessly integrate with Fresh Grocers' current inventory management, order processing, and delivery tracking systems. If there are gaps in the current infrastructure, additional investments in hardware or software will be required. For example, the company may need to upgrade its servers or cloud storage to handle the data generated by the new system, ensuring that the system runs smoothly even during peak hours.

Another key consideration is employee training. Customer service representatives and delivery personnel will need to be trained on how to use the new system. The system should be intuitive and user-friendly to minimize the learning curve, but comprehensive training will still be necessary to ensure smooth operations. A lack of proper training could lead to errors in order processing or delivery, which could negatively impact customer satisfaction.

Scalability is another important aspect of operational feasibility. As Fresh Grocers grows, the system should be able to handle increased traffic, more orders, and a larger number of delivery agents. The platform should be built with scalability in mind, ensuring that additional resources, such as servers or cloud infrastructure, can be added as the business expands. This will prevent performance bottlenecks and ensure that the system continues to operate efficiently as demand increases.

Lastly, resource allocation is critical for operational success. Fresh Grocers will need to allocate sufficient resources, both financial and human, to the project. This includes hiring developers,

purchasing software licenses, and investing in the necessary infrastructure. Operational feasibility ensures that the company can manage these resources effectively and that the system will deliver a positive return on investment in the long run.

Alternative Solutions

Alternative solutions refer to other potential approaches to meeting Fresh Grocers' business needs in case the proposed web-based system faces significant challenges. One alternative is the development of a mobile app, which may provide a more convenient and accessible platform for customers, especially those who prefer using smartphones over desktop computers. A mobile app could also offer additional features, such as location tracking, push notifications, and personalized offers, enhancing the customer experience.

Another option is a hybrid system, where Fresh Grocers could implement a simple online ordering platform while continuing to accept orders via phone calls. This hybrid approach would allow customers who are not comfortable with online shopping to place orders through traditional means, ensuring that the company does not exclude any potential customer segment.

An additional solution is outsourcing delivery services to third-party logistics companies. By doing so, Fresh Grocers could eliminate the need for building and maintaining an internal delivery network, which could reduce operational costs and complexity. Outsourcing would allow the company to focus on its core competencies, such as inventory management and customer service, while third-party providers handle deliveries more efficiently.

Lastly, incremental development could be considered as an alternative. Instead of launching a fully-featured web-based platform all at once, Fresh Grocers could start with basic functionalities, such as order placement and delivery scheduling, and gradually expand the system over time. This approach would reduce upfront costs and allow the company to gather user feedback to improve the system progressively.

By considering these alternative solutions, Fresh Grocers can identify the most cost-effective and efficient path forward, ensuring that the company meets its business objectives while managing risks and challenges.

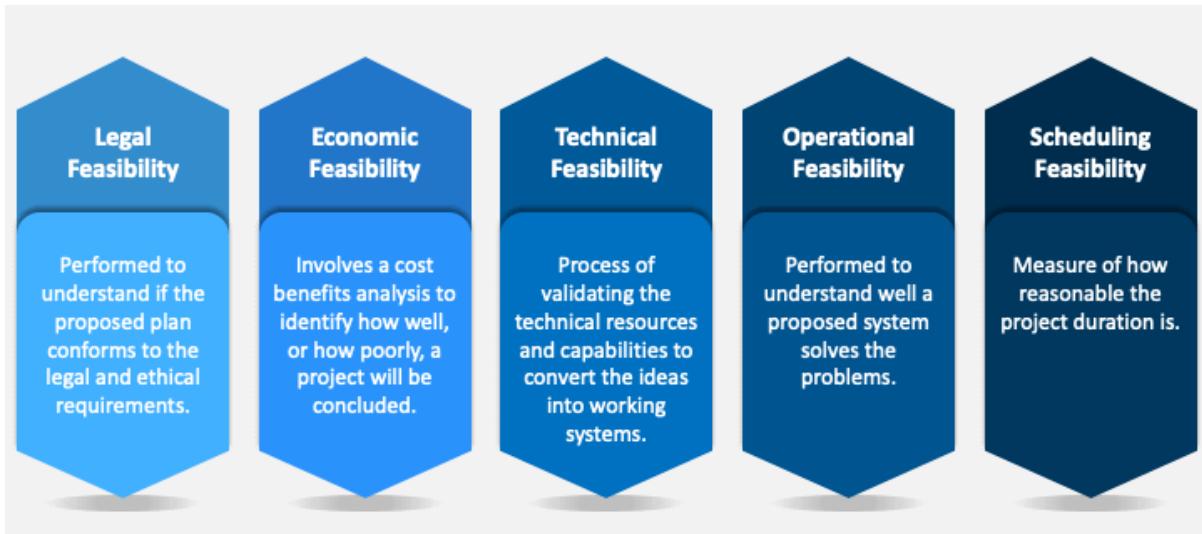
Criteria	Off-the-Shelf Financial Software	In-House Development
Customization	Limited, cannot fully align with specific needs	Full customization, tailored to business requirements
Initial Cost	Lower	Higher
Implementation Time	Shorter (1-3 months)	Longer (6-12 months)
Scalability	Limited, future growth may require additional purchases	Fully scalable, future updates can be easily implemented
Security	Standard security features may lack specific compliance requirements	Full control over security, can meet all regulatory standards
Ongoing Costs	Annual license fees (15-20% of initial cost)	Maintenance costs (15-20% of initial cost)
Vendor Dependency	High, reliant on vendor for updates and support	Low, no dependency on external vendors

Conclusion

In conclusion, the feasibility study for the web-based system of Fresh Grocers has provided a comprehensive evaluation of the project's viability from several perspectives. Legal feasibility ensures that the platform complies with regulations such as data protection and consumer protection laws, safeguarding both the company and its customers. Cultural feasibility highlights the need for a user-friendly system with multilingual support and flexible payment options, ensuring inclusivity across diverse customer segments. Operational feasibility demonstrates that, with the right infrastructure, training, and resource allocation, Fresh Grocers can successfully integrate the system into its existing operations while scaling for future growth. Alternative solutions, such as a mobile app or hybrid system, offer viable options to overcome potential challenges in system adoption or delivery logistics. Overall, the study

confirms that the development of a web-based platform is not only feasible but also essential for Fresh Grocers to remain competitive in the growing online grocery market. With careful attention to legal, cultural, and operational factors, the project promises to enhance customer experience, improve operational efficiency, and drive long-term success for the business.

Feasibility criteria types



Technical Feasibility

In Technical Feasibility, present resources, including hardware and software, as well as necessary technologies, are analyzed/assessed in order to create a project. This technical feasibility analysis determines if the necessary resources and technology are available for project development. Along with this, the feasibility study examines the technical skills and capabilities of the technical team, if existing technology can be utilized, whether maintenance and upgrade are simple or difficult for the chosen technology.

If technical feasibility is not taken into account, a project may face unforeseen technical obstacles, resulting in delays, cost overruns, and project failure. It assures that the project can be completed with the technical resources and skills that are available.

Operational Feasibility

Operational Feasibility degree is analyzed on how well will product be able to provide service to requirements and how much easy to operate and maintaining the product after its deployment. This along with these operational scopes of determining usability of product, determining suggested solution by software development team is acceptable or not, etc.

Economic Feasibility

The cost and utility of the project are examined in the Economic Feasibility study. This feasibility study provides a detailed analysis of the project's development costs, which include all essential costs for final development such as hardware and software resources, design and development costs, and operations costs, among other things. Following that, it is determined if the initiative will be financially advantageous to the organization.

Without doing an economic feasibility analysis, a company may invest in a project that turns out to be financially unsustainable or inefficient. It assists in making sound judgments about resource allocation and project finance.

Legal Feasibility

The project is examined from a legal standpoint in the Legal Feasibility study. This involves examining project legal implementation difficulties such as data protection acts or social media legislation, project certificate, license, copyright, and so on. Overall, a Legal Feasibility research is a research to determine whether a proposed project meets legal and ethical standards.

Neglecting legal feasibility can lead to legal wrangling, penalties, or the necessity for costly project adjustments to meet legal standards. It aids in the prevention of legal and ethical difficulties that might jeopardize the project's reputation or legal status.

Schedule Feasibility

In schedule Feasibility Study the timeliness/deadlines are analyzed on the proposed project, consisting how much time the teams will take to complete the final project which has a huge effect to the organization as the project is not assured if not finished within the deadline.

Cultural and social feasibility

Cultural and social feasibility studies look at how a project could affect the cultural and social fabric of a community or stakeholders. They take into account aspects such as societal acceptance, cultural sensitivity, and community support for the initiative. (Satyabrata_Jena, 2025)

Neglecting cultural and social feasibility might result in stakeholder opposition, unfavorable public perception, or societal conflicts. It contributes to ensuring that the project is consistent with social norms and expectations.

Examples for Feasibility criteria types

Feasibility Criteria	Example
Technical Feasibility	A small bakery is considering implementing a new online ordering system. The technical feasibility involves assessing whether the bakery has the necessary infrastructure, such as internet connectivity and hardware, to support the system. It also involves evaluating whether the bakery's existing website platform can integrate with the new ordering system.
Operational Feasibility	A small retail store considers implementing a new inventory management system to streamline operations. Operational feasibility involves evaluating how easily employees can adapt to the new system and whether it enhances workflow efficiency. Conducting training sessions for staff and piloting the system with a small team can help assess its impact on day-to-day operations before full implementation.
Economic Feasibility	A local coffee shop wants to install solar panels to reduce energy costs. Economic feasibility involves calculating the upfront investment required for purchasing and installing the solar panels, as well as estimating the long-term savings on electricity bills. If the projected savings outweigh the initial investment within a reasonable payback period, the project is economically feasible.
Legal Feasibility	A startup app development company plans to launch a mobile application that collects user data for targeted advertising. Legal feasibility involves conducting research to ensure compliance with data protection laws, such as the General Data Protection Regulation (GDPR) or the California Consumer Privacy Act (CCPA). The company must assess whether its data collection practices align with regulatory requirements to avoid legal issues.
Schedule Feasibility	A construction company is planning to build a new office building within a specific timeframe. Schedule feasibility involves evaluating whether the proposed project timeline is realistic and achievable. Factors such as the

	availability of materials, labor resources, and permits play a crucial role in determining schedule feasibility. The company must consider potential delays due to unforeseen circumstances like adverse weather conditions or regulatory approval processes. Conducting a detailed project schedule analysis and allocating sufficient time for each construction phase can help ensure that the project stays on track and meets its deadlines.
Cultural and Social Feasibility	An international fast-food chain wants to introduce a new menu item tailored to local tastes in a foreign market. Cultural feasibility involves understanding the cultural preferences, dietary habits, and food customs of the target market. Conducting surveys or focus groups to gather feedback from locals can help assess whether the new menu item aligns with cultural norms and preferences.

Evaluate the Importance of the Feasibility Criteria in the Systems Investigation for the Business-Related Problem (Fresh Grocers Web-Based Platform)

In any systems investigation for solving a business-related problem, understanding the feasibility criteria is one of the most critical steps. Without looking at if something is truly possible and practical, just designing a system blindly can be risky and costly for a business. Feasibility is about checking if the system can actually be developed, if it will be accepted by the users, and if it will meet the goals of the organization in real life, not just on paper.

For Fresh Grocers, who want to build a web-based grocery ordering and delivery system in Sri Lanka, understanding these feasibility areas are even more important. This is because they are investing in a new digital business model and are depending on technology to compete and serve customers. A failure in development, usability or cost could make the entire effort useless. So, evaluating all main types of feasibility technical, economic, legal, cultural and operational is necessary before starting development.

Technical Feasibility

Technical feasibility means checking if the system can be developed using the existing or new technology that the business can access. For Fresh Grocers, this involves looking at what kind

of hardware, software, development tools, and network infrastructure is available. Also, if the developers and IT team have the required skills to build, test and maintain the system.

For example, if Fresh Grocers plan to offer live tracking of delivery agents on a map, they must evaluate if their current server setup can handle real-time location updates. Also, if they want to include SMS alerts, they need to ensure their backend system can integrate with telecom APIs. If this is not checked early and turns out to be technically too complex or expensive later, the business may suffer.

Another technical point is whether the users' devices can support the system. If most customers in rural areas use low-end phones with weak internet, building a high-graphic heavy web app will not work well. So understanding the technical environment and adjusting the system's design is very important for successful development.

Economic Feasibility

Economic feasibility checks whether the proposed system is financially viable. This means understanding all the development and long-term costs, and comparing them with the expected financial benefits. In Fresh Grocers' case, this includes the cost of web development, database servers, hiring delivery agents, payment gateway setup, and digital marketing.

If Fresh Grocers don't do proper financial planning, they might run out of money before the system is complete or the system may not bring enough return on investment (ROI). Let's say they invest LKR 1 million on development but only make LKR 100,000 profit per month it would take a long time to recover the cost. Also, ongoing maintenance and updates cost money. So, these should be estimated too.

An economic feasibility study will also help decide what features are essential and what can be delayed. Maybe launching with only the web-based platform first and adding a mobile app later could save money. By breaking down the costs clearly, it's easier to convince stakeholders or investors that the system is a good investment.

Legal Feasibility

Legal feasibility is another critical area, especially in a digital project like Fresh Grocers. It means making sure that the system will not break any laws or regulations. In Sri Lanka, there

are certain laws related to digital payments, consumer protection, data privacy, and online business operations. If Fresh Grocers collects user data such as phone numbers and addresses, they must protect that data and inform users how it will be used.

If the system violates the laws or regulations, it can result in fines or even being shut down. For example, using someone else's copyrighted software without a license, or misusing customer data, could bring legal trouble. It's also important to look at tax laws when selling groceries online. If different tax rates apply to different types of items, the system must be able to calculate this correctly.

For these reasons, doing a legal feasibility study early helps avoid future problems. It may require consulting legal experts to understand the exact obligations. But it's better to invest in legal advice now than face bigger issues later.

Cultural Feasibility

Cultural feasibility means checking how well the new system fits with the organization's existing values, communication style, and the people's willingness to change. For Fresh Grocers, this involves not just the technical staff, but also shop managers, delivery agents, call center workers, and even customers.

Some employees may be afraid that automation or digital systems will replace their jobs. Others may be used to paper-based systems and may not want to use computers or dashboards. If there is strong resistance to change, the system may not be used even if it is perfectly built.

So understanding the company culture, the language used by the staff, and their comfort with technology is important. Maybe the interface needs to support Sinhala or Tamil in addition to English. Or perhaps training sessions are needed to help employees become familiar with the system.

Ignoring cultural feasibility could lead to a situation where the system is technically fine, but nobody wants to use it. That would be a total failure. So involving employees in early discussions and gathering their feedback helps improve cultural alignment.

Operational Feasibility

Operational feasibility refers to whether the system can actually work in the day-to-day activities of the business. Even if something is technically and legally possible, it may still not be operationally practical. In Fresh Grocers' case, it means checking if the web-based system can really handle the grocery orders, payments, deliveries, customer support, and other tasks.

For example, suppose the platform allows customers to choose delivery times. The company must ensure that enough delivery staff are available during those time slots. If a system allows something that can't be fulfilled operationally, it causes customer complaints and reduces trust in the brand.

It also includes checking if employees are ready to manage the system. If customer support is supposed to track orders using the backend, but the system is too complicated or buggy, then they will struggle. So operational feasibility must test ease of use, staff readiness, and process compatibility.

This step also helps plan for future growth. What happens when the number of customers grows? Can the system scale? If operational capacity is not studied now, the system may work fine for 100 users but fail completely at 1000 users.

One important thing to understand is that all these feasibility types are connected. A system that is technically feasible but too expensive is useless. A system that is economically feasible but illegal is dangerous. A system that works well operationally but is rejected by staff will not survive. So during the systems investigation, the team must look at all criteria together and balance them.

In the case of Fresh Grocers, suppose they decide to use a third-party delivery service to save operational costs. This might make economic sense, but from a legal point of view, they must ensure that customer data is not shared without permission. Similarly, it might be technically easier to develop in English only, but operationally, it may confuse many local users. These trade-offs must be studied properly to choose the best approach.

Feasibility Study in the Real Process

In a real-world project, feasibility study is usually done at the early stage of system development life cycle (SDLC), often during the planning or system investigation phase. It involves discussions with stakeholders, interviews with users, studying current systems, doing cost analysis, and legal review. It may not give a yes-or-no answer, but it helps in understanding risks, limits and opportunities. Sometimes, it may even suggest to not go ahead with the project if the risks are too high.

For Fresh Grocers, doing a proper feasibility study before writing even a single line of code is wise. It helps them decide if the project is worth doing, what changes are needed in the business process, and how to make the solution work smoothly. Skipping this step can result in a product that is either not usable, too expensive, or fails to meet business goals.

Critically evaluate the strengths and weaknesses of the traditional and agile systems methodologies and feasibility study in solving a specific business-related problem.

When developing a system to solve a business issue like what Fresh Grocers facing, it is important to pick the right development method and do good feasibility checking. Fresh Grocers is planning a online grocery ordering system, which need features like real-time delivery tracking, customer-friendly interface, and SMS updates. To decide how to build it right, we have to look critically at both traditional and agile methodologies and also how feasibility studies help the planning.

Traditional system methodologies like the Waterfall or Iterative Waterfall gives a very structured and step-by-step process. One big advantage is that it make everything planned clearly before the actual development start. For example, in a Waterfall model, all the requirements from Fresh Grocers would be gathered at first, then design, then coding, then testing and finally deployment. This makes it easier to control the whole process, especially if the project is big or includes multiple departments like inventory, finance, and logistics. It allows Fresh Grocers' management to see clear project timeline, budget plan, and documentation at each stage.

Another strength is the focus on documentation. Traditional methods make sure that every phase is recorded with proper documents like Software Requirement Specifications (SRS),

design diagrams, user manuals etc. This is very useful for a company like Fresh Grocers because maybe in the future they change the team or outsource part of the project, and new developers can understand the system better through documentation.

However, there are also serious weaknesses. First, traditional approaches are not flexible. Once the requirement phase is done and signed off, it is very hard to make changes. Imagine if halfway through the project Fresh Grocers realizes they need to add a mobile app version, or they want to include push notification alerts. Under Waterfall, that would mean going all the way back to the requirement or design phase, wasting time and money. Also, customer feedback doesn't come until the very end, after the whole system is developed. This can cause mismatch between what developers build and what users actually need.

In contrast, Agile methodologies like Scrum or Kanban are designed to be more flexible and responsive to changes. In Agile, the project is divided into small parts called "sprints" or "iterations." After every sprint (which can be 2-4 weeks), the development team presents a working feature to stakeholders for feedback. For Fresh Grocers, this is very useful. Let's say they release a first version where customers can only order groceries but not schedule a delivery time. If customer feedback says this is a problem, the development team can prioritize that feature in the next sprint and release it quickly.

Agile also encourages active involvement of the customer. In Scrum, for example, there is a "Product Owner" who represents Fresh Grocers and helps define what features are most important. This role makes sure that the system is always aligned with business needs. Also, the system starts to become usable earlier, since parts of it get released sooner. This can create early business value.

But Agile has some challenges too. First, it may not be easy for all organizations to adopt it. Fresh Grocers might not have people who understand Agile practices or who can take on roles like Product Owner or Scrum Master. Also, Agile depends on close collaboration, so if team members are not used to working this way or are from different departments that don't communicate well, it can slow down progress. Also, Agile projects can sometimes lose focus if priorities keep changing too often.

Besides development methods, feasibility study plays a huge role in making sure that the project starts on a solid base. A good feasibility study checks several key areas: technical, economic, legal, operational and cultural. For Fresh Grocers, let's look at each.

Technical feasibility checks whether the project is possible with the current technology the company has. For example, does Fresh Grocers already have a website or server system that the new platform can integrate with? Do they have staff who know how to maintain web and mobile systems, or will they need to hire new people? A system that needs complex GPS-based delivery tracking might not be possible if the company only has a basic tech setup.

Economic feasibility looks at whether the project makes financial sense. This includes development cost, training, software licenses, hosting charges etc. For a small business, a high-cost solution with long ROI may not be feasible. A feasibility report will help Fresh Grocers decide if they can afford the platform or need to scale it down.

Legal feasibility is very important when dealing with customer data. If Fresh Grocers plans to collect personal info like addresses or use cookies, they need to follow data protection laws. A feasibility study will point out legal risks and help them stay compliant. For example, sending SMS alerts might require opt-in from users due to telecom regulations.

Operational feasibility checks if the system will actually work with how the company runs daily. It looks at whether employees can use the system, whether it improves current workflow or adds confusion. If the platform is too complex, or staff are not trained, it may fail even if technically perfect.

Cultural feasibility is often ignored but very important. This checks whether staff and management are open to change. If Fresh Grocers has a traditional paper-based system, switching to digital orders may face resistance. A good feasibility study identifies these problems and suggests how to overcome them like training, change management etc.

If Fresh Grocers chooses a traditional method, the feasibility study becomes even more critical. Since changes are hard to make later, the feasibility report must get everything right at the start technical scope, budget, legal issues, etc. A mistake in feasibility might mean the whole project becomes unworkable mid-way.

With Agile, the impact of feasibility is different. Since Agile allows gradual development, some technical or operational issues can be discovered and fixed along the way. But still, a baseline feasibility check is needed to make sure the company isn't walking into something that's too expensive, illegal, or impractical.

One weakness of feasibility studies is they depend on assumptions. For example, economic feasibility might assume certain number of users or expected profits, but in real life, these numbers could be off. Also, market trends or competitor moves are not always predictable. If Fresh Grocers doesn't account for fast-changing customer behavior, the feasibility results may become outdated.

Another issue is that feasibility studies may ignore some "soft" factors. For example, customer satisfaction, employee morale, or market reputation might not be measured well. But these can affect the success of the platform a lot.

Both traditional and Agile methodologies have strengths and weaknesses. Traditional methods are great when things are predictable and need tight control. Agile is better when flexibility and customer feedback are important. The choice depends on the business context. For Fresh Grocers' case, where they are building a new customer-facing web-based platform, Agile seems more suitable because it allows faster delivery, feedback loops, and improvements.

But whichever method is used, a detailed feasibility study is key to success. It gives the management a clear picture of what's possible, what's risky, and how the plan can be adjusted. Even if Agile is flexible, starting without knowing technical or legal limits can cause failure later. And for traditional projects, the feasibility report is almost like a roadmap if it's wrong, the whole project might crash. The Fresh Grocers web system was developed to replace the old manual order-taking process with a modern, easy-to-use online platform for grocery ordering and delivery. To make this transformation successful, two important tools were used: systems analysis methodologies and a detailed feasibility study. Both helped guide the project in the right direction and reduce the risk of failure.

At the start, traditional methods like the Waterfall model were helpful. They provided a clear structure, with step-by-step planning and documentation. This was useful for organizing the early stages of the project and understanding the company's basic needs, especially since Fresh Grocers was shifting from a manual system. However, Waterfall became less effective when

customer needs started changing. For example, customers began asking for real-time delivery updates and more digital payment options. Waterfall was too rigid to handle these fast changes.

Agile, especially the Scrum method, worked better for this type of project. Agile allowed the team to build the system in smaller steps (called sprints), starting with key features like customer accounts, delivery tracking, and SMS alerts. These features were released early and improved over time based on user feedback. This was perfect for the fast-moving grocery delivery industry, where customer satisfaction is a top priority.

The feasibility study also played a big role. It checked if Fresh Grocers had the right technology, money, people, and legal compliance to build and run the system. For instance, the legal check made sure the system followed Sri Lankan ICT and customer laws. The operational review showed that current staff could handle online orders with a bit of training. The technical study confirmed that the company had the internet tools and software needed, including support for SMS and map-based delivery tracking.

Overall, combining Agile with a good feasibility study helped Fresh Grocers successfully move into the digital world.

So, while both systems and feasibility studies have their limits, using them wisely together is the best way to solve business problems like what Fresh Grocers is trying to do.

Activity 3 – Requirements

Requirements

The requirements of a software project are the functions, features, and limitations that the final product must meet. In other words, the requirements specify what the program should perform, how it should appear, and any criteria that must be satisfied for it to be regarded successful. Gathering requirements is necessary in order to build a product that fits the demands of the consumer or client. (visuresolutions, 2025)

Types of requirements

There are two types of requirements

1) System Requirements

System requirements may be thought of as an enlarged form of user needs. System requirements are the starting point for each new system design. These criteria provide a clear explanation of the user needs that the system must meet.

Functional requirements

As the name implies, functional requirements explain the functionalities of the system to be created. It is a description of what the system will be and how it will work to meet the demands of the users. They give a detailed description of how the system is expected to respond to a certain command, as well as the features and what the users may anticipate.

Nonfunctional requirements

Non-Functional Requirements describe the system's restrictions and constraints. These requirements have no bearing on the application's operation.

2) User requirements

A user requirement is made up of both functional and non-functional needs. These user criteria must be constructed in such a way that people with no technical understanding may readily grasp them. As a result, they must be written in simple tables, forms, and diagrams in normal

language. Also, ensure that the paper contains no information about system architecture, software, or formal notations.

Requirements for Fresh Grocers Web-based System

1. User Requirements

These describe what users expect the system to do

- ❖ Customers should be able to register and log in to their account.
- ❖ Users must be able to browse products by category (fruits, vegetables, dairy, etc.).
- ❖ Customers should be able to add items to a shopping cart and modify quantities.
- ❖ The system should allow users to place orders and choose delivery dates/times.
- ❖ Customers must receive SMS/email confirmation after placing an order.
- ❖ Users should be able to track their delivery status in real time.
- ❖ Customers should view their past orders and reorder items quickly.
- ❖ Admins should be able to add, update, or remove products from the system.
- ❖ Store staff should be able to view and process customer orders.
- ❖ Customers should be able to contact support or submit complaints/feedback.

2. Functional System Requirements

These are detailed technical specifications that define how the system should behave.

- ❖ The system shall allow user account creation with email verification.
- ❖ The system shall authenticate users through login credentials.
- ❖ The system shall enable product search by name, category, and price range.
- ❖ The system shall support adding/removing items from the cart.
- ❖ The system shall calculate the total price including tax and delivery fees.
- ❖ The system shall integrate with a payment gateway for secure payments.
- ❖ The system shall store delivery address and preferences in user profiles.
- ❖ The system shall generate an invoice for each order.
- ❖ The system shall allow admins to manage inventory (add, update, delete products).
- ❖ The system shall provide a dashboard for order tracking and status updates.
- ❖ The system shall allow staff to assign delivery personnel and update delivery status.

3. Non-Functional System Requirements

These define how the system should operate rather than what it should do.

- ❖ **Performance:** The system should support at least 200 concurrent users with no more than 3 seconds of page load time.
- ❖ **Scalability:** The system must be scalable to handle increased traffic during promotions or holidays.
- ❖ **Availability:** The system shall be available 99.9% of the time, with maintenance windows outside peak hours.
- ❖ **Usability:** The platform must be user-friendly, accessible via both desktop and mobile devices.
- ❖ **Reliability:** The system should ensure no order is lost during checkout due to server failures.
- ❖ **Security:** The system shall comply with data protection laws (e.g., GDPR) and ensure customer data is protected.
- ❖ **Maintainability:** The system shall be modular, with easily upgradable components and clear documentation for maintenance.
- ❖ **Compatibility:** The system should be compatible with major browsers (Chrome, Firefox, Safari, Edge).

System Requirements for Fresh Grocers Web-based System

Hardware Requirements

- ❖ Server hardware with sufficient processing power and memory to support the system's operations.
- ❖ Backup storage devices and systems to ensure data integrity and recovery.
- ❖ Client workstations or devices (desktops, laptops, mobile devices) capable of running the system's frontend application.

Software Requirements

- ❖ **Operating System:** The system must be compatible with modern operating systems (e.g., Windows, Linux) depending on the chosen infrastructure.

- ❖ Database Management System (DBMS): The system must utilize a relational database management system (RDBMS) like MySQL, PostgreSQL, or SQL Server to store client and transaction data.
- ❖ Web Server (if web-based): Apache, Nginx, or a similar web server to host the application.
- ❖ Frontend Framework: A modern frontend framework such as React, Angular, or Vue.js (for web-based applications).
- ❖ Backend Framework: A suitable backend framework like Node.js, Django, or Ruby on Rails, depending on the chosen architecture.
- ❖ Security Protocols: SSL/TLS encryption for secure data transmission, multi-factor authentication (MFA) for access control, and data encryption for storage.

Network Requirements

- ❖ Internet Connectivity: The system requires a stable internet connection to ensure seamless communication between users, drivers, and the server.
- ❖ Mobile Network Connectivity: Since customers receive SMS notifications, the system should be capable of sending SMS messages over the mobile network.

Security Requirements

- ❖ Encryption: Data transmission between clients and the server should be encrypted using protocols like HTTPS to ensure security and privacy.
- ❖ Authentication and Authorization: The system should implement secure login mechanisms for customers, drivers, and the phone operator, with proper authentication and authorization controls.

Feasibility techniques used to identify the requirement

Data collection is the interaction between the software developer (in this example, a business analyst) and the clients (including users). Data collection approaches include interviews, meetings, observations, surveys, content analysis and focus groups. (saylor, 2025)

✓ Questionnaires and surveys

A survey entails asking a series of structured questions to a sample of persons or organizations. Questionnaires are written collections of questions that responders must complete. These techniques are frequently used to collect self-reported data on attitudes, preferences, and habits.

Surveys, as well as questionnaires, stand out as excellent methods of data collection from a quantitative perspective. They set a standard for gathering data about the attitudinal, liking, and behavioral reactions and responses. These weapons by-passing the limitations of a small-scale survey and generate information from variegated demographics. The findings of the investigation give you the information about consumers' positions and points of view.

	Survey	Questionnaire
Individual questions	✓	✓
Used for gathering data	✓	✗
Analyzes responses	✓	✗
Provides insights about an individual	✓	✓
Provides insights about a group of people	✓	✗

✓ Interviews

Interviews are direct, face-to-face, or remote discussions between a researcher and a respondent. They can be organized (with specific questions) or unstructured (with more open-ended conversations). Interviews are useful for in-depth topic investigation.

Through interviews broad opportunities for deep study of an issue or the topic is provided. Both forms of interviewing, i.e. structured and unstructured, provide researchers with an opportunity to directly interact with respondents, inquire and seek deeper understanding, as well as verify what has been said. This approach is needed because of the quality of qualitative data gathered where the respondent's reasons for the behavior, motives, and underlying factors are revealed.

TYPES OF INTERVIEW



✓ Observation

Observation is the systematic observation and recording of events, behaviors, or processes without directly interacting with the individuals. It is used to gather information on naturally occurring behaviors or occurrences.

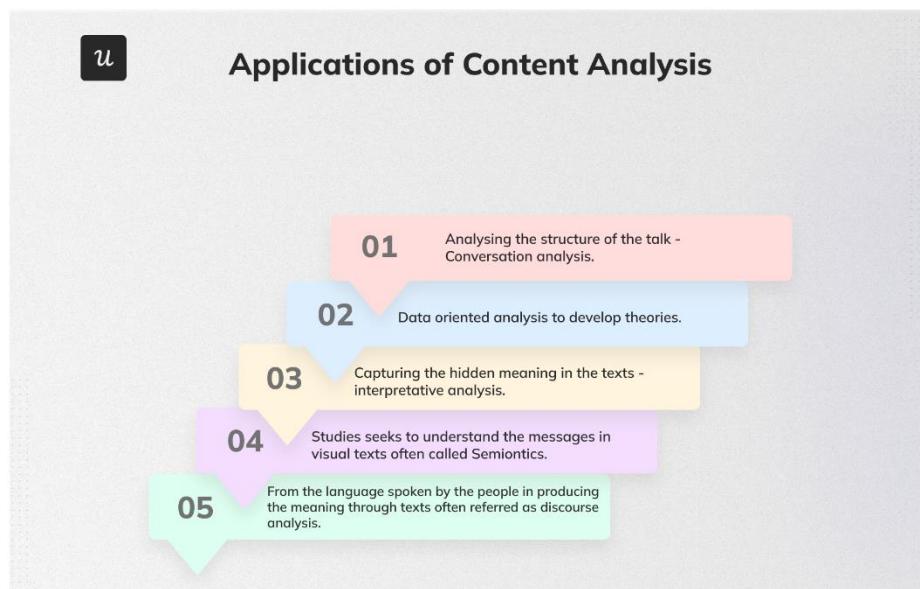
An observation provides researchers with access to real life setting in which research team can collect data on behavior, event, or process and remain unseen in the process. It enables one-to-one, unfiltered data collection because of the fact that the information is collected in the real world. Observation turns out to be very suitable for recording non-verbal signals, behavioral patterns, and contextual components which can only be monitored by other methods.



✓ Content analysis

Content analysis is the process of analyzing textual, visual, or auditory content, such as texts, photos, videos, or social media postings. To get insights, researchers categorize and evaluate the information.

Content analysis is the key tool in cultural research to find trends, patterns, and themes in the visual image, textual narrative or audiobook. The ability to do a systematic analysis of large volumes of data is conveyed with an example of texts, photos, videos, or social media postings to embrace insights on or patterns of the key issue. With content analysis, researchers are able to comprehend the surreptitious tongue, sentiment, or meanings within the text. It gives the unique perspective of public opinion, media portrayal, or communication strategies.



✓ Focus Groups

A focus group is a moderated discussion in which a small group of participants share their thoughts and experiences on a specific topic. This method is frequently used in market research and product creation.

The use of focus groups yields participation in group discussions among a limited number of individuals, and it provides researchers with an opportunity to explore and to explore peoples' opinions, attitudes and experiences together. It allows peers to interact, think up ideas, and to have a chance to exchange information and opinions, providing valuable data on one issue or matter, that is of a qualitative nature. Focus groups are particularly helpful in this case, for

identifying ground for the formation of the same opinion, or a disagreement, and in the communication of the most advanced ideas and solutions through the interaction in groups.



✓ **Prototyping**

Developing prototypes of the web-based reservation system and conducting usability testing sessions with stakeholders can help assess the feasibility of the proposed features. Prototyping allows stakeholders to interact with a tangible representation of the system, providing valuable feedback on its usability and functionality.

✓ **Market Analysis**

Conducting market analysis to assess the demand for web-based taxi reservation services in Sri Lanka can help determine the viability of the project. This involves researching market trends, competitor offerings, and customer preferences to identify potential opportunities and challenges.

✓ **Cost-Benefit Analysis**

Conducting a cost-benefit analysis to evaluate the financial feasibility of developing and implementing the web-based reservation system. This involves estimating the costs associated with system development, maintenance, and operation, and comparing them against the expected benefits and potential revenue generation opportunities.

✓ **Risk Assessment**

Identifying and assessing potential risks and challenges associated with the project, such as technological risks, regulatory compliance issues, and market competition. This helps in developing risk mitigation strategies to address potential obstacles and ensure project success.

Advantage and Disadvantage of Data collection techniques

Technique	Advantage	Disadvantage	Example
Questionnaires and surveys	<ul style="list-style-type: none"> • Surveys may be distributed to a big number of people at the same time, making them a cost-effective strategy. • The questions are standardized, which reduces interviewer bias. • Respondents may feel better at ease responding to difficult questions anonymously. 	<ul style="list-style-type: none"> • Surveys may not capture subtleties in replies or specific information. • Participants may give socially acceptable replies or misinterpret questions. • Obtaining a sufficient quantity of responses might be difficult. 	Customer satisfaction surveys, as well as political opinion polls.
Interviews	<ul style="list-style-type: none"> • Researchers can go extensively into issues and ask follow-up inquiries. • Interviews can be modified based on the responses of respondents. • In a one-on-one interview, participants may feel more involved and open up. 	<ul style="list-style-type: none"> • Interviews may be time-consuming and expensive. • Unintentionally, interviewers can sway replies. • The findings may be difficult to generalize to a wider group. 	Job interviews, qualitative social science study.
Observation	<ul style="list-style-type: none"> • Self-reporting bias is less prevalent in observations. • Researchers can observe behavior in its natural setting. 	<ul style="list-style-type: none"> • Observers may not understand why a behavior occurs. 	Investigating classroom dynamics and observing

	<ul style="list-style-type: none"> Nonverbal indicators and actions might be observed. 	<p>Observations need both time and effort.</p> <p>When people are aware that they are being watched, they may change their conduct.</p>	consumer behavior at a retail business.
Content analysis	<ul style="list-style-type: none"> Capable of analyzing enormous amounts of text or media data. Reduces the amount of subjectivity in data analysis. Content modifications may be tracked over time. 	<p>When evaluating isolated material, context may be lost.</p> <p>Large datasets might be time-consuming to analyze.</p> <p>It can be difficult to get consensus among developers.</p>	Examining social media posts for trends and mood in news stories.
Focus Groups	<ul style="list-style-type: none"> Insights are gathered from a diverse variety of people. Participants can build on one another's ideas. Researchers can study how people react to stimuli in real time. 	<p>Dominant participants may sway the ideas of others.</p> <p>The findings may not apply to a bigger population.</p> <p>The moderator's actions can have an influence on the debate.</p>	Assessing reactions to a new product proposition, obtaining input on a marketing effort.

Techniques Used to Review the Requirements of Fresh Grocers Web-Based System

To gather the requirements for the Fresh Grocers online grocery system, we used mainly two techniques: questionnaires and interviews. These methods helped us collect useful information from different types of users, like customers, delivery workers, and staff who work inside the company. Using these techniques was important to understand what people really want from the system.

The questionnaire was created with Google Forms and sent to many users. It had questions about important features like how customers want to register, how they want to place orders, how they would track their deliveries, and what kind of notifications they want. Also, the questionnaire asked about rating the delivery agents. The questionnaire gave us a lot of answers quickly and helped us find out what most users expect. Because it was easy to share online, many people could respond from different places. However, some users might not answer carefully or may skip some questions. Sometimes, people only give short answers without explaining their real problems.

Besides the questionnaire, we did interviews with some internal workers, such as customer service team, logistics coordinators, and IT staff. The interviews were done one by one or in small groups. They helped us get more detailed information about how the current system works and what problems they have in their daily jobs. In interviews, people could explain their thoughts better and talk about things that are hard to ask in a questionnaire, like concerns about data privacy or how to track delivery agents' locations. But interviews take more time and we could only talk to a few people, so the information may not cover all staff opinions.

Using both questionnaires and interviews was good because they balanced each other. Questionnaires gave us numbers and facts that show common needs, while interviews gave us more stories and ideas that we could not get from forms. Together, they gave us a better picture of what the system should do and how it should look.

Other ways to gather requirements like observing workers or having group workshops were thought about but were not used because they were difficult to organize or cost more time. For example, watching delivery agents all day would take a lot of effort and could disturb their work. Workshops need many people at the same time and this was hard to arrange.

This way of reviewing requirements helped Fresh Grocers understand both what customers want and how the company works behind the scenes. This is very important because the system should be easy for customers and also help staff to do their jobs better. Other methods such as direct observation, focus groups, workshops, and prototyping were also considered. Observation would mean following delivery agents or staff during their work to see problems firsthand. While this might provide useful data, it is costly, time-consuming, and intrusive to workers' schedules.

Focus groups and workshops are good for brainstorming and discussing ideas with multiple stakeholders at once. However, organizing these was difficult due to the different locations and availability of participants. Prototyping sessions, where users test a simple version of the system to give feedback, were planned for later stages after the initial requirements review.

Some problems with the techniques used include the risk that people might not give honest answers or might not understand the questions well. To reduce these problems, we tried to make the questions clear and sent the questionnaire to many different users. Also, interviews were carefully planned to make people comfortable and open to share. The techniques used have some challenges. Questionnaires can suffer from low response rates or superficial answers. To reduce this, the team sent the questionnaire to a wide audience and tried to keep questions simple and clear. Interviews risk being subjective and limited in scope, so multiple staff from different departments were interviewed to get a wider perspective.

Another challenge is managing conflicting requirements. For example, customers may want many features and quick delivery, but delivery workers need realistic schedules and manageable workloads. Balancing these needs requires careful analysis and sometimes trade-offs in system design. The thorough review of requirements using these techniques helped reduce the risk of building a system that does not meet users' needs. It also helped the team identify key features to prioritize, such as real-time delivery tracking and rating systems for delivery personnel, which improve both customer experience and operational control.

Collecting diverse inputs ensured the system would support Fresh Grocers' business goals like increasing customer satisfaction and improving delivery efficiency. This also helps avoid costly redesigns or fixes after the system is launched. Using questionnaires and interviews was a suitable method for reviewing the requirements of Fresh Grocers web-based system. These techniques helped gather both broad and deep information which will help to build a system that meets the needs of users and supports the business well.

Assessing the Effectiveness of SCRUM Methodology for Fresh Grocers Web-Based System

Choosing the right methodology is important to develop a system that truly meets business needs and can adapt to changes efficiently. For Fresh Grocers, a web-based grocery ordering platform, the SCRUM methodology is a popular and practical choice. This is because SCRUM offers flexibility, encourages continuous user involvement, and supports iterative development. However, while SCRUM has many benefits, it is also important to critically assess how effective it really is in this particular business context and consider any challenges it might bring.

Why SCRUM is Suitable for Fresh Grocers

First, the grocery delivery business is very dynamic. Customer preferences can change quickly—people may want new features like scheduled deliveries, special payment options, or improved order tracking. Demand can also vary with seasons or promotions. SCRUM's iterative process, which breaks work into short sprints of two to four weeks, allows the development team to deliver parts of the system frequently and get feedback from users early and often. This means the Fresh Grocers platform can be improved continuously, rather than waiting until the end of a long project when changes might be too expensive or difficult.

SCRUM encourages close collaboration between developers, business stakeholders, and users. Roles like the Product Owner make sure that user needs and business priorities are always clear, while the Scrum Master helps the team stay focused and remove obstacles. Daily stand-ups and sprint reviews improve communication, so problems are identified quickly and can be solved before they cause bigger delays. For Fresh Grocers, which has many different users—customers, delivery agents, and internal staff—this ongoing communication is very important to ensure the system meets everyone's needs.

Another important benefit of SCRUM is that it empowers the development team to be self-organizing. Team members take responsibility for planning their work and solving problems creatively. This can lead to faster issue resolution and more innovative solutions, which is valuable in a competitive market like online grocery delivery.

Sprint retrospectives at the end of each sprint provide a chance to reflect on what went well and what could be improved. This helps the team become more efficient and effective over

time. For Fresh Grocers, this means the platform can evolve based on real user feedback, improving customer satisfaction and business performance with every release.

Critical Assessment of SCRUM's Effectiveness

While SCRUM offers many advantages, it is not without challenges. One issue is that SCRUM requires high levels of discipline and experience from the team. Since the team is self-organizing, if members lack experience or are not committed, the project may suffer from poor planning, missed deadlines, or unclear priorities. Fresh Grocers may need to invest in training and coaching to make sure the SCRUM team works well. Another challenge is the dependency on active involvement from business stakeholders, especially the Product Owner. The Product Owner must be available to clarify requirements and make decisions quickly. If Fresh Grocers' management or users are busy or not engaged enough, this can slow down progress or lead to misunderstandings about what the system should do.

SCRUM works best when requirements are expected to change frequently. However, if some parts of Fresh Grocers' system are very stable and clearly defined (for example, payment processing or legal compliance), SCRUM's iterative approach might introduce unnecessary overhead. In such cases, a more traditional or hybrid methodology might be more efficient.

Moreover, SCRUM focuses on delivering working software but does not prescribe detailed documentation. For Fresh Grocers, which may need to comply with legal regulations on data privacy or must have clear operational procedures, the lack of thorough documentation early on can be a risk. SCRUM also relies heavily on effective communication and team collaboration. For Fresh Grocers, if the development team is distributed across locations or includes external contractors, this can cause coordination issues. Time zone differences, cultural differences, and remote work can reduce the benefits of face-to-face communication that SCRUM promotes.

Comparison With Other Methodologies

Compared to traditional Waterfall methodology, SCRUM is more flexible and better suited to handling changing requirements. Waterfall's sequential phases could cause delays if Fresh Grocers needs to add new features based on market trends or customer feedback. However, Waterfall might be simpler to manage if the project scope is fixed and well understood from the start.

Another alternative is the Kanban methodology, which focuses on continuous delivery and limiting work in progress. Kanban might offer more flexibility for Fresh Grocers if priorities shift constantly, but it lacks SCRUM's structured sprint cycles and formal roles, which may reduce accountability. Hybrid approaches, combining SCRUM with some traditional elements, could also work well for Fresh Grocers using SCRUM for development but Waterfall for legal compliance and testing phases. This would balance flexibility with control.

Impact on Project Success

SCRUM's biggest strength for Fresh Grocers is its customer-focused, adaptive approach. The frequent delivery of functional features means users can start using the system early, and feedback can shape future development. This reduces the risk of building a system that does not meet real needs and helps avoid costly rework. However, if SCRUM is not implemented well for example, if the team is inexperienced or stakeholders are not engaged it could lead to confusion, missed deadlines, and lower quality. This would hurt Fresh Grocers' reputation and could delay their digital transformation.

To maximize SCRUM's effectiveness, Fresh Grocers should invest in training the team, appoint a dedicated and empowered Product Owner, and maintain strong communication channels among all parties. Also, some upfront planning to identify stable system requirements and compliance needs will help avoid surprises during sprints. Overall, SCRUM is an effective methodology for the Fresh Grocers web-based system because it supports iterative development, quick response to change, and continuous user involvement. It fits well with the fast-moving grocery delivery industry and allows the business to stay competitive by delivering valuable features regularly.

At the same time, SCRUM requires discipline, experience, and active stakeholder engagement to work well. Fresh Grocers must be aware of these challenges and put in place measures to address them. With careful management, SCRUM can deliver a user-centered, flexible, and successful online grocery platform that meets both customer needs and internal business goals.

Justification for Selecting Methodology agile SCRUM for Fresh Grocers Web-Based Platform

Implementing the SCRUM for the Fresh Grocers web-based platform is a very suitable and effective choice considering the specific needs and challenges of this project. The grocery delivery business is fast-moving and constantly changing, so the development process must be flexible and able to adapt to shifting user expectations and market conditions. SCRUM provides a clear way to manage this complexity through its iterative, incremental, and collaborative approach, making it an ideal methodology to support the development of the Fresh Grocers system.

Handling Evolving and Complex Requirements

One of the biggest reasons for selecting SCRUM is its ability to handle evolving requirements efficiently. Unlike traditional development methods that follow a linear process where all features are planned in detail before any coding starts SCRUM allows the Fresh Grocers development team to break the work down into smaller, manageable chunks called sprints. Each sprint typically lasts 2 to 4 weeks and results in a working piece of software with specific features.

For Fresh Grocers, this means the team can first focus on the most important and foundational elements such as user registration, browsing the product catalog, and managing the shopping cart. Once these core features are working well and tested, later sprints can add more advanced functionality such as real-time delivery tracking, personalized promotions, or loyalty programs. This phased approach reduces risk because the system is usable earlier and does not depend on all features being ready at the same time.

Moreover, grocery delivery involves many changing factors like seasonal promotions, supply chain disruptions, or new payment options which means the requirements can change during development. SCRUM's iterative cycles allow Fresh Grocers to adjust the product backlog based on ongoing customer feedback and business priorities, rather than being locked into an outdated plan. For example, if customers demand contactless delivery or faster checkout options after the initial release, these can be incorporated into upcoming sprints without major disruption.

Strong Stakeholder Involvement and Collaboration

Another key reason why SCRUM fits Fresh Grocers well is its focus on collaboration and communication among all stakeholders. The grocery delivery service involves many different groups who each have unique needs and perspectives: end users (customers ordering groceries), warehouse staff managing inventory, delivery drivers handling orders, customer support teams responding to queries, and management overseeing operations and business goals.

SCRUM encourages regular interaction between these groups and the development team through events like daily stand-ups, sprint planning, sprint reviews, and sprint retrospectives. This means all parties stay informed and engaged throughout the development process. For instance, during sprint planning, warehouse managers and customer service representatives can help prioritize features that will improve their workflows or customer satisfaction. During sprint reviews, stakeholders can test new features, provide feedback, and suggest improvements before the next sprint starts.

This continuous involvement helps prevent misunderstandings about what the system should do, ensures that the platform truly meets operational needs, and builds a sense of shared ownership over the project. It also boosts transparency, so everyone knows the project's progress, challenges, and next steps. This collaborative culture is very important for Fresh Grocers because the platform must balance customer convenience, delivery efficiency, and backend operations, all of which depend on smooth teamwork.

Encouraging Continuous Improvement

SCRUM's emphasis on continuous improvement through sprint retrospectives is another strong advantage for Fresh Grocers. After each sprint, the team reflects on what went well and what did not, discussing ways to improve processes and the product itself. This could mean fixing bugs faster, enhancing the user interface, or optimizing the order processing flow.

For a customer-facing digital platform like Fresh Grocers, maintaining high usability, reliability, and performance is critical to keep users satisfied and loyal. Because SCRUM supports frequent releases of functional software, feedback can be collected and acted upon quickly. For example, if users report that the product search is slow or inaccurate, the development team can prioritize fixing this issue in the next sprint rather than waiting for a major update months later.

This iterative cycle of development, feedback, and improvement helps Fresh Grocers stay competitive in a market where customer expectations evolve rapidly. It also allows the team to

be more innovative trying out new ideas or features in small, low-risk increments before fully committing to them.

Flexibility and Responsiveness to Market Needs

The grocery delivery industry is highly competitive and influenced by trends such as convenience shopping, mobile app usage, and sustainable delivery options. SCRUM's flexibility allows Fresh Grocers to respond quickly to these changing market demands.

For example, if competitors introduce new payment gateways or delivery time slots that customers like, Fresh Grocers can add similar features more quickly by adjusting their sprint backlog. This adaptability is critical because delays in responding to market changes can result in lost customers.

Additionally, Fresh Grocers may discover new regulatory or data security requirements as the platform grows. SCRUM allows the development team to incorporate these compliance needs iteratively rather than trying to build everything upfront, reducing compliance risks while maintaining agility.

Supporting a Self-Organizing and Motivated Team

SCRUM promotes self-organizing teams where members take ownership of their work, plan their tasks, and collaborate to solve problems. This empowers the Fresh Grocers development team to be proactive, responsible, and creative in delivering features and resolving issues.

Such a mindset can increase motivation and job satisfaction, leading to better productivity and innovation. When team members feel trusted and valued, they are more likely to go beyond the minimum requirements to improve the system. This can translate into a better user experience for Fresh Grocers customers and more efficient operations behind the scenes.

Consideration of Limitations and Risks

While SCRUM has many benefits, it is not without challenges. One potential risk is that SCRUM requires experienced team members who understand agile principles and can work effectively in a self-managed environment. If the Fresh Grocers team is new to SCRUM, they might struggle initially with planning, estimating, or prioritizing work, which could lead to missed deadlines or scope creep. Additionally, SCRUM depends heavily on the Product Owner to represent stakeholder interests and make timely decisions. If Fresh Grocers cannot dedicate

a knowledgeable and engaged Product Owner, the project may suffer from unclear priorities or delays in resolving questions.

SCRUM also assumes close collaboration and frequent communication, which may be difficult if team members or stakeholders are geographically dispersed or have conflicting schedules. For Fresh Grocers, who may work with external vendors or remote teams, these challenges must be managed carefully.

Furthermore, SCRUM focuses on working software over documentation, which can be a problem for parts of the Fresh Grocers platform requiring detailed compliance records or maintenance guides. This means additional efforts might be needed to ensure proper documentation alongside the SCRUM process. Despite these limitations, the flexibility, customer focus, and rapid feedback cycles of SCRUM generally outweigh the risks for Fresh Grocers. The methodology's benefits in managing changing requirements, improving team collaboration, and delivering value early are critical for success in this competitive, evolving market.

Comparison With Other Methodologies

Compared to traditional Waterfall methodology, SCRUM is much more adaptable. Waterfall's sequential phases can delay the delivery of usable software, which is risky if Fresh Grocers needs to react quickly to market trends or user feedback. While Waterfall offers strong documentation and clear milestones, it can be rigid and slow for dynamic projects like Fresh Grocers. Another option could be Kanban, which emphasizes continuous workflow without fixed sprints. Kanban might offer even more flexibility in prioritizing tasks but lacks SCRUM's structured roles and ceremonies that promote accountability and regular reflection. For Fresh Grocers, the clear framework of SCRUM can help manage complexity better.

Some organizations use hybrid approaches that combine SCRUM with traditional practices for compliance or testing phases. This can suit Fresh Grocers well by balancing agility in development with control where needed.

SCRUM provides a structured yet adaptable framework that matches the needs of Fresh Grocers' web-based grocery ordering system. Its iterative development cycles, strong stakeholder engagement, and focus on continuous improvement allow the project to manage complexity, changing requirements, and multiple user groups effectively.

While SCRUM demands discipline, experience, and commitment from the team and stakeholders, its benefits in delivering value early and often, improving collaboration, and staying responsive to market changes make it the best choice for Fresh Grocers.

By adopting SCRUM, Fresh Grocers can build a user-friendly, reliable, and scalable platform that evolves with customer needs and business goals, supporting the company's growth and success in the competitive grocery delivery market. Scrum was chosen as the most suitable development for the Fresh Grocers web-based system because it offers flexibility, fast delivery, and strong focus on customer needs. These qualities matched the nature of the Fresh Grocers business, which operates in a fast-moving retail environment where customer preferences and market trends can change quickly.

In the early planning stages, the team noticed customers wanted basic features like product browsing and SMS notifications. However, while development was ongoing, new needs came up, such as repeat ordering and real-time delivery tracking. If the team had used a traditional method like Waterfall, it would have been difficult and expensive to make these changes once development had already started. Waterfall follows a fixed plan, so new ideas often cause delays or require going back to earlier steps. In contrast, Scrum is designed to handle this kind of situation. It works in short development cycles called "sprints." After each sprint, the team can review progress, gather feedback, and make changes before starting the next cycle. This helped the Fresh Grocers team to respond quickly to customer feedback and adjust the project as needed.

Another reason Scrum was a good fit is because it encourages releasing parts of the system early. For example, the Fresh Grocers team finished user login and staff-managed order entry features in the first few sprints. These were shared internally for testing and early feedback, allowing the company to check if the direction was correct before continuing with more features. This reduced the chance of building the wrong solution.

Scrum also supports strong communication between business stakeholders and developers. In this project, a Product Owner worked closely with the Scrum team, representing Fresh Grocers and making sure business goals were always in focus. Regular meetings such as sprint planning and sprint reviews helped the team stay on track and respond to any issues quickly. This kind of close collaboration is harder to achieve in Waterfall, where most communication happens through long documents and fewer meetings.

Scrum helped Fresh Grocers manage their project in a flexible and efficient way. It allowed for changes, quick delivery of useful features, and close teamwork, all of which are important for succeeding in the competitive grocery delivery market in Sri Lanka.

Activity 4 – Design documentataion

Design specification for Fresh Grocers Ordering System

Table of content

1. Introduction
2. System Architecture
3. Design Specification
4. Use Case Diagram
5. Entity Relationship Diagram
6. Flow Chart
7. Data Flow Diagram
8. Wireframes
9. Instralation Guid

Introduction

The Design Specification document serves as a foundational guide for the development of the Fresh Grocers Web-Based Ordering Platform. Its primary purpose is to provide a clear and organized visual representation of the system's overall structure, including the layout of user interfaces, data flow management, and system functionality. This document outlines the essential technical and functional design requirements needed to build a reliable, user-friendly, and scalable application that aligns with the goals outlined in the system requirement analysis. By detailing the design standards, system components, and their interactions, the Design Specification helps ensure that the platform is developed in line with industry best practices. It also provides a framework that promotes clarity, consistency, and maintainability throughout the development process, ensuring that all stakeholders from developers to testers have a shared understanding of how the system is intended to work. This ensures that the final product will meet user expectations while being secure and easy to operate.

System Architecture

Software architecture

The software architecture of the Fresh Grocers Web-Based System follows a client-server model and is structured using a three-tier architecture. On the client side, customers are provided with a user-friendly web interface that allows them to browse available grocery items, add products to their cart, and place orders online. This platform acts as the main touchpoint for customers to interact with the system, enabling them to manage their orders, track deliveries, and access product information conveniently from any device with internet access.

Once an order is placed, the information is transmitted securely over the internet to a centralized server. This server acts as the middleware, bridging the communication between the customer-facing interface and the backend systems. It processes user inputs, validates orders, and coordinates the overall workflow of the platform. The server also handles user authentication, manages session data, and ensures smooth operation of all system features.

The third layer consists of the backend database, which stores essential data such as customer profiles, product inventories, order histories, and delivery schedules. This database ensures data consistency, accuracy, and security, supporting fast retrieval and updates as required by

the system. The architecture is modular and distributed, which supports scalability, maintainability, and reliability, making it easier to introduce future enhancements or handle increased user demand.

By using modern web technologies and secure communication protocols, the Fresh Grocers system provides a seamless and efficient online ordering experience while maintaining data integrity and system performance. This architecture supports the integration of advanced features like delivery tracking, agent assignment, and personalized recommendations, helping the business meet evolving customer expectations.

Hardware architecture

The hardware architecture of the Fresh Grocers Web-Based System is designed to support high performance, scalability, and availability for a seamless online shopping experience. The architecture is built around a combination of web servers, database servers, networking equipment, and user-end devices. These components work together to ensure smooth data flow, system reliability, and efficient handling of customer interactions.

At the core of the system are the web servers, which host the main application and serve web pages to users accessing the site. These servers are equipped with high-performance processors, sufficient memory, and fast storage to manage multiple user requests simultaneously. They are responsible for rendering the user interface, processing user inputs, and managing real-time interactions such as browsing, ordering, and checking out. Load balancers are often employed to distribute traffic evenly across multiple servers, ensuring optimal performance and system uptime during high-demand periods.

Supporting the web servers are the database servers, which store all critical information such as product listings, customer profiles, order details, and delivery records. These servers run a robust Database Management System (DBMS) designed to handle large volumes of data transactions with accuracy and speed. Redundancy and backup systems are implemented to prevent data loss and ensure recovery in the event of system failure. Replication and regular backups help maintain data integrity and provide business continuity.

On the client side, customers access the system using various devices such as smartphones, tablets, laptops, or desktop computers connected to the internet. These devices communicate

with the web servers via standard network protocols, allowing users to browse products, place orders, and track deliveries in real time. Additionally, administrative staff and delivery coordinators access the system through secure workstations, which offer specialized interfaces for managing inventory, processing orders, and coordinating logistics.

The overall hardware setup is optimized to ensure the Fresh Grocers Web-Based System performs reliably under varying loads, supports future growth, and delivers a smooth and responsive user experience. This architecture ensures that both customers and internal staff can interact with the platform effectively, contributing to a streamlined and efficient grocery shopping process.

Design Specification

A design specification is a document that describes the needs, expectations, and constraints of a product or system. A design specification's objective is to guarantee that all stakeholders understand what is necessary, anticipated, and achievable before any work starts.

A design specification should be developed early in the product development cycle, ideally during the ideation phase. It should be evaluated and updated on a regular basis as the product changes. The design specification should be made available to all stakeholders and anybody involved in the project.

Importance of design specification

Design specifications are essential for product development since they specify particular criteria, characteristics, and suggestions. Clear communication among stakeholders improves design decisions, reduces errors, aligns expectations, promotes efficient manufacturing, saves time and money, and ensures legal compliance. The specification helps coordinate efforts, prevent misunderstandings, lead designers towards exact objectives, and decrease guesswork. It enables early fault identification, cost-effective production, and project management. Clear material selection, methods, and finishes simplify operations. The specification ensures legal and regulatory compliance while prioritizing safety and environmental considerations. Design criteria are essential for creating successful products that fulfill client and end-user needs.

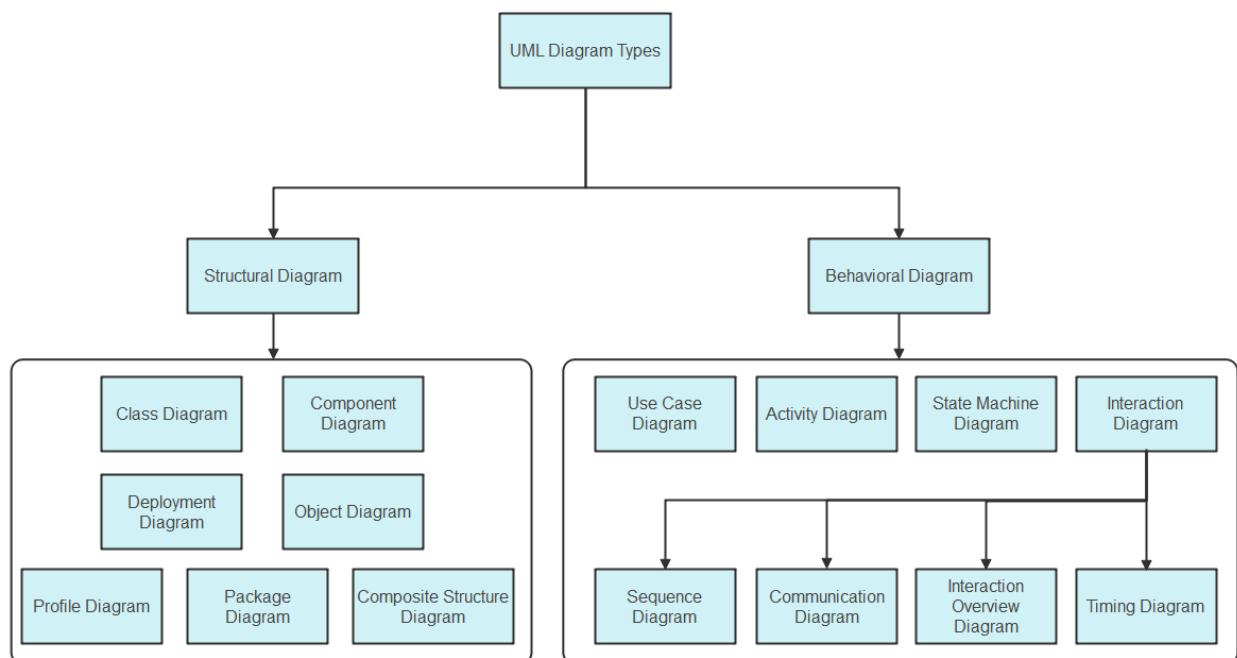
Why we create design specification?

Design specifications generally have a pivotal influence on the software development process because they have several reasons. Initially, the development process is relying on the guidance provided by the requirement specifications. Through the description of the architecture, components, interfaces, and features of the system, design specifications define a plan that programmers will use in the code they develop to match the project specifications. Without specifications, the developers may lack direction because they will be confused. Hence the risk of quality being compromised on the finished product will be eminent. Expressing of design specifications act as a tool providing people involved with the project with an opportunity to collaborate and align. Due to the fact that the design specifications document design decisions, requirements and constraints, it is ensured that not only the developers, designers, project managers and clients but other stakeholders have a joint idea of the system's design, behaviour and function. This common understanding helps to avoid threats, such as disagreements, conflicts and doubts that may come about when the different stakeholders have different interpretations of the project at hand. Clear directions and consistent supply of information, being the design specifications in the technical language, deliver seamless cooperation and increase probability of success.

Maintenance and expansion of the software system will be shaped by the prescribed requirements. When describing architecture, components and interfaces of the system, design specifications serve as conduit for valuable insights about the system's organizational structure. All of these data is monumental especially during the future maintenance, upgrading and improvements process because it helps in explaining the interaction and dependency of the different parts of the system with each other. Besides, designs would always give the developers the ability to pick up areas of improvement or optimization in order to make wisest decision on how to grow the system in a period of time. In this way, designers would specify the specifications that will support the long-term sustainability and the software system evolution, where the system will be adaptable and responsive to the changing requirements of the software and technological advancements.

Unified Modelling Language (UML)

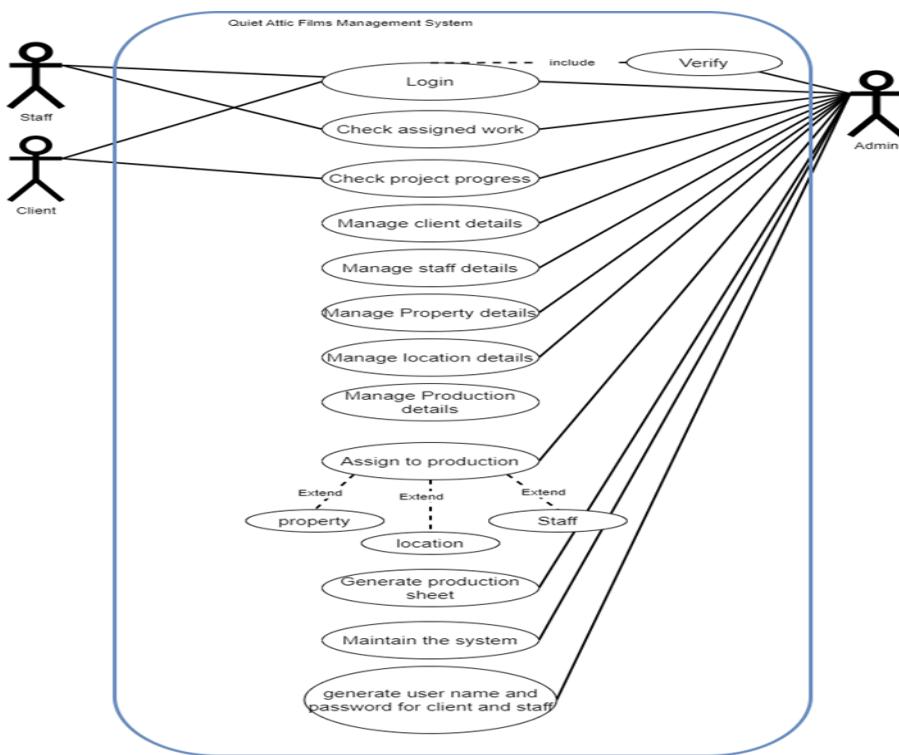
UML, which stands for Unified Modeling Language, is a standardized modeling language comprised of an integrated set of diagrams designed to assist system and software developers in specifying, visualizing, constructing, and documenting software system artifacts, as well as business modeling and other non-software systems. The UML is a set of best engineering practices that have proven useful in modeling big and complex systems.



Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. You may represent a complicated system using a single use-case diagram or multiple use-case diagrams to model the system's components. Use-case diagrams are often created in the early stages of a project and referred to throughout the development process.

Eg:



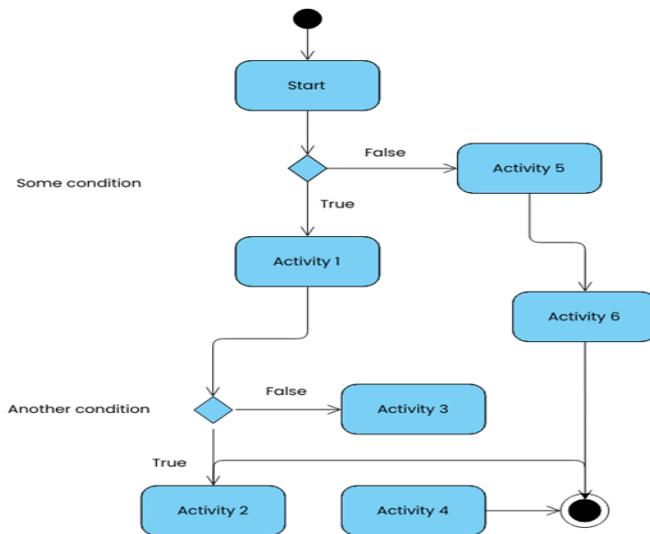
Importance of Use Case Diagram

A use case diagram is Important because it depicts how a system interacts with its users or other external entities in a clear and visible manner. It enables stakeholders, including as developers, designers, and business analysts, to quickly grasp the system's capabilities and needs. This diagram serves as a significant tool for successful communication and requirements analysis by showing the numerous use cases and their linkages, ensuring that the system satisfies its intended goals and user demands.

Activity Diagram

An activity diagram, like a flowchart or a data flow diagram, visually represents a succession of actions or the flow of control in a system. In business process modelling, activity diagrams are frequently employed. They may also use a use case graphic to illustrate the stages. Modelled activities can be both sequential and concurrent. An activity diagram will have a beginning (an initial state) and an end (a final state) in both circumstances.

Eg:



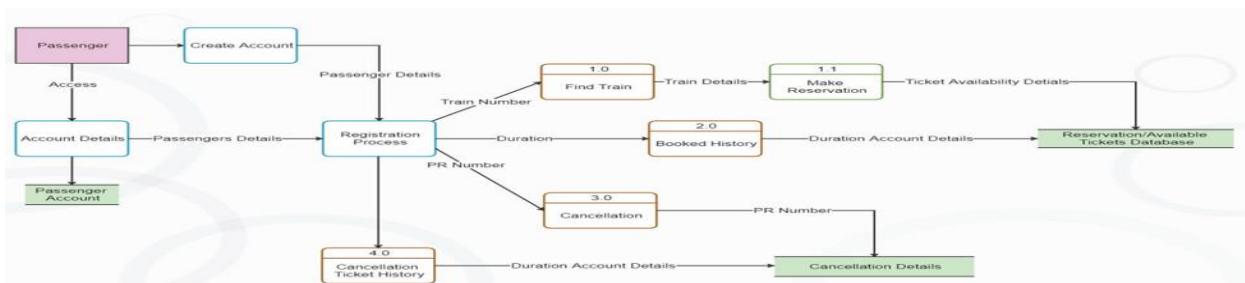
Importance of Activity Diagram

Activity diagrams are Important because they depict the flow of activities inside a system or process visually. They assist stakeholders, like as developers and project managers, with better understanding and communicating complicated operations. Activity diagrams improve clarity, eliminate misconceptions, and allow effective project planning and execution by simplifying the depiction of activities, choices, and dependencies.

Dataflow Diagram

A data flow diagram (DFD) shows how a system processes data in terms of inputs and outputs. Its concentration, as the name implies, is on the flow of information, where data originates from, where it moves, and how it is kept.

Eg:



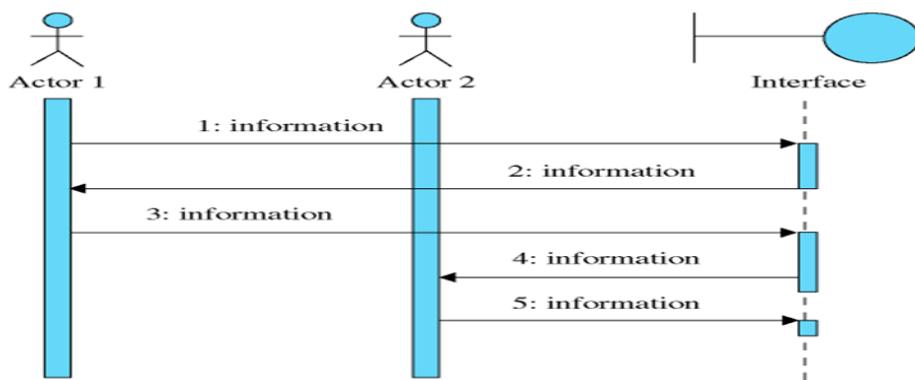
Importance of Dataflow Diagram

A dataflow diagram is important for visualizing and comprehending information flow inside a system or process. It simplifies complicated systems by clearly and intuitively displaying data inputs, processes, and outcomes. This visual tool improves stakeholder communication, allowing for improved decision-making, system design, and troubleshooting. Finally, dataflow diagrams are critical in guaranteeing the efficiency, transparency, and dependability of information flow inside any business or project.

Sequence Diagram

Sequence diagrams show interactions between classes as a series of messages sent over time. They are also known as event diagrams. A sequence diagram is an effective tool for visualizing and validating various runtime scenarios. These can assist forecast how a system will act and identify responsibilities that a class may need to have while modelling a new system.

Eg:



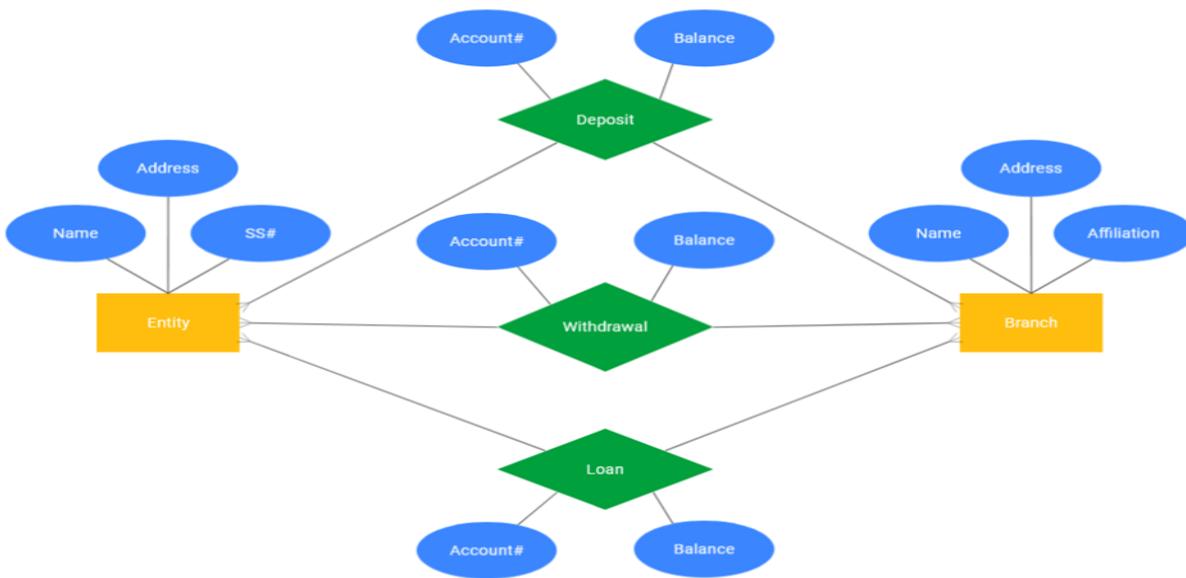
Importance of Sequence Diagram

Sequence diagrams show interactions across classes as an exchange of messages over time. They're also known as event diagrams. A sequence diagram is an excellent tool for visualizing and validating various runtime scenarios. These can assist forecast how a system will behave and identify responsibilities that a class may need to have when modelling a new system.

ER Diagram

The acronym ERD stands for entity relationship diagram. These diagrams are also known as ER diagrams and Entity Relationship Models. An ERD depicts the relationships between entities in a database, such as persons, things, or concepts. An ERD will frequently depict the properties of these entities. An ER diagram may depict the logical structure of databases by specifying the entities, their properties, and the interactions between them. This is useful for engineers who want to either describe an existing database or sketch up a concept for a new database.

Eg:



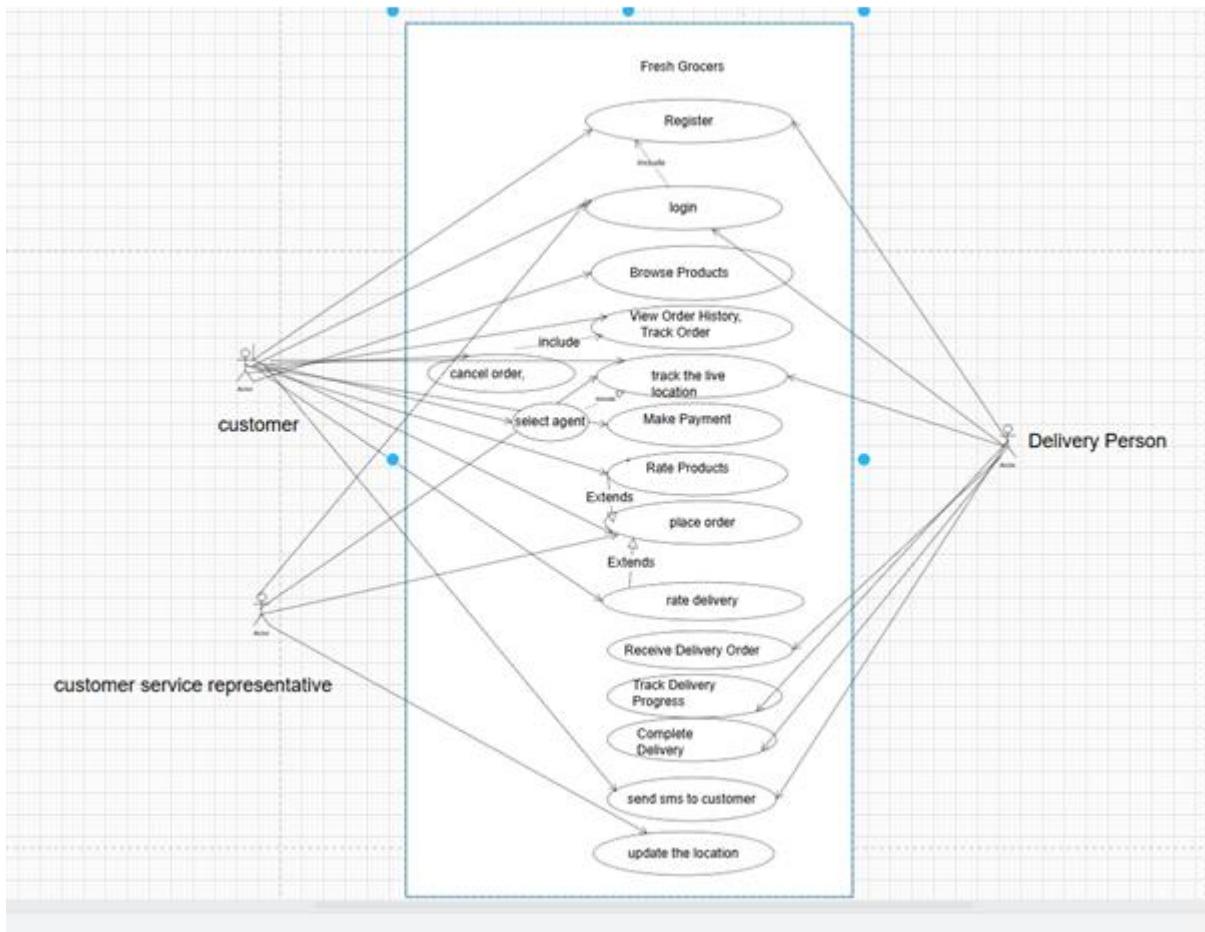
Importance of ER Diagram

An Entity-Relationship (ER) diagram is important in database architecture because it gives a visual depiction of a system's data structure and connections. This graphic aids in the clarification of complicated data models, ensuring that database tables are effectively arranged and data is saved and retrieved appropriately. An ER diagram serves as a vital communication tool between stakeholders, including developers, designers, and business analysts, by depicting

entities, attributes, and connections between them, fostering a shared understanding of the data schema and facilitating effective database development and maintenance.

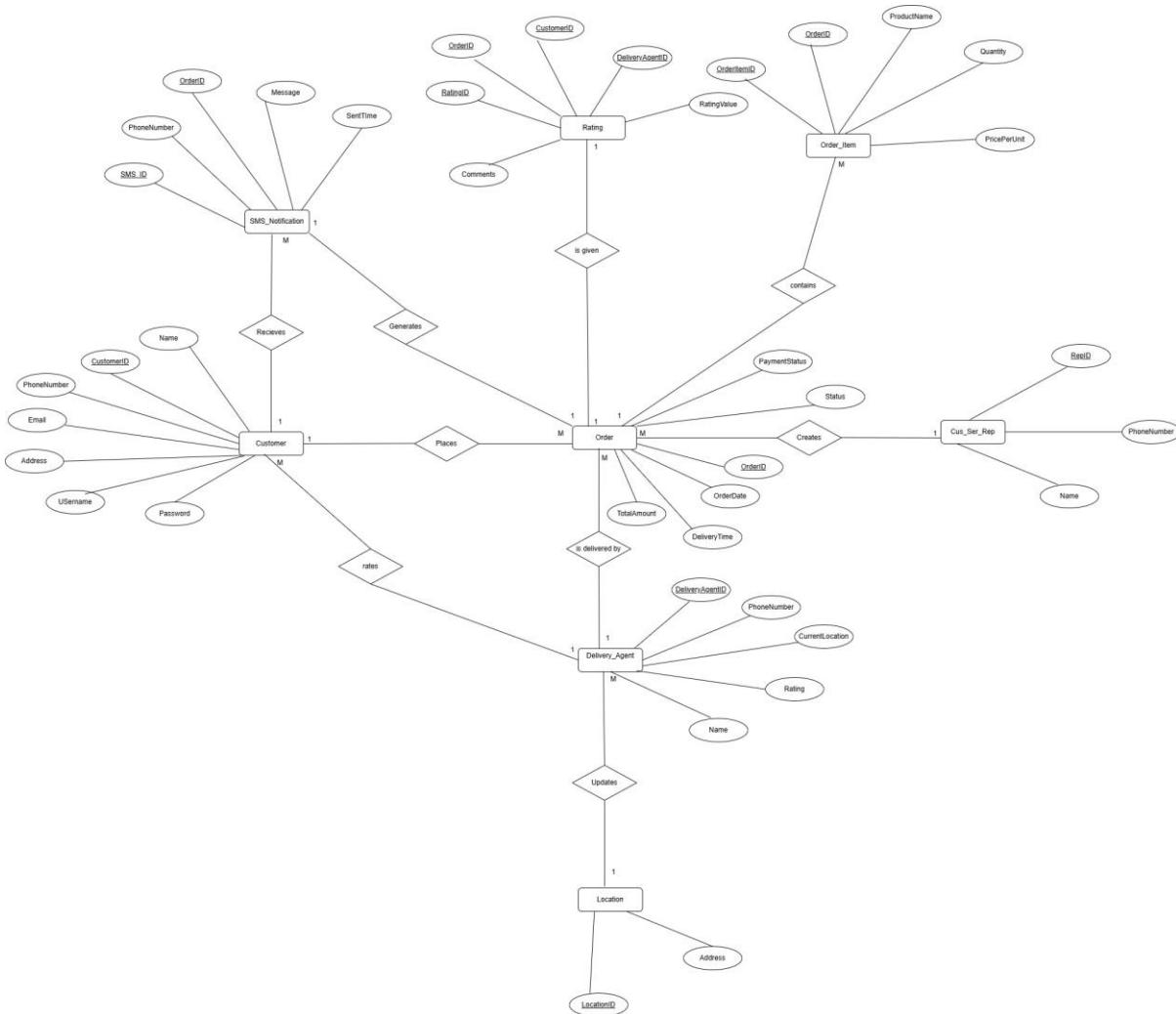
Design Documentation for Fresh Grocers

Use case Diagram



The author identifies Customer, delivery person and customer service representative as key Actors in the proposed use case diagram. The author has highlighted each major method as a distinct use case. The image depicts the interactions and duties of different system participants, emphasizing their specific activities. This strategy ensures clarity and understanding of the use cases in the situation.

Entity Relationship Diagram



This Entity-Relationship Diagram (ERD) is a conceptual representation of the data structure for the Fresh Grocers web-based grocery ordering system. It includes five main entities: Customer, delivery agent, Order, Product, and location. The Customer entity, which likely stores information such as customer ID, name, address, and contact details, is linked to the Order entity. The Order entity contains data like order ID, date, time, total amount, and delivery status, representing each grocery order placed by a customer.

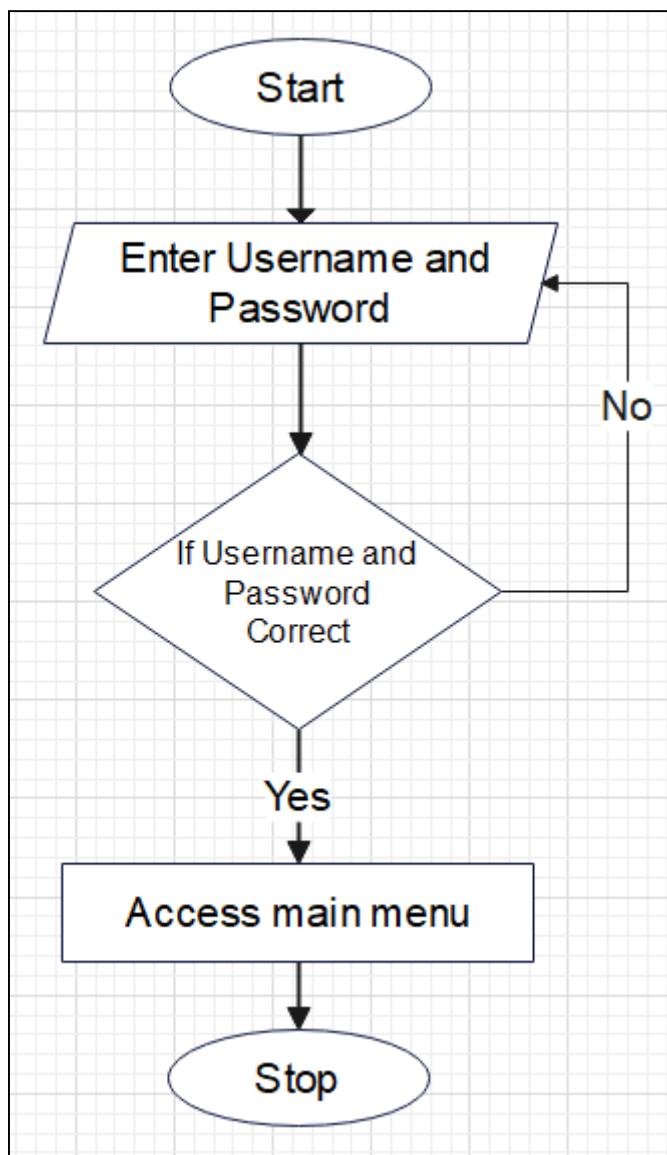
The delivery agent entity, which includes agent ID, name, and assigned role, is also related to the Order entity, indicating which agent is delivered the order. The Product entity holds

information about grocery items such as product ID, name, category, price, and stock quantity. This entity is connected to Order through an associative relationship (like Order Details), since one order can contain multiple products, and one product can appear in many orders.

Lastly, the User entity manages login credentials for both Customer, delivery agent including username and password. Since both need to log into the system, they are linked to the User entity for authentication purposes. This ERD captures the relationships between these entities and provides a clear view of how data flows within the Fresh Grocers system.

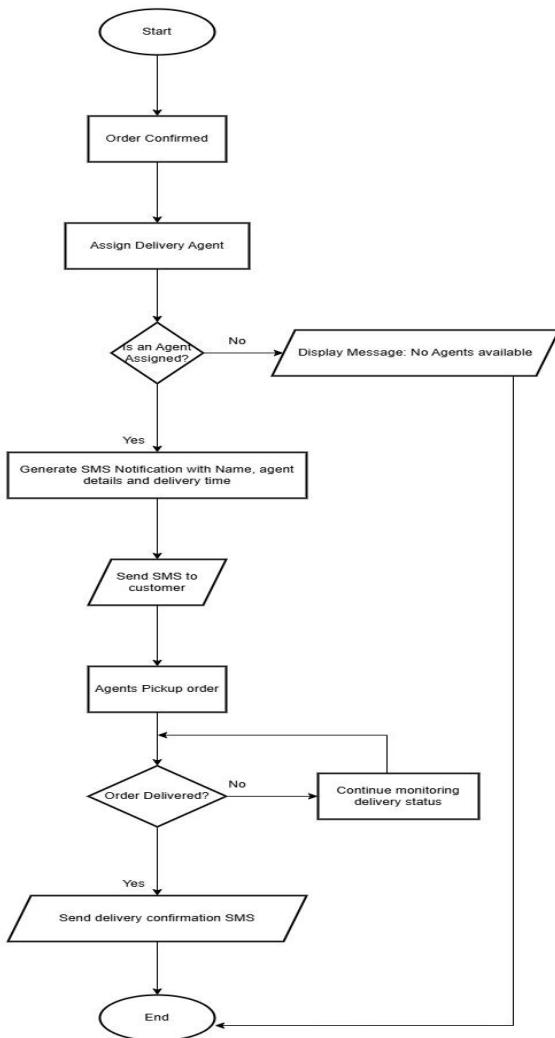
Flow chart

Flow chart - Login



The flowchart depicts the login screen, which requires users to enter their username and password. Access to the main menu is allowed upon successful entrance; otherwise, invalid credentials result in login failure.

Flow chart – Delivery

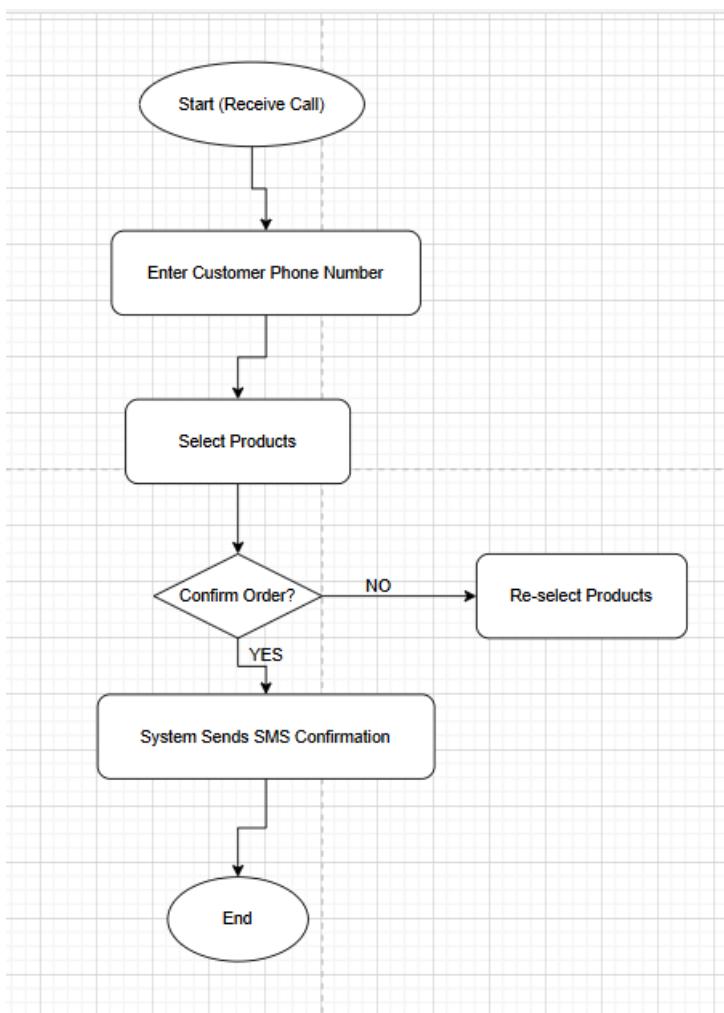


This flowchart shows how a product is delivered to a customer. It starts with confirming the order, then a delivery agent is assigned to that order. After that, the agent goes and delivers the product to the customer's location. The flow is clear and steps are simple to understand. It helps to see how the delivery process works from start to finish. But still, it is a bit too basic. It doesn't show what happens if the delivery fails – like if the address is wrong, or the customer is not at home. Also, there is no

info about tracking updates. These days, customers usually want to see where their delivery is and get updates on phone or app. This flowchart can be better if it show when and how those updates are sent.

Another thing missing is the feedback or action taken when delivery is late. And what if the delivery agent is sick or can't do it? The chart don't show any backup process. Also, it don't show if customer can contact support during delivery. Adding these parts will help to understand the full process better. Right now, it only show the normal steps, not the problems or solutions. So, it's a nice simple flow, but need more to handle real-life cases and customer service parts.

Flow chart –Manual Order by Customer Service Representative

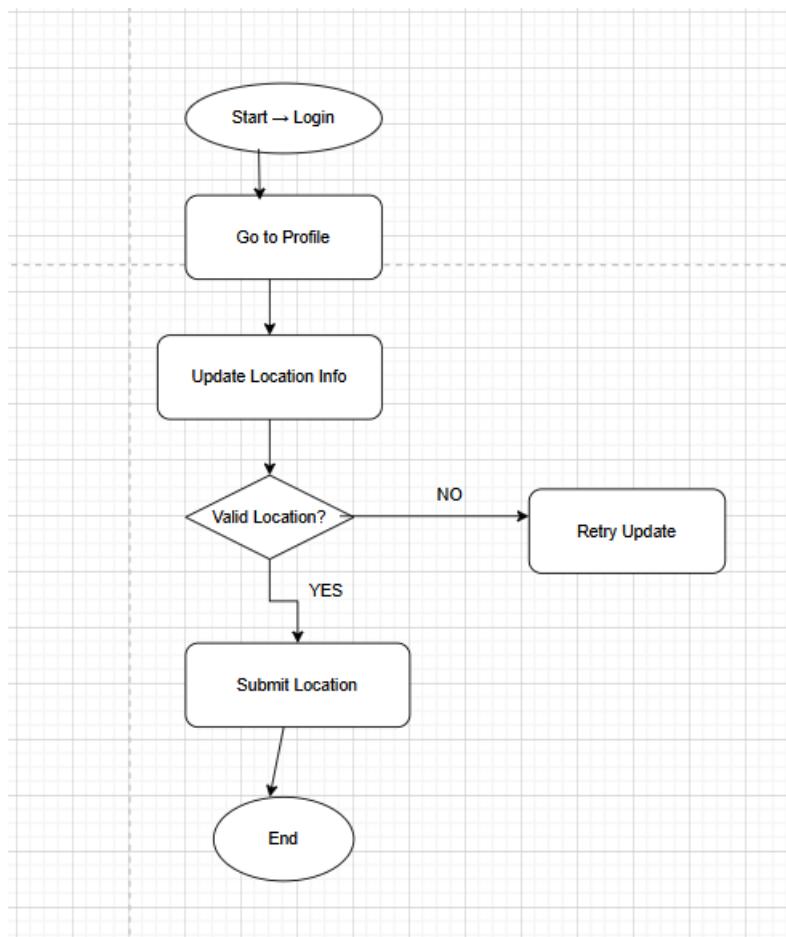


This chart explain how a customer service rep take an order manually from a customer. This is helpful when customer don't order online and need help. The chart begin when customer call

or visit the shop, then the rep ask for order info, check product availability, and place the order into the system. It shows how human interaction work in order process. It's useful for understanding the offline or phone order method. But some parts are missing which are important.

For example, it doesn't show what happens if the product is out of stock. Does the rep suggest other product? Or does he cancel the order? Also, there's no step for correcting errors if wrong product code or address is entered. That can happen often when writing things manually. The flowchart also doesn't mention how payment is handled. Is it cash, card, or online? It's not clear. Another missing part is if customer change their mind after placing order. Can the rep cancel or edit the order? Also, no info is given about confirmation – does the customer get a receipt or message? The chart is good for showing the basic flow, but it needs more info about what happens when problems come or customer requests changes. Adding decision boxes and more action steps will make the chart more useful and realistic for training or system designing.

Flow chart – Delivery Agent Updating Location



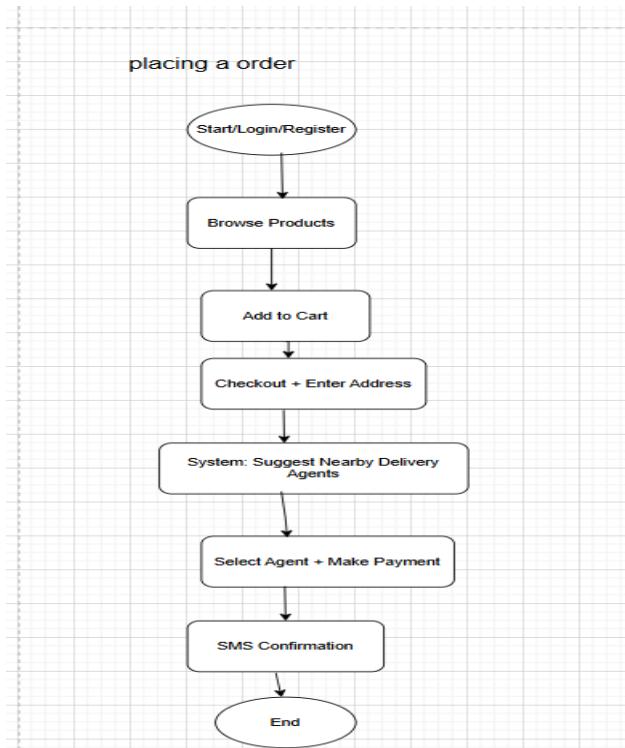
This flowchart is small and focus on just one task – the delivery agent updating their location. It's easy to understand and follow, and show how agent inform the system where they are during delivery. This is important so customers and the system know where the order is. But the chart is too simple. It only show agent entering location, but don't show how system receive it or what happens next.

For example, there's no info if the location update fails – like if internet not working, or GPS gives wrong data. Also, it don't show if agent get confirmation after sending the location. In real systems, location updates usually link to tracking feature, and customer get notified. That part is missing.

It also don't say how often location should be updated, or if agent can skip it. No step for error checking or backup method if app crashes. Also, system response or customer alerts should be added in this chart to make it more useful.

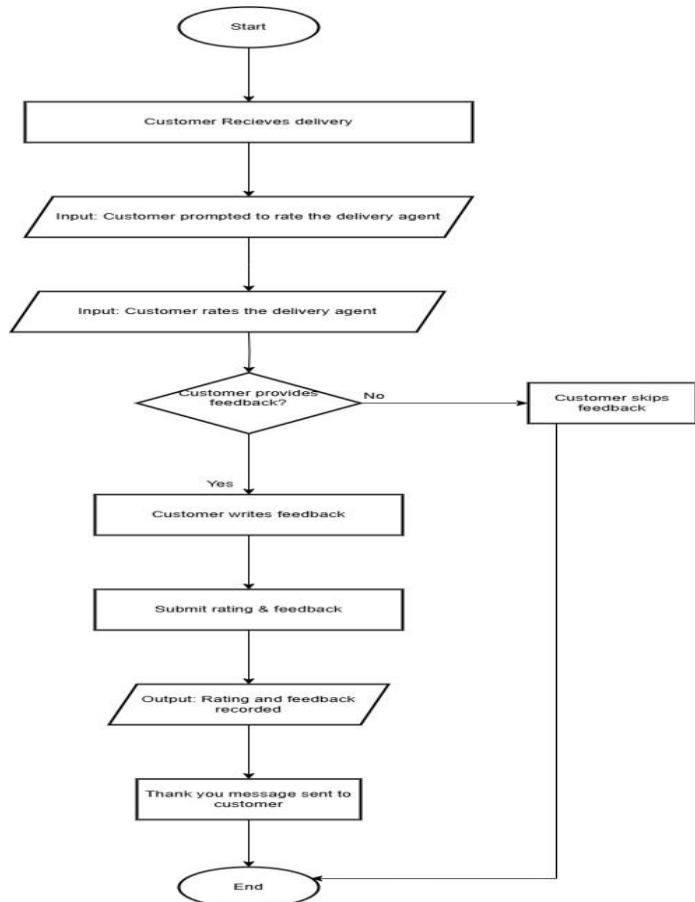
So, while it is a good chart to understand the location update step, it feel incomplete. In real life, many technical things can go wrong, and those should be in the chart too. Adding more conditions and actions will help make it stronger and ready for use in real delivery app or system.

Flow chart – Placing an Order



This chart shows how a customer places an order, probably using a website or app. It starts from the moment when the customer browses the items, selects what they want, adds them to the cart, then goes to checkout and pays. The steps are clear and in the right order, so anyone can understand how online ordering works. But the chart doesn't show what happens if something goes wrong. For example, if payment fails or internet disconnects during checkout, there is no action shown. Also, sometimes customers change their mind and remove items from the cart. That's not shown in the flowchart. It also doesn't mention if the customer needs to log in or sign up before placing an order. In most real systems, login is required before payment. The chart can also include options like applying discount codes, choosing delivery method, or checking estimated delivery time. After payment, customers usually get a confirmation message or receipt, but that is not shown here. It would be better if the chart had these extra steps. So, while the flowchart is good for showing the basic order process, it needs to include more real situations like errors, cancellations, or payment options. With some more steps, it can become a full and proper guide for an online ordering system.

Flow chart – customer ratings for delivery agent



This chart explain how customer give feedback after getting the delivery. It show simple steps like delivery done, customer open app, rate the agent and submit the feedback. It is a useful part of delivery process because rating help to improve service. The steps are clear and easy to follow, but the chart is very short and miss many parts.

It don't say what happen after rating is submitted. Does the company store it? Is the agent informed about their rating? If the rating is bad, do managers take any action or talk to the agent? These things are important but not included. Also, it don't show if customer can add comments or just give stars. Sometimes, written feedback is more helpful than just rating.

Another thing missing is what happen if customer skip the rating or close the app. Is the rating optional or required? That decision box should be added. Also, there is no step about how the system use the rating – does it go to agent profile or help in performance review?

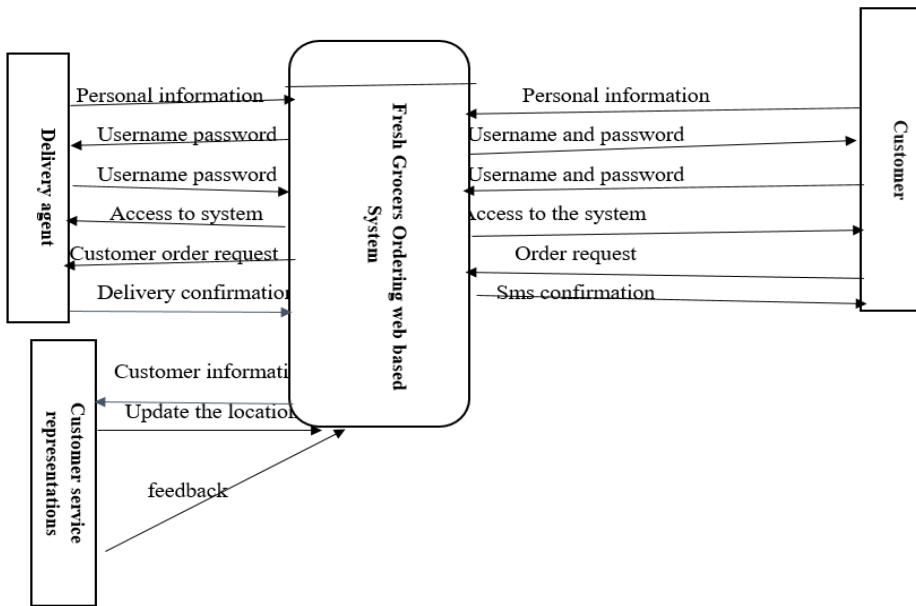
Overall, the chart show a good starting point for customer feedback, but it need more details to show what happens after rating is given and how it is used for improvement in the delivery process.

Data Flow Diagram for Fresh Grocers

Level 0 - Data Flow Diagram Fresh Grocers

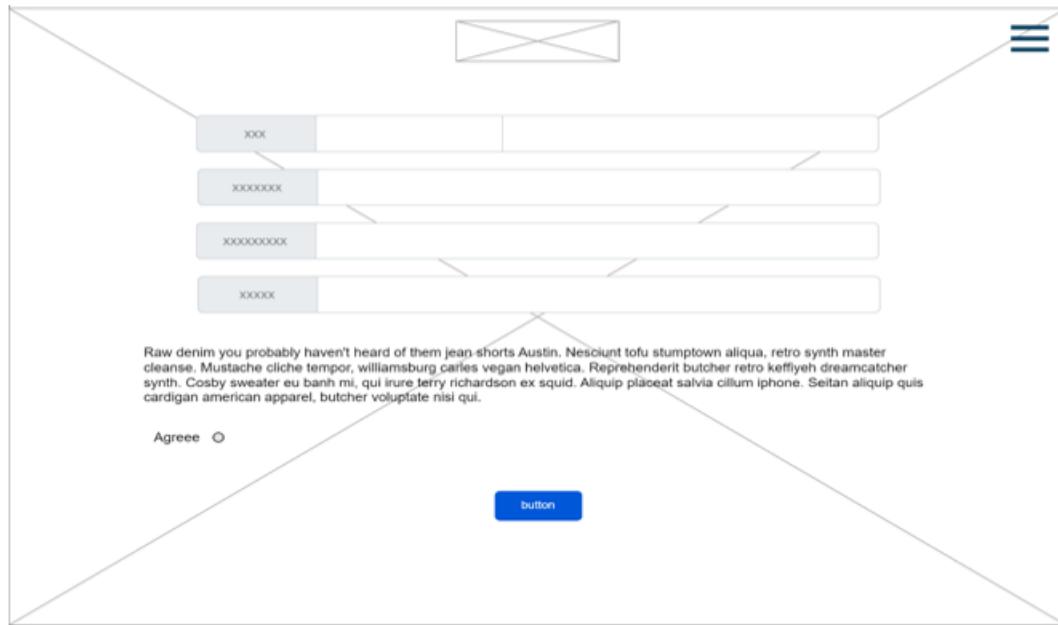
Data Flow Diagram

Context Diagram (Level 0)



Prototypes of fresh grocers

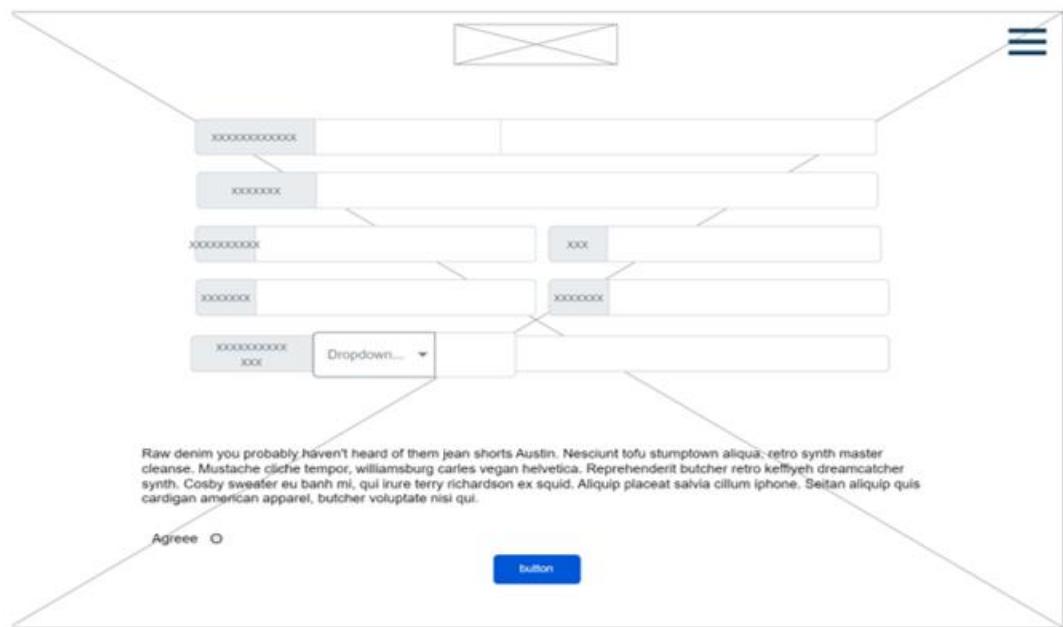
Registration page of customer



This is a wireframe prototype of a registration page for a customer. The layout includes:

- A header section with a close button (X) and a menu icon (three horizontal lines).
- Four input fields for personal information, each preceded by a placeholder like "XXXXXX" or "XXXXXXXX".
- A large text area containing placeholder text: "Raw denim you probably haven't heard of them jean shorts Austin. Nesciunt tofu stumptown aliqua, retro synth master cleanse. Mustache cliche tempor, williamsburg carles vegan helvetica. Reprehenderit butcher retro keffiyeh dreamcatcher synth. Cosby sweater eu banh mi, qui irure ferris richardson ex squid. Aliquip placeat salvia cillum iphone. Seitan aliquip quis cardigan american apparel, butcher voluptate nisi qui."
- Two buttons at the bottom: "Agreee" with a radio button and a blue "button" button.

Registration page of delivery agent



Delivery agent rating page



Customer service representative page

CSR of Fresh Grocers

ONGOING DELIVERY

SMS

PENDING DELIVERY

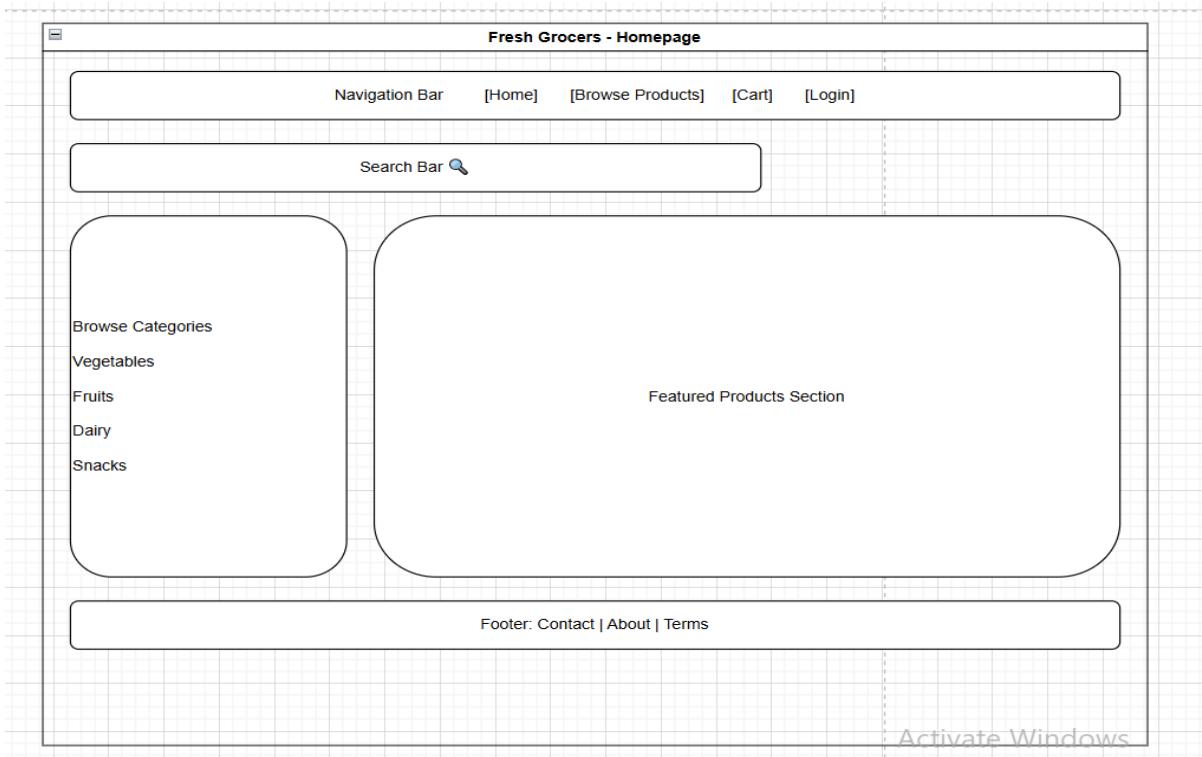
TODAY FINISHED DELIVERY

NAME
ADDRESS
PHONE NUMBER
PRODUCT
EMAIL ID

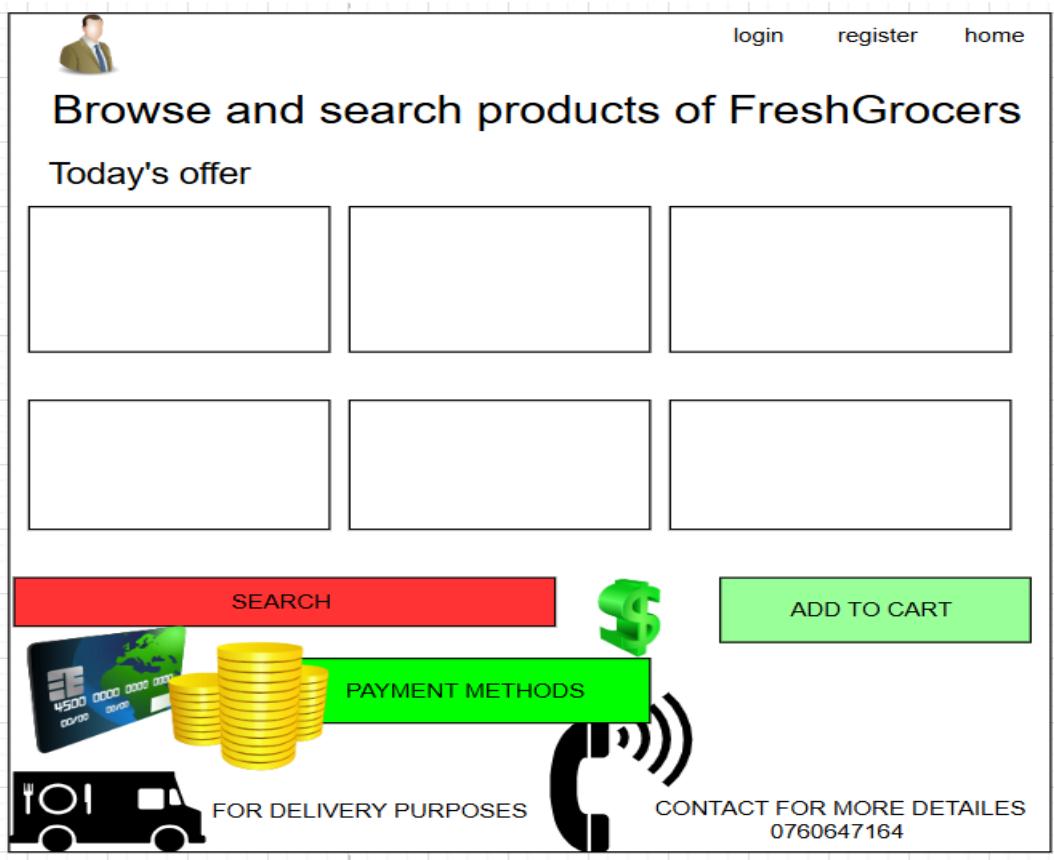
update current location

SUBMIT

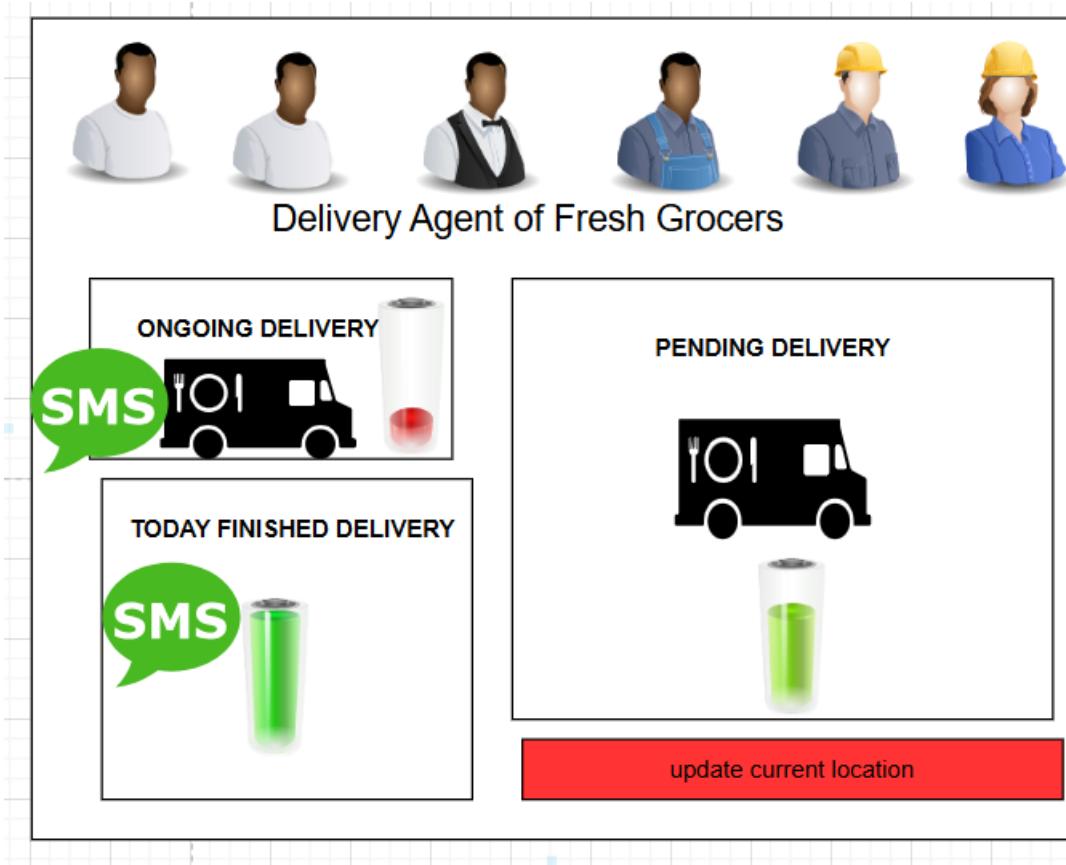
Home page of fresh grocers



Browse and search product of fresh grocers



Delivery agent of fresh grocers



Login/registration page

The form is titled "Login / Register". It features two input fields: "Username" and "Password", each with a small icon to its left. Below the fields are two buttons: a green "Login" button and a blue "Register" button.

Add to cart of fresh grocers



login register home search cart

Add to cart of Fresh Grocers

YOUR SELECTED PRODUCTS



CONTACT
0760647164

SEARCH

CANCEL PRODUCT

Shopping cart of fresh grocers



login register home search

Shopping cart of Fresh Grocers

YOUR BUYING PRODUCTS

SEARCH



CONTACT
0760647164





CANCEL PRODUCT

PAYMENT METHODS

Past orders and payment

This mockup shows a user interface for managing past orders and payments. At the top, there's a navigation bar with links for login, register, home, search, and cart. A profile icon of a person in a uniform is on the left. The main title is "PAST ORDERS AND PAYMENT". Below it, a section titled "YOUR PAST ORDERS PRODUCTS" displays four items:

- Item 1: Placeholder box with a green dollar sign icon and text "RS 1000".
- Item 2: Placeholder box with a yellow coin stack icon and text "RS 2000".
- Item 3: Placeholder box with a blue credit card icon and text "RS 10000".
- Item 4: Placeholder box with a yellow coin stack icon and text "RS 1100".

Below these items are two buttons: "CONTACT 0760647164" next to a delivery truck icon, and a red "SEARCH" button. At the bottom is a red "ANY INQUIRIES" button.

Installation Environment

Registration page of customer

This mockup shows a registration form. It features a header with a close button and a menu icon. Below the header are four input fields, each containing a placeholder starting with "XXXX". A large paragraph of placeholder text follows, reading: "Raw denim you probably haven't heard of them jean shorts Austin. Nesciunt tofu stumptown aliqua, retro synth master cleanse. Mustache cliche tempor, williamsburg carles vegan helvetica. Reprehenderit butcher retro keffiyeh dreamcatcher synth. Cosby sweater eu banh mi, qui irure terry richardson ex squid. Aliquip placeat salvia cillum iphone. Seitan aliquip quis cardigan american apparel, butcher voluptate nisi qui." At the bottom left is a checkbox labeled "Agreee" with a radio button. In the center is a blue "button" with white text.

Menu Button, Three horizontal lines indicating additional options or navigation.

Name Field, A text box for entering the customer's full name.

Email Field, A text box for entering the customer's email address.

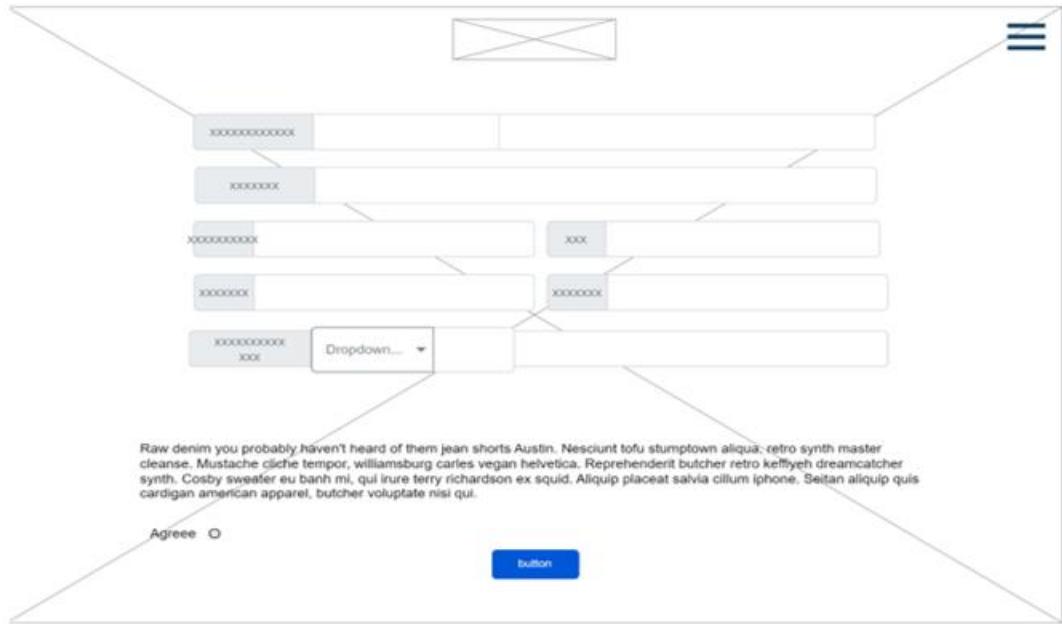
Phone Field, A text box for entering the customer's phone number.

Address Field, A text box for entering the customer's physical address.

Agreement Checkbox, A checkbox for agreeing to terms and conditions.

Submit Button, A button to submit the registration form.

Registration page of delivery agent



The wireframe shows a registration form with the following components:

- Header:** Includes a close button (X) and a menu icon (three horizontal lines).
- Text Fields:** Six input fields labeled with placeholder text: "XXXXXXXXXXXXXX", "XXXXXXX", "XXXXXXXXXX", "XXXXXXX", "XXXXXXX", and "XXXXXXX".
- Dropdown:** A dropdown menu labeled "Dropdown.... ▾".
- Text Area:** A large text area containing placeholder text: "Raw denim you probably haven't heard of them jean shorts Austin. Nesciunt tofu stumptown aliqua; retro synth master cleanse. Mustache cliche tempor, williamsburg carles vegan helvetica. Reprehenderit butcher retro keffiyeh dreamcatcher synth. Cosby sweater eu banh mi, qui irure terry richardson ex squid. Aliquip placeat salvia cillum iphone. Seitan aliquip quis cardigan american apparel, butcher voluptate nisi qui."
- Buttons:** Two buttons at the bottom left: "Agreee" with a radio button and a blue "button" with white text.

Menu Button, Three horizontal lines indicating additional options or navigation.

Name Field, A field for entering the driver's full name.

Email Field, A field for entering the driver's email address.

Phone Field, A field for entering the driver's phone number.

Address Field, A field for entering the driver's address.

License Number, A field for entering the driver's license number.

Expiry Date, A section for entering the expiry date of the license or registration.

Vehicle Details, Fields to enter vehicle model & registration number with a

dropdown option for car models.

Agreement Checkbox, A checkbox indicating agreement to terms and conditions.

Submit Button, A button to submit all entered information.

Delivery agent rating page



The Delivery Ratings Page allows customers to share their feedback on the delivery experience after receiving their orders. This page is usually accessible from your order history or after the delivery status shows as “Delivered.”

On this page, you will typically find options to rate various aspects of the delivery service, such as timeliness, condition of the package, and the professionalism of the delivery staff. Ratings are often given using stars (for example, 1 to 5 stars) or smiley faces, along with a space to write additional comments or suggestions.

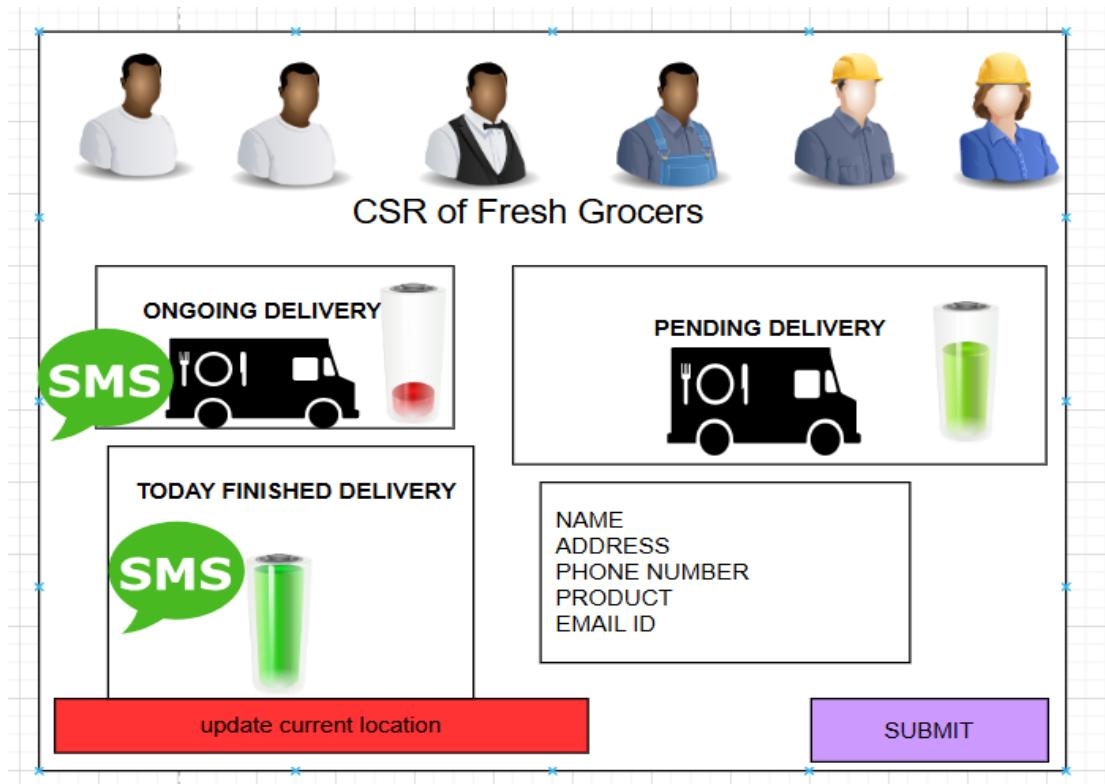
By submitting your delivery rating, you help the store or delivery service understand what they are doing well and where they can improve. High ratings indicate satisfaction and encourage the team, while constructive feedback highlights areas that need attention.

Using the delivery ratings page benefits both customers and businesses. Customers

feel heard and involved in improving the service, while businesses gain valuable insights to enhance delivery speed, packaging quality, and overall customer satisfaction.

Providing honest and thoughtful ratings contributes to better service for everyone and helps build trust between the company and its customers. It's a quick and easy way to make your voice count after every purchase.

Customer service representative page



Could show delivery status if clicked.

Example: "Order Confirmed," "Out for Delivery," or "Delivered."

Helps customers track their order after purchase.

The Order Tracking feature is usually found near the contact section of the website or app, often indicated by an icon like a food truck or delivery van. Although this might be a future addition, its purpose is to keep customers informed about the current status of their orders after they have made a purchase.

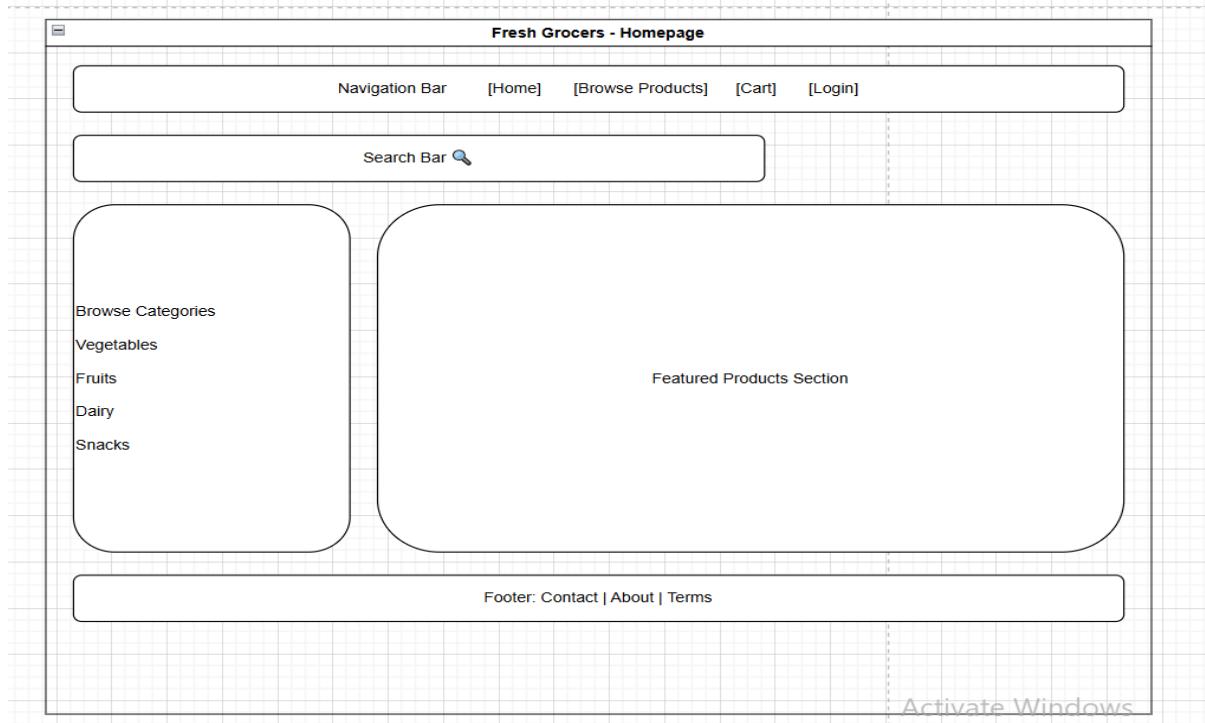
When fully implemented, clicking on the food truck icon could open a small window or page showing real-time updates on your order's progress. For example,

it may display messages such as “Order Confirmed,” “Out for Delivery,” or “Delivered.” These updates provide clear and timely information, so you know exactly where your order is and when to expect it.

The main benefit of this feature is to enhance customer satisfaction and reduce anxiety by providing transparency throughout the delivery process. Instead of guessing or having to call customer service for updates, you can quickly check your order status with a simple click.

Order tracking improves trust and convenience, helping customers feel more confident in their online shopping experience. It also helps in planning for receipt of the items, making the whole process smoother and more reliable.

Home page of fresh grocers



Takes you back to the main page.

Displays featured products and categories like vegetables, fruits, and snacks.

Useful if you’re browsing or lost in other pages.

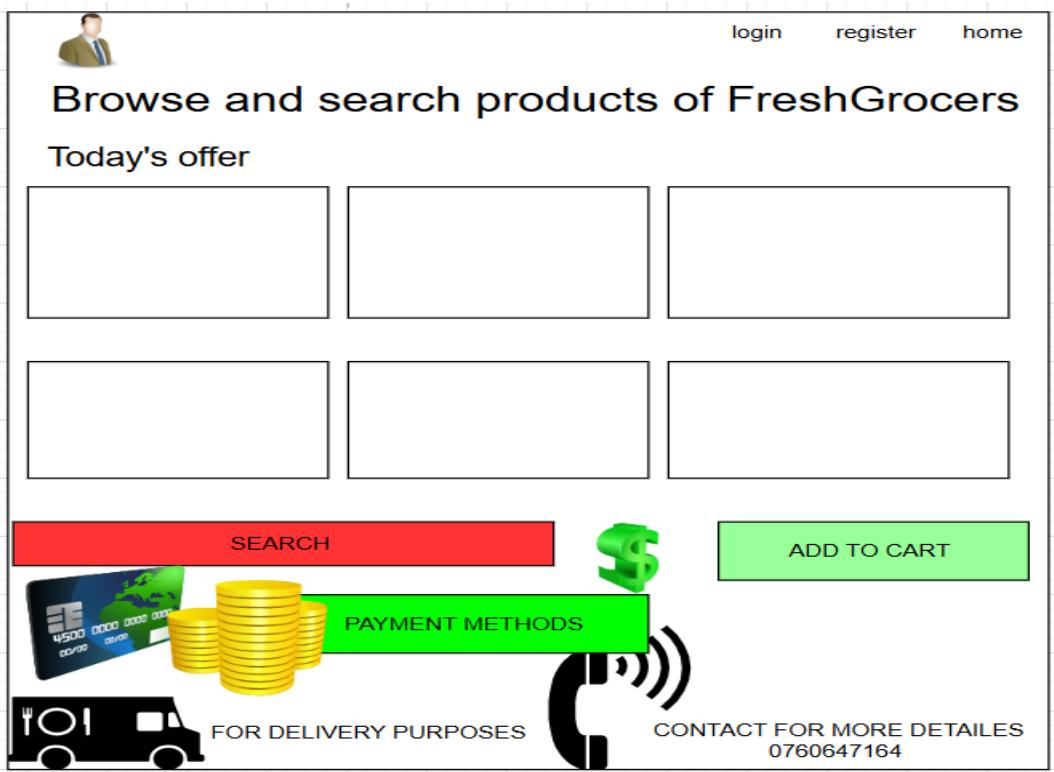
The Home button is placed at the top of the website, usually on the navigation bar where it's easy to see. It works as the main point where users can always return if they are confused or want to start fresh. When someone clicks the Home button, the

website will take them back to the main page of the Fresh Grocers site. On this page, users can see featured products, special offers, and popular categories like vegetables, fruits, dairy, bakery, and snacks.

This button is very useful because it acts like a restart. If a user was searching for something, checking their cart, or browsing a specific category and wants to stop and go back to the beginning, they just need to click the Home button. It clears any filters or search results and shows a clean overview of all the main products available in the store.

For users who are not very familiar with websites, the Home button is a simple and safe place to click when they are unsure what to do next. It helps keep the shopping experience easy and less confusing by always offering a clear place to start over.

Browse and search product of fresh grocers



Search box and red SEARCH button at the center or top.

After typing a product name (e.g. "Apple"), clicking the button will show results below.

The screen updates to show only matching items.

Quickly find specific products from a large list.

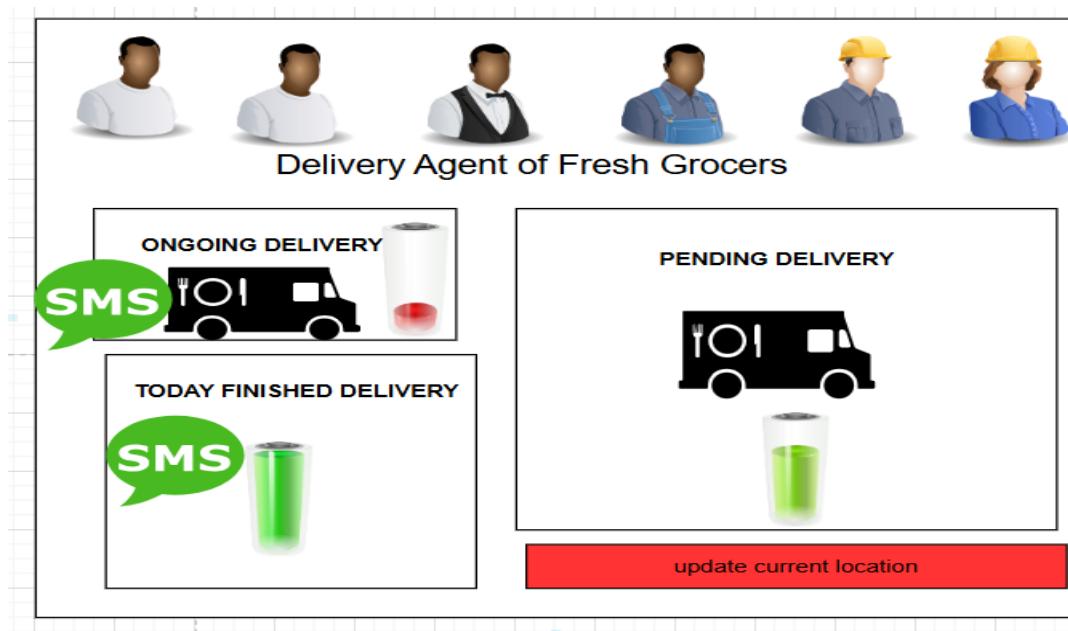
The Search feature is designed to help you quickly find specific products within a large collection. Typically, the search box is located prominently at the center or top of the webpage or app screen, accompanied by a red SEARCH button beside or below it.

To use this feature, simply click inside the search box and type the name of the product you want to find, such as “Apple.” Once you have entered your keyword, click the red SEARCH button to begin the search process.

After clicking the button, the page will update to display only the items that match your search term. For example, if you searched for “Apple,” the results will show products related to apples, filtering out unrelated items. This real-time update helps you quickly narrow down your options and find exactly what you need without having to browse through the entire catalog.

The main use of the search function is to save time and improve convenience, especially when dealing with large inventories. Instead of scrolling through multiple pages or categories, you can directly access relevant products. This makes the shopping experience more efficient and user-friendly.

Delivery agent of fresh grocers



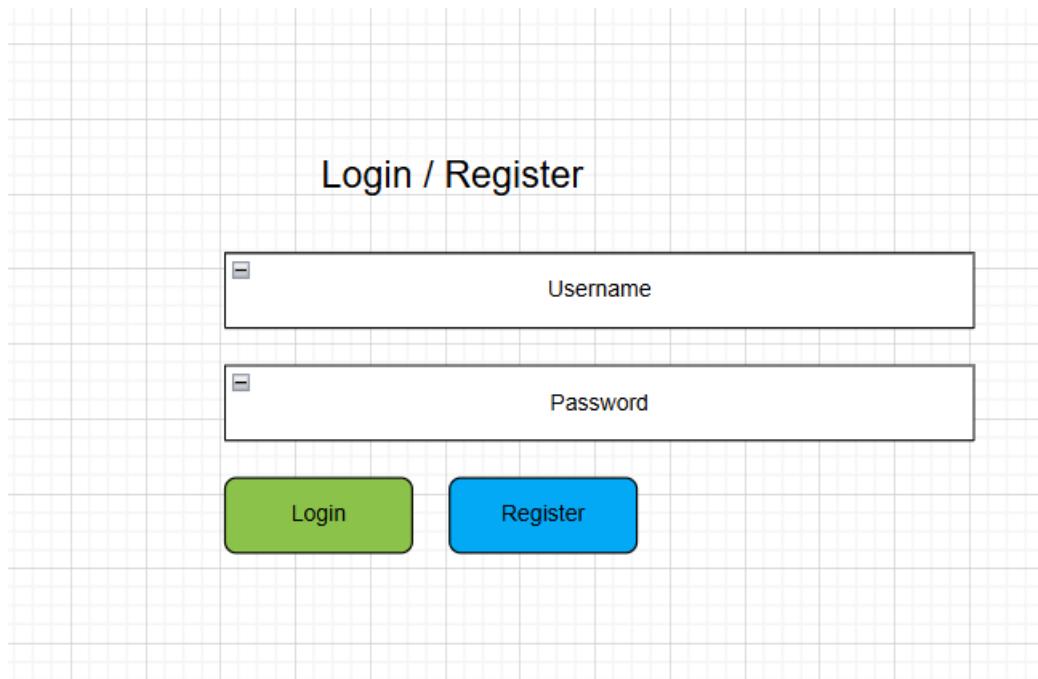
The delivery status icon can be found near the contact section of the website. It is

represented by a food truck symbol, which is easy to recognize and matches the idea of grocery delivery. Even though this feature may not be active yet, it is planned as a future upgrade to help customers know what is happening with their orders.

When this feature becomes active, clicking on the food truck icon could open a small window or page that shows the current status of your order. For example, it might say “Order Confirmed” if the shop has received your order, “Out for Delivery” if the delivery agent is on the way, or “Delivered” once the items have arrived at your door. This type of real-time information helps users feel more secure and informed after they’ve made a purchase.

This future delivery status feature is especially helpful for users who want to plan their day around the arrival of their groceries. It reduces the need to call customer support to ask for updates. Instead, users can simply check their delivery progress on the website by clicking the icon, making the experience easier and more convenient.

Login/registration page



The Login / Register button is usually located at the top-right corner of the website or app screen, clearly labeled as Login or Register. This button is the gateway for

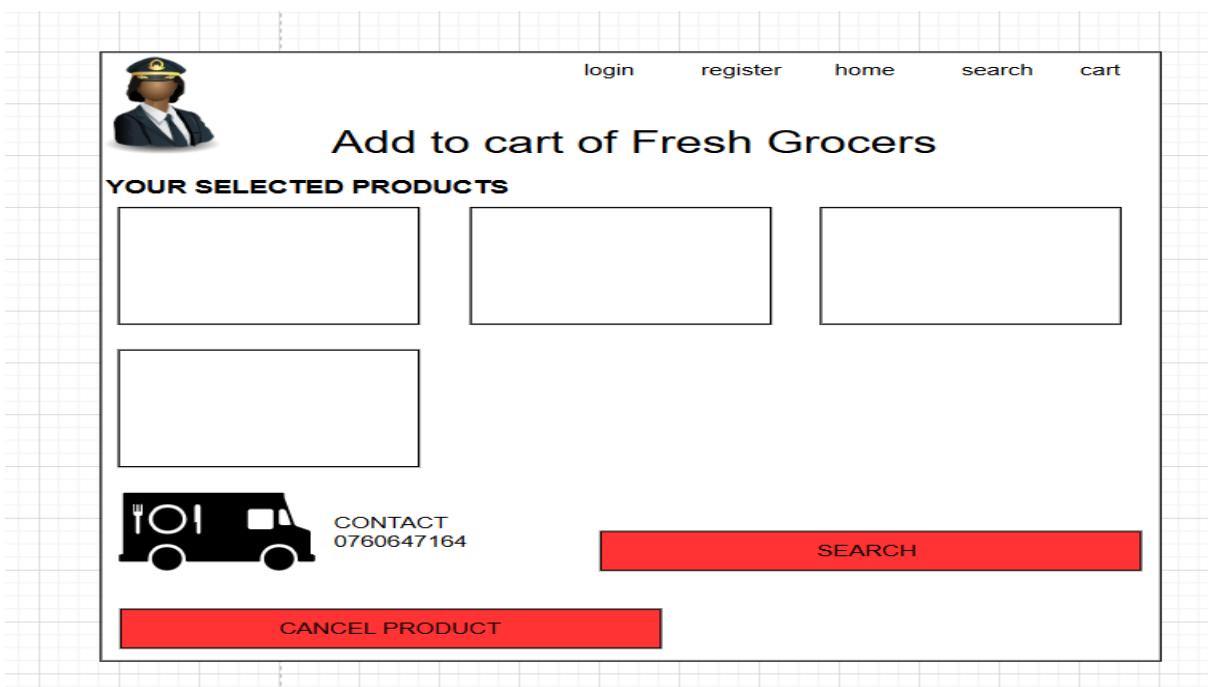
users to access their personal accounts or create a new one, which is necessary before placing orders or accessing personalized features.

When you click the Login button, a pop-up box or a new page appears prompting you to enter your username and password. This secure process verifies your identity, allowing you to access your saved preferences, order history, and personalized settings.

If you are a new user, clicking the Register button will open a form where you need to provide basic details such as your full name, email address, password, and sometimes your delivery address. This information creates a new account linked to you, making future shopping faster and easier.

The main purpose of this feature is to give users a secure way to manage their shopping experience. By logging in, returning customers can quickly reorder items, track shipments, and save multiple delivery addresses. Registering an account also helps the platform offer personalized recommendations and special offers. Overall, the Login/Register function is essential for a smooth, personalized, and secure online shopping experience.

Add to cart of fresh grocers



View and manage selected items.

Section titled “YOUR SELECTED PRODUCTS.”

Review your selected items here.

Click CANCEL PRODUCT to remove any unwanted item.

On or near product images – a button labeled “Add to Cart.”

The product will be added to your cart section on the same page.

A small notification or tick mark may appear confirming the item is added.

Allows you to gather all products you want to buy before checkout.

All products you have clicked “Add to Cart” on will appear here.

Each item will show name, price, and quantity.

Review your choices before placing the order.

The section titled "YOUR SELECTED PRODUCTS" is found in the middle part of the screen. This area is very important because it shows you all the items you have chosen to buy. Every time you click the “Add to Cart” button on a product, that item will appear in this section. It works just like a virtual shopping basket where you can see what you’ve picked up while browsing the website.

In this section, you will clearly see the name of each product, its price, and the quantity you selected. For example, if you added two packs of apples and one bottle of milk, both will be listed with the correct details. This helps you double-check what’s in your cart before moving to the checkout or payment page.

This area is very useful for reviewing your items. If you notice that something is missing or if you added the wrong item, you can go back and fix it easily. It saves time and helps you avoid mistakes. Before finalizing your purchase, always check this section to make sure your grocery list is correct. It makes shopping online simple, clear, and stress-free.

Shopping cart of fresh grocers



On or near product images – a button labeled “Add to Cart.”

The product will be added to your cart section on the same page.

A small notification or tick mark may appear confirming the item is added.

Allows you to gather all products you want to buy before checkout.

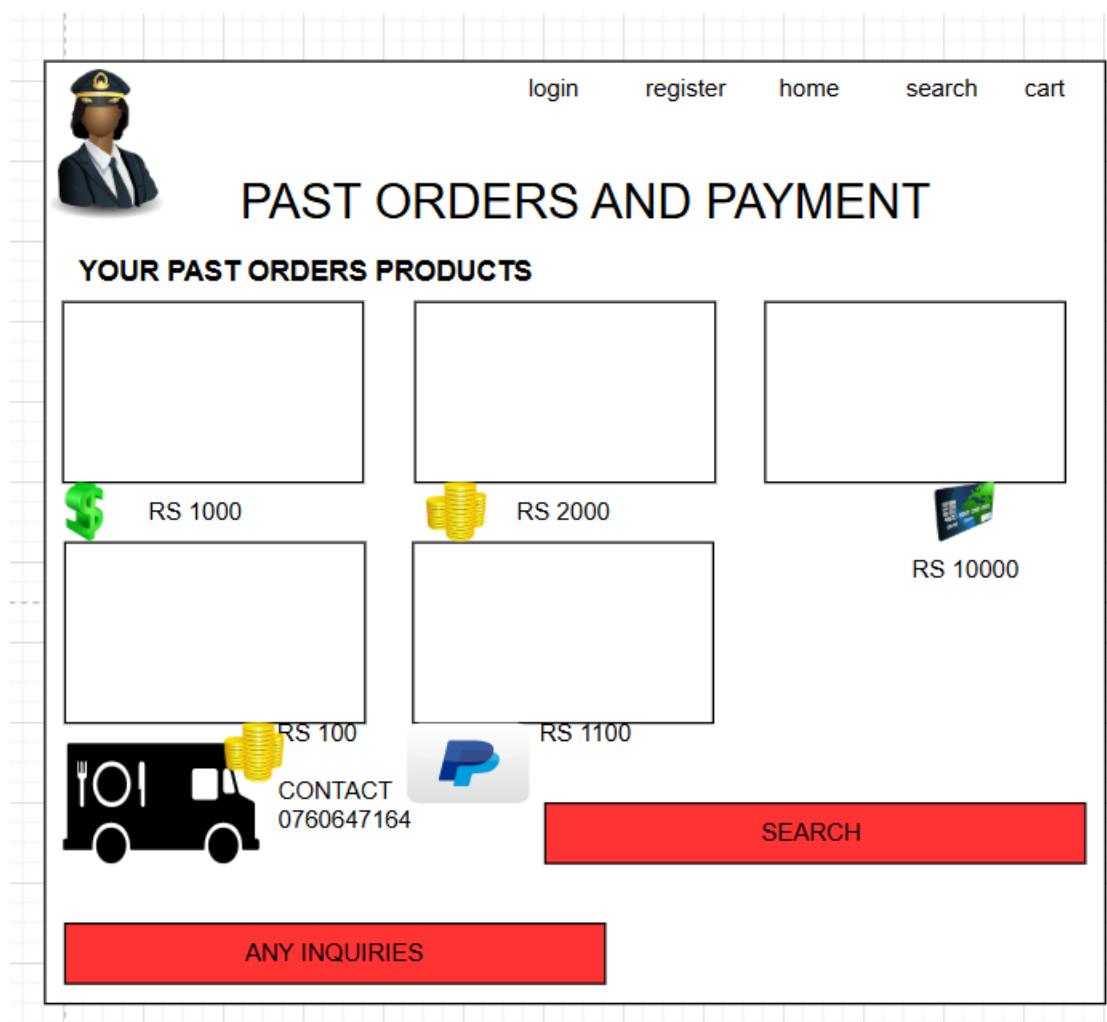
The Add to Cart function is an essential part of online shopping, designed to help users select and gather products they wish to purchase before proceeding to checkout. Typically, you will find an Add to Cart button located on or near the product image or description. When you click this button, the product is instantly added to your shopping cart, which is usually visible somewhere on the same page, such as in a sidebar or a floating icon.

Once an item is added, a small notification, confirmation message, or a checkmark

often appears to reassure you that the product has been successfully included in your cart. This feedback helps avoid confusion and ensures a smooth shopping experience.

The main purpose of the Add to Cart feature is to allow users to continue browsing and selecting multiple items without immediately committing to purchase each one. By collecting all desired products in a single cart, users can review their selections, adjust quantities, remove items, or apply discount codes before finally completing the purchase at checkout. This approach enhances convenience and efficiency, making the overall online shopping process more user-friendly and less time-consuming.

Past orders and payment



View your profile and order history.

After login, navigate to your Profile section.

Update your delivery address, view previous orders, or repeat purchases.

The User Profile and Past Orders feature allows you to manage your personal information and keep track of your previous purchases, enhancing your overall shopping experience. Once you have logged into your account, you can navigate to the Profile section, usually accessible from the main menu or a user icon at the top of the page.

In the Profile section, you can update important details such as your delivery address, contact information, and payment preferences. Keeping your information current ensures that your orders are delivered accurately and on time, preventing any delays or errors.

Additionally, the Past Orders area provides a detailed history of all your previous purchases. You can view order dates, items bought, prices, and order statuses. This helps you keep track of what you have bought and when, which is useful for budgeting, returns, or warranty claims.

One convenient function often included is the ability to quickly reorder items from your past purchases. This saves time if you want to buy the same products again without searching for them. While this feature might be optional on some platforms, it significantly improves user convenience and satisfaction by making shopping more personalized and efficient.

Release Plan

Milestones & Tools

Phase	Description	Tools Used
Requirements Analysis	Gather user and business requirements	Google Docs, Miro
Design	DFDs, ERD, Use Case, Flowcharts	Draw.io, Lucidchart
UI/UX Prototypes	Low to high fidelity interface	Figma, Adobe XD

	designs		
Development	Back-end and Front-end implementation	HTML/CSS, JavaScript, PHP orNode.js	
Testing	Functional, UI, and User Testing	Selenium, Postman	
Deployment	Launch MVP version	GitHub, Netlify/Vercel	
Maintenance & Update	Collect feedback, monitor, improve features	Google Forms, Jira	

Assessing the Effectiveness of the System Design with Reference to the Methodology Used and How the Design Meets User and System Requirements

The design of the Fresh Grocers web-based ordering system clearly shows how effective it is by using several important tools like the use case diagram, entity-relationship diagram (ERD), data flow diagram (DFD), and well-designed user interfaces. These tools are very useful because they support the SCRUM methodology, which was chosen for the system development. SCRUM is a flexible and iterative approach that allows the development team to work in small sprints, regularly get feedback from users and stakeholders, and improve the system step by step. This approach helps the system to stay adaptable and responsive to changing needs or problems that may arise during development.

Starting with the use case diagram, it plays a big role in understanding how the different types of users interact with the system. The Fresh Grocers platform has multiple user groups such as customers, delivery staff, and administrators. The use

case diagram helps to clearly show the tasks each user can do. For instance, customers can browse through different grocery categories, view product details like price and quantity, add items to their shopping cart, place orders, and even track the delivery status. Meanwhile, delivery staff can receive delivery assignments, update delivery status, and confirm order completion. Administrators or managers use the system to manage the product inventory, update stock levels, and oversee delivery schedules. This visualization helps make sure all the important functions that users need are included from the start. It also allows the team to focus on real user needs and adjust the system as feedback is gathered during SCRUM sprints. However, one limitation is that use case diagrams can sometimes oversimplify complex interactions or miss some edge cases, so the team has to complement them with other tools.

Next, the entity-relationship diagram (ERD) is essential because it defines how data is structured inside the system's database. The ERD shows entities such as customers, orders, products, categories, and delivery schedules, along with how they are related. For example, a customer can place multiple orders, each order contains multiple products, and each product belongs to a certain category. By clearly defining primary keys and foreign keys, the ERD helps to maintain database normalization which reduces data redundancy and improves data integrity. This is important because it ensures that information like customer profiles, order details, and stock levels are accurate and up-to-date. The ERD also supports efficient querying, which speeds up processes like order tracking and inventory management. Despite these advantages, creating a perfect ERD can be challenging if the system's requirements are constantly changing. In SCRUM, the team has to update the ERD iteratively as new features or changes are introduced, which can sometimes cause delays or inconsistencies if not managed carefully.

The user interfaces (UI) of the system are another key factor in making the Fresh Grocers platform successful. A good UI ensures that users have a pleasant and straightforward experience while interacting with the system. For customers, the interface is designed to be clean, simple, and responsive, so they can easily browse products by category, filter items based on preferences like price or brand, and see detailed information before adding items to their cart. The checkout process is made smooth to avoid confusion or cart abandonment, including clear steps for payment

and order confirmation. For staff and administrators, the UI provides functional dashboards that allow quick access to order lists, inventory updates, and delivery management tools. Because SCRUM promotes continuous feedback, the UI is improved regularly based on user testing and sprint reviews. This means the team can fix usability issues quickly and add features that users request. One drawback is that frequent UI changes can sometimes confuse users if not communicated well, so balancing improvements with user familiarity is important.

The data flow diagram (DFD) is useful in visualizing how data moves through the system from start to finish. It shows the flow of data between users, processes, data stores, and external systems. For example, when a customer places an order, the DFD illustrates how the order data is first validated, then processed through payment systems, followed by updates to inventory databases and delivery scheduling. This clear mapping helps the development team understand the system's internal workings and identify any bottlenecks or inefficiencies. For instance, if the payment processing step takes too long or fails, it can cause delays downstream. The DFD supports SCRUM's focus on continuous improvement by highlighting areas where optimization can be applied in future sprints. However, DFDs can become very complex for large systems, making them harder to maintain or communicate to all stakeholders, so they need to be kept clear and focused.

Together, these modeling tools and user interfaces work well within the SCRUM framework to build a system that meets both functional and non-functional requirements. SCRUM's iterative nature means the development team works in short cycles called sprints, each ending with a potentially shippable product increment. This allows the team to get regular feedback from users or stakeholders, adapt quickly to changes, and avoid long delays typical of traditional waterfall methods. The use case diagram guides feature development focused on real user needs, the ERD ensures data is correctly managed, the UI delivers a user-friendly experience, and the DFD provides insight into process flows. This combination helps the Fresh Grocers system deliver reliable service, fast order processing, and good customer satisfaction.

That said, there are some challenges with using SCRUM and these design tools. Since SCRUM encourages changes and new features based on feedback, it can lead to scope creep or unclear requirements if not managed well. Also, continuously

updating diagrams like the ERD or DFD during development requires discipline and good communication within the team. If the documentation falls behind, it can cause confusion or errors. Finally, balancing quick development with thorough testing is essential to avoid bugs or performance issues.

Fresh Grocers web-based ordering system's design demonstrates its effectiveness by integrating key tools like the use case diagram, ERD, DFD, and user interfaces under the SCRUM methodology. These tools ensure that user needs are captured, data is well structured, processes are transparent, and users have a positive experience. SCRUM's iterative process supports ongoing improvements and adaptability, which are vital for a modern e-commerce platform where customer expectations and technology change rapidly. While there are some risks and challenges in maintaining documentation and managing changing requirements, the benefits of this approach help Fresh Grocers provide a high-quality grocery delivery service that satisfies customers and supports business growth.

Conclusion

The decision by Fresh Grocers to implement a web-based ordering system represents a significant leap forward in adapting to the rapidly changing landscape of the grocery retail industry. By harnessing the power of modern information technology, the company aims to enhance customer convenience, streamline operations, and gain a competitive advantage in the market. The introduction of this system underscores Fresh Grocers' commitment to meeting the increasing demands of customers who prefer to shop online and use mobile platforms for their grocery needs. With features that allow customers to browse products, place orders online, and receive real-time updates, Fresh Grocers is positioned to offer a seamless and intuitive shopping experience that aligns with contemporary consumer expectations.

The integration of a customer review and rating system enables shoppers to share their experiences and offer feedback, fostering transparency and accountability within the service. This not only improves customer satisfaction but also motivates suppliers and delivery personnel to uphold high standards of service quality. Additionally, the option for customers to place orders over the phone for those with limited access to digital platforms demonstrates Fresh Grocers' dedication to inclusivity and ensuring accessibility for all users. By providing multiple ordering channels, the company ensures that customers can enjoy a convenient shopping experience regardless of their technological proficiency.

The system's automated inventory management and order fulfillment features significantly enhance operational efficiency by streamlining product availability tracking and order processing. By optimizing these processes, Fresh Grocers can reduce fulfillment times, minimize stockouts, and ensure faster delivery, improving overall service reliability. The focus on capturing real-time data for inventory management and order updates further strengthens the company's commitment to operational excellence and customer satisfaction.

In conclusion, the implementation of a web-based ordering system by Fresh Grocers represents a strategic investment in modernizing operations, enhancing the customer experience, and staying ahead of the competition. With a strong emphasis on customer-centric features and seamless integration of technology, Fresh Grocers is well-positioned to thrive in the competitive and dynamic grocery retail market. The company's commitment to providing an accessible, reliable, and efficient shopping experience sets the stage for sustained growth and customer loyalty in the evolving market landscape.

Gantt chart

	March/06				March/31				April/05				May /16	
Activity 1														
Activity 2														
Activity 3														
Activity 4														

References

References

agilealliance, 2020. *agilealliance*. [Online]
Available at: <https://www.agilealliance.org/glossary/xp/>
[Accessed 14 04 2025].

- Almeida, J., 2025. *distantjob*. [Online]
Available at: <https://distantjob.com/blog/iterative-development/>
[Accessed 28 04 2025].
- Anon., 2025. *tutorialspoint*. [Online]
Available at: https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm
[Accessed 03 05 2025].
- Anon., n.d. *gwentechembedded*. [Online]
Available at: <http://gwentechembedded.com/five-stages-of-a-software-development-life-cycle/>
[Accessed 22 04 2025].
- Anon., n.d. *javatpoint*. [Online]
Available at: <https://www.javatpoint.com/software-engineering-software-development-life-cycle>
[Accessed 22 04 2025].
- atlassian, 2025. *atlassian*. [Online]
Available at: <https://www.atlassian.com/agile>
[Accessed 24 04 2025].
- BWC, A., 2019. *bigwater*. [Online]
Available at: <https://bigwater.consulting/2019/04/08/software-development-life-cycle-sdlc/>
[Accessed 19 04 2025].
- CFI Team, 2025. *corporatefinanceinstitute*. [Online]
Available at: <https://corporatefinanceinstitute.com/resources/management/feasibility-study/>
[Accessed 13 04 2025].
- Dileep, R., 2025. *knowledgehut*. [Online]
Available at: <https://www.knowledgehut.com/blog/agile/pros-and-cons-scaled-agile-framework#the-hurricane-effect:%C2%A0%C2%A0>
[Accessed 19 04 2025].
- Ghosh, A., n.d. *ellow*. [Online]
Available at: <https://ellow.io/advantages-and-disadvantages-of-sdlc/>
[Accessed 29 04 2025].
- indeed, 2025. *indeed*. [Online]
Available at: <https://www.indeed.com/career-advice/career-development/what-is-system->

analysis

[Accessed 30 03 2025].

Laoyan, S., 2025. *asana*. [Online]

Available at: <https://asana.com/resources/agile-methodology>

[Accessed 22 04 2025].

Martin, M., 2025. *guru99*. [Online]

Available at: <https://www.guru99.com/software-engineering-prototyping-model.html>

[Accessed 24 04 2025].

nearshore-it, 2025. *nearshore-it*. [Online]

Available at: <https://www.nearshore-it.eu/articles/project-management-leadership/extreme-programming-complete-guide/>

[Accessed 14 04 2025].

pm-partners, 2025. *pm-partners*. [Online]

Available at: <https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview/>

[Accessed 25 04 2025].

productplan, 2025. *productplan*. [Online]

Available at: <https://www.productplan.com/glossary/lean-software-development/>

[Accessed 14. 04 2025].

productplan, 2025. *productplan*. [Online]

Available at: <https://www.productplan.com/glossary/scaled-agile-framework/>

[Accessed 17 04 2025].

productplan, 2025. *productplan*. [Online]

Available at: <https://www.productplan.com/glossary/disciplined-agile/>

[Accessed 10 04 2025].

Quick, L., 2025. *knowledgehut*. [Online]

Available at: <https://www.knowledgehut.com/blog/agile/sdlc-vs-agile>

[Accessed 04 04 2025].

Raghavan, R., 2025. *acodez*. [Online]

Available at: https://acodez.in/lean-software-development/#Advantages_and_Disadvantages_of_Lean_Software_Development

[Accessed 18 04 2025].

Sami, M., 2012. Software Development Life Cycle Models and Methodologies. 24 04.

sanyajgy, 2025. *geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/system-analysis-system-design/>

[Accessed 30 04 2025].

Satyabrata_Jena, 2025. *geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/>

[Accessed 13 04 2025].

saylor, 2025. *saylor*. [Online]

Available at:

<https://learn.saylor.org/mod/book/view.php?id=33038#:~:text=There%20are%20seven%20techniques%20we,documents%2C%20and%20review%20of%20software.>

[Accessed 17 04 2025].

senwork.com, 2024. *feasibility-conceptual-study/*. [Online]

Available at: <https://senwork.com/engineering/feasibility-conceptual-study/>

[Accessed 22 04 2025].

simplilearn, 2025. *simplilearn*. [Online]

Available at: <https://www.simplilearn.com/tutorials/programming-tutorial/what-is-software>

[Accessed 24 04 2025].

synopsys, 2025. *synopsys*. [Online]

Available at: <https://www.synopsys.com/glossary/what-is-sdlc.html>

[Accessed 30 04 2025].

tutorialspoint, 2025. *tutorialspoint*. [Online]

Available at: https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm

[Accessed 23 04 2025].

visuresolutions, 2025. *visuresolutions*. [Online]

Available at: <https://visuresolutions.com/blog/requirements-definition/>

[Accessed 10 04 2025].