

Online Pizza Delivery System

Abstract

The **Online Pizza Delivery System** is a web-based application designed to simplify the process of ordering pizza online for customers and managing orders efficiently for vendors and administrators. The primary goal of this system is to provide a digital platform where customers can browse pizza items, place orders, track order status, and give feedback, while vendors can manage products, orders, and customer interactions effectively. The admin controls the overall system by managing users, categories, notifications, and feedback.

This system reduces manual work, minimizes errors, and enhances customer satisfaction by offering a fast and user-friendly ordering experience. The scope of this project includes online ordering, bulk orders, cart management, notifications, and role-based access for Admin, Vendor, and Customer. The Online Pizza Delivery System ensures efficiency, transparency, and better coordination between customers and vendors.

Introduction

With the rapid growth of internet technology and digital services, online food ordering systems have become an essential part of modern lifestyles. Customers prefer online platforms to order food conveniently without visiting physical restaurants. Traditional pizza ordering methods such as phone calls or in-person visits are time-consuming and error-prone.

The **Online Pizza Delivery System** is developed to overcome these limitations by providing a centralized web-based solution. Customers can easily browse pizza menus, add items to the cart, place orders, and track them online. Vendors can manage their pizza products, view customer orders, and handle bulk orders efficiently.

The admin ensures smooth functioning of the system by monitoring users, categories, feedback, and notifications.

Scope and Objective

Scope:

The scope of the **Online Pizza Delivery System** includes:

- Online pizza ordering and bulk orders
- Role-based access for Admin, Vendor, and Customer
- Cart and address management
- Feedback and notification system
- Secure user authentication

The system can be expanded in the future with online payment integration, live order tracking, and mobile application support.

Objectives:

1. To provide a user-friendly online pizza ordering platform
2. To reduce manual work and order errors
3. To improve customer experience and satisfaction
4. To help vendors manage orders efficiently
5. To maintain centralized control through admin

Literature Survey

Several studies and existing systems have been reviewed to understand the current trends and limitations in online food and pizza delivery systems:

1. **Traditional Pizza Ordering Systems**– Traditional systems rely on phone calls or in-person ordering at restaurants. These methods involve manual order taking, which often leads to errors in order details, delays in processing, and

lack of proper order tracking. Customers do not receive real-time updates, and vendors face difficulties in managing multiple orders efficiently.

2. **Restaurant-Specific Online Ordering Websites**– Some restaurants offer their own basic online ordering platforms. However, many of these systems lack advanced features such as cart management, bulk order handling, feedback systems, and real-time notifications. User experience is often limited, and system scalability is poor.
3. **Third-Party Food Delivery Applications** Popular food delivery applications provide online ordering and tracking, but they charge high commission fees to vendors. Small vendors face difficulties in managing their products independently, and customization options are limited. Admin-level control and direct vendor–customer interaction are also restricted.
4. **Gaps Identified** From the study of existing systems, a clear need is identified for a centralized, cost-effective, and user-friendly online pizza delivery system. The system should support multiple roles such as Admin, Vendor, and Customer, provide efficient order and bulk order management, real-time notifications, feedback handling, and full administrative control to ensure smooth and transparent operations.

Existing System

In the existing system, pizza ordering is mostly done through phone calls or walk-in visits to the restaurant. Orders are noted manually, leading to possible mistakes in order details, pricing, or delivery addresses. Order tracking is difficult, and customers do not receive real-time updates.

Vendors manually manage orders and customer details, which increases workload and delays. There is no centralized system for managing feedback, notifications, or bulk orders.

Drawbacks of Existing System

1. Manual order processing leads to errors
2. No real-time order tracking for customers
3. Time-consuming ordering process

4. Difficult to manage bulk orders
5. Poor customer interaction and feedback management
6. High dependency on staff availability

Proposed System

The proposed **Online Pizza Delivery System** is a fully automated web application that allows customers to order pizza online easily. The system provides separate modules for Admin, Vendor, and Customer, each with specific functionalities.

Customers can register, browse pizzas, add items to the cart, place orders, and track their order status. Vendors can manage pizza items, view orders, and process bulk orders. Admins supervise the entire system by managing users, categories, notifications, and feedback.

Key highlights:

- Fully online pizza ordering system that allows customers to browse menus, add items to cart, and place orders anytime and anywhere.
- Real-time order management and notifications that keep customers, vendors, and admins updated about order status.
- Secure user authentication with role-based access for Admin, Vendor, and Customer modules.
- Centralized product and category management for vendors to easily add, update, and manage pizza items.
- Efficient bulk order handling to support large or group orders without manual intervention.
- Admin supervision for monitoring users, orders, feedback, and overall system performance.

- Integrated feedback and review system to improve service quality and customer satisfaction.
- Reduced manual workload and errors through automated order processing and data management.

Advantages of Proposed System

1. Faster and easier ordering process
2. Real-time notifications and order updates
3. Reduced manual work and errors
4. Better management of bulk orders
5. Improved customer satisfaction
6. Centralized and secure data management

Modules:

Admin Module

The Admin module manages the entire application. Admin can log in, view customers and vendors, manage categories, add notifications, view feedback, and monitor system activities. Admin ensures data security and smooth system operation.

Vendor Module

The Vendor module allows vendors to manage pizza products and orders. Vendors can add and view products, manage customer orders and bulk orders, post blogs, view notifications, and update profiles.

Customer Module

The Customer module enables users to register, browse pizza items, manage cart, place orders, view order history, add feedback, manage delivery addresses, and receive notifications.

Software Requirements

- Operating System : Windows
- User Interface : HTML, CSS , JS and Bootstrap
- Programming Language : Java
- Web Applications : JSP JDBC
- IDE/Workbench : Eclipse
- Database : MYSQL, SQL, SQLYOG
- Server Deployment : Apache Tomcat

Hardware Requirements

- Processor : Intel I3 Processor
- RAM : 4 GB
- Monitor : 15 inch color monitor or LED
- Hard disk : 256 GB
- Key board : Standard 102 keys
- Mouse : Optical

Functional Requirements:

Home

About

Contact

Admin:

➤ Admin Functions

- Login
- Home
- View Contacts
- View Customers
- View Vendors

- Add Notification
- View Notifications
- Add Category
- View Categories
- View Feedbacks
- Change Password
- Logout

Customer:

➤ Customer Functions

- Register
- Login
- Home
- View Notifications
- View Blogs
- View Products
- My Orders
- View Bulk Orders
- Add Feedback
- View Feedbacks
- Add Address
- My Cart
- My Profile
- Change Password
- Logout

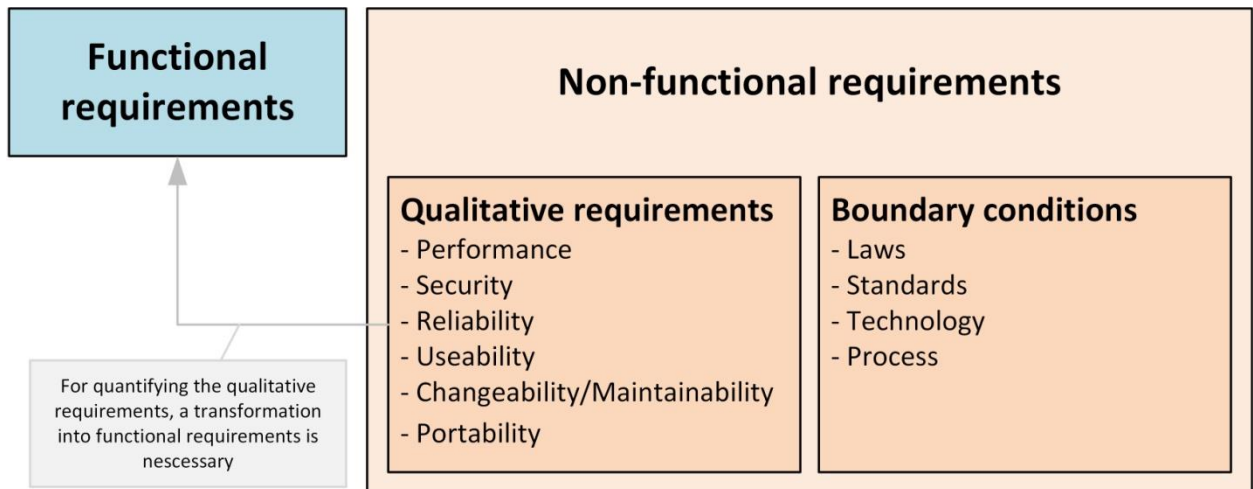
Vendor:

➤ Vendor Functions

- Login
- Home
- View Notification
- Add Blog
- View Blogs
- Add Product
- View Products
- View Orders
- View Bulk Orders
- Add Feedback
- View Feedbacks
- My Profile
- Change Password
- Logout

Non Functional Requirements:

Non functional testing is a type of software testing that verifies non functional aspects of the product, such as performance, stability, and usability. Whereas functional testing verifies whether or not the product does what it is supposed to, non functional testing verifies how well the product performs.



The system should shows below non functional requirements

1. High Security
2. Multi browser and operating system compatibility
3. Mobile responsiveness
4. High quality and performance
5. Memory optimized and User Friendliness
6. Easy maintenance and scalable capability
7. It's should support good accessibility and usability
8. It should follow regulatory and compliance policies

Feasibility Study:

The possibility of the undertaking is broke down during this stage and strategic plan is advanced with an exceptionally broad arrangement for the task and a few quotes. During project examination the likelihood investigation of the proposed project is to be completed. this is often to ensure that the proposed project isn't a weight to the organization. For practicality examination, some comprehension of the many necessities for the project is prime.

After the approval of the request to the organization and project guide, with an investigation being considered, the project request must be examined to determine precisely what the system requires.

Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, its cost, priority, completion time and personnel requirement is estimated and used to determine where to add it to any project list. Truly speaking, the approval of those above factors, development works can be launched

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are.

ECONOMICAL FEASIBILITY

This examination is completed to see the financial effect that the framework will wear the association. The measure of store that the organization can fill the innovative work of the framework is constrained. The consumptions must be advocated. Accordingly, the created framework also inside the budget and this was accomplished in light of the very fact that an outsized portion of the advancements utilized are uninhibitedly accessible. Just the modified items must be bought. Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. And also in this project work all software is free and open source. It can be developed using our personal machines. So its economically its feasible to complete the project.

TECHNICAL FEASIBILITY

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The system is developed for platform Independent environment. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

This investigation is completed to see the technical practicality, that is, the specialized prerequisites of the framework. Any framework created must not have a popularity on the accessible specialized assets. this may prompt high requests on the accessible specialized assets. this may prompt high requests being assail the customer. The created framework must have a humble prerequisite, as just insignificant or invalid changes are required for actualizing this framework.

SOCIAL FEASIBILITY

The a part of study is to see the degree of acknowledgment of the framework by the client. This incorporates the way toward preparing the client to utilize the framework effectively. The client must not feel undermined by the framework, rather should acknowledge it as a requirement . The degree of acknowledgment by the clients exclusively relies upon the techniques that are utilized to show the client about the framework and to form him familiar with it. His degree of certainty must be raised with the goal that he's likewise able to make some helpful analysis, which is invited, as he's the last client of the framework.

OPERATIONAL FEASIBILITY

This region of attainability ensures that the undertaking fulfills all the prerequisites that were talks about during introductory necessity examination stage. This undertaking met all the necessities proposed before. It ensured that the modules examined during the proposing of the framework were actualized with stable precision and with no mistakes.

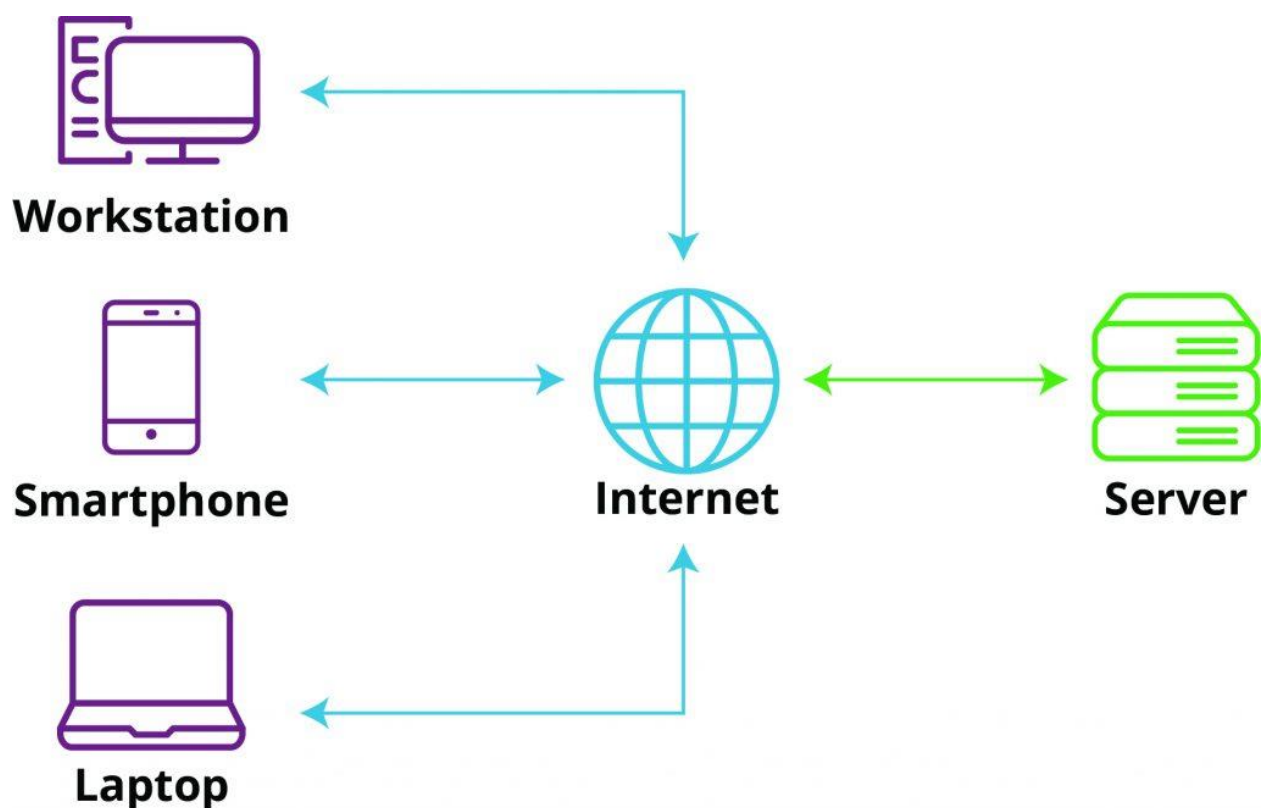
Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

Client Server Architecture

Client server architecture, architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host

computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns.

The Client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and delivers the data packets requested back to the client. Clients do not share any of their resources. Examples of the Client-Server Model are Email, World Wide Web, etc.



Web Application MVC Architecture

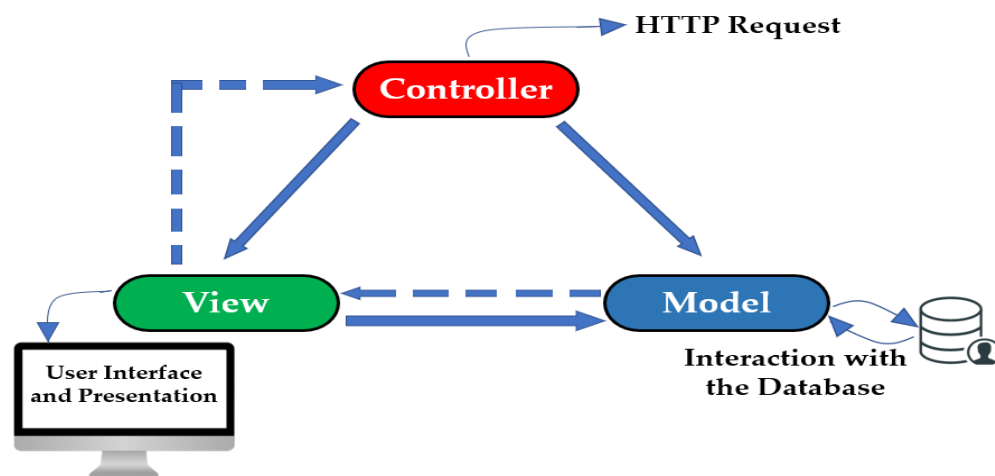
The Model-View-Controller (MVC) is a well-known design pattern in the web development field. It is a way to organize our code. It specifies that a program or application shall consist of data model, presentation information and control

information. The MVC pattern needs all these components to be separated as different objects.

The MVC pattern architecture consists of three layers:

- **Model:** It represents the business layer of application. It is an object to carry the data that can also contain the logic to update controller if data is changed.
- **View:** It represents the presentation layer of application. It is used to visualize the data that the model contains.
- **Controller:** It works on both the model and view. It is used to manage the flow of application, i.e. data flow in the model object and to update the view whenever data is changed.

Model-View-Controller Architecture Block Diagram



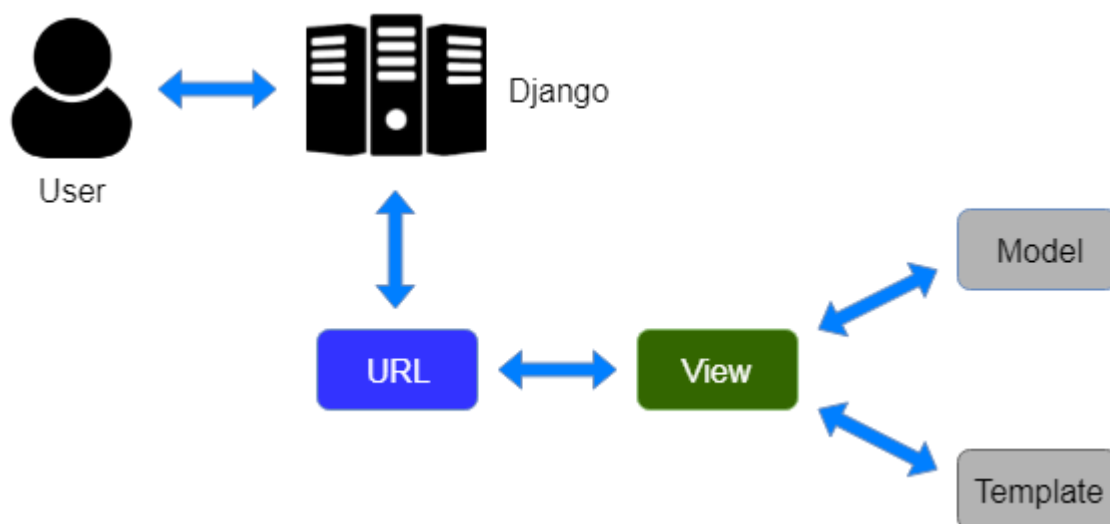
Web Application MVT Architecture

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data.

The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template.

The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres).

The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript and Jinja files.



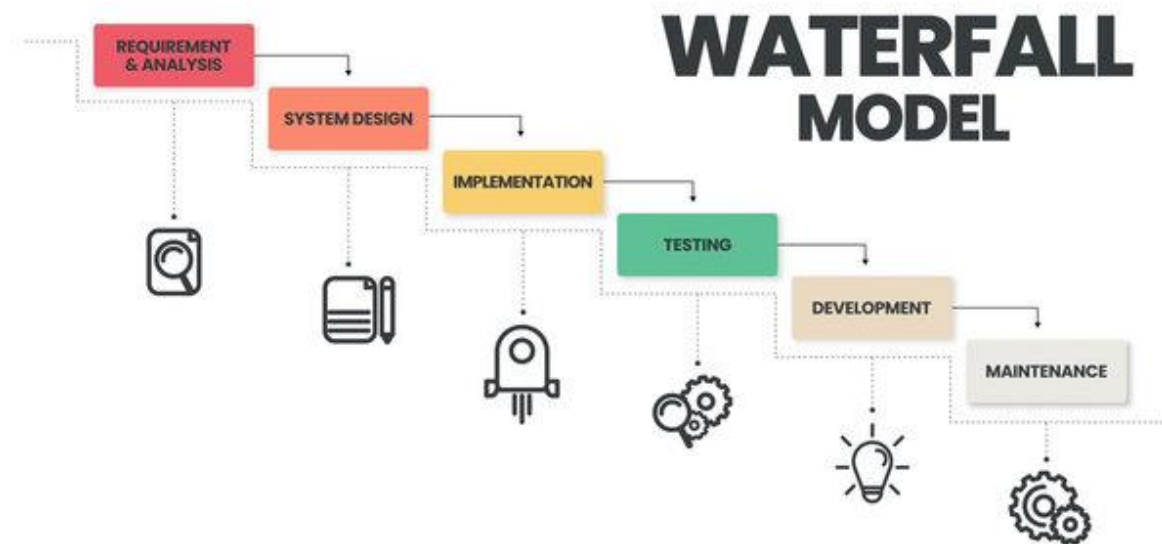
Process Models

1. Waterfall Model:

Waterfall model indicates that all the SDLC steps will be going on step by step. In this model once one stage completed then only it will start the another process. This will be implemented when the requirements well known and clearly defined.

In this model we can implement all the steps one by one. First we will do requirements gathering i.e in this step business analyst will collect the all functional requirements from the client and will discuss with the manager and development team. Second step is analysis where people can analyses the requirements so that they will do risk

basement, risk mitigation, risk controlling steps. And also they will do the cost-time analysis to estimate the cost. They will also do the detail feasible study over the requirements of project.



In third phase called design page generally people will design the project i.e architect will prepare the UML diagrams and prototypes for project. And then developers will start programming or coding based on design . Finally testing team will test the content to identify and fix the bugs. After successful user acceptance testing application will be deployed and maintained in the client machine.

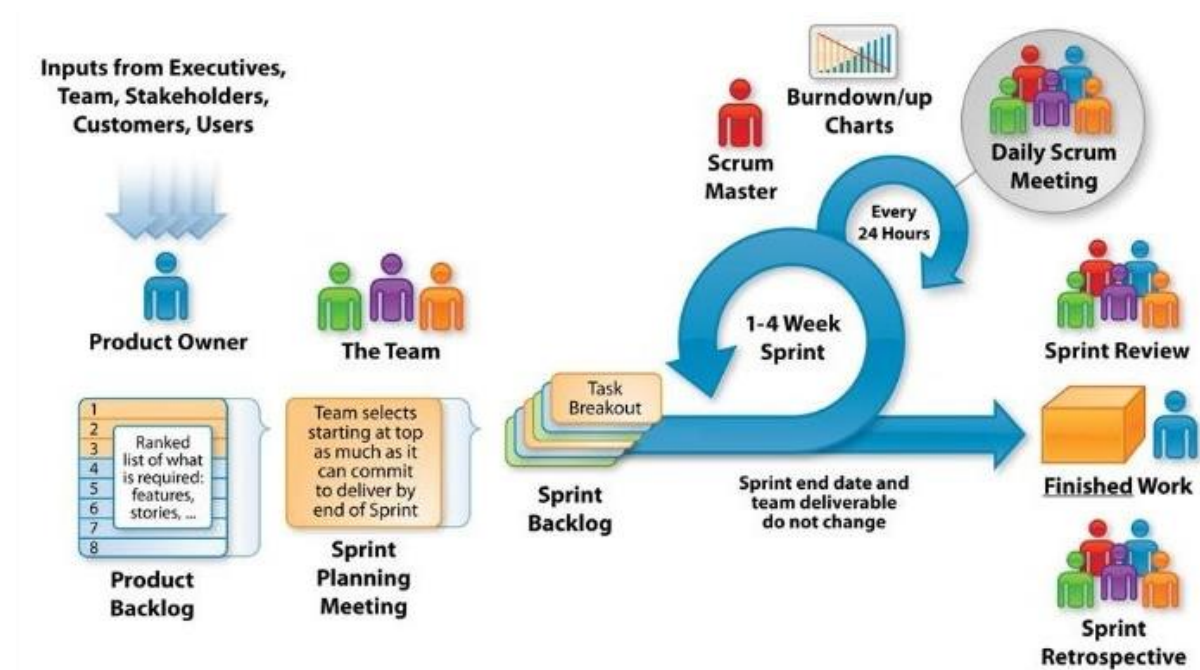
The waterfall model was selected as the SDLC model due to the following reasons:

- When the requirements are very clear and not changed.
- Available technology is enough to implement.
- Project is easy and simple to understand to finish.
- And if we don't have any confusion in requirements.
- We can easily manage the review process and task management.
- Every stage defined clearly and with timelines

2. Agile Scrum Model:

Agile scrum methodology is the combination of the agile philosophy and the scrum framework. Agile means “incremental, allowing teams to develop projects in small

increments. Scrum is one of the many types of agile methodology, known for breaking projects down into sizable chunks called “sprints.” Agile scrum methodology is good for businesses that need to finish specific projects quickly.



Agile scrum methodology is a project management system that relies on incremental development. Each iteration consists of two- to four-week sprints, where the goal of each sprint is to build the most important features first and come out with a potentially deliverable product. More features are built into the product in subsequent sprints and are adjusted based on stakeholder and customer feedback between sprints. Whereas other project management methods emphasize building an entire product in one operation from start to finish, agile scrum methodology focuses on delivering several iterations of a product to provide stakeholders with the highest business value in the least amount of time.

Agile scrum methodology has several benefits. First, it encourages products to be built faster, since each set of goals must be completed within each sprint's time frame. It also requires frequent planning and goal setting, which helps the scrum team focus on the current sprint's objectives and increase productivity.

The greatest benefit of agile scrum methodology is its flexibility. With the sprint-based model, the scrum team typically receives feedback from stakeholders after each sprint. If there are any problems or changes, the scrum team can easily and quickly adjust product goals during future sprints to provide more valuable iterations. This way,

stakeholders are happier because they get exactly what they want after being involved every step of the way.

Conclusion

The **Online Pizza Delivery System** is an efficient and reliable solution that modernizes the traditional pizza ordering process. It provides convenience to customers, improves order handling for vendors, and ensures effective system control for administrators. By reducing manual efforts and enhancing transparency, the system delivers a smooth and satisfying food ordering experience. This project demonstrates how web-based applications can improve service quality, operational efficiency, and customer engagement in the food delivery industry.