PARUL UNIVERSITY
FACULTY OF ENGINEERING
SOFTWARE ENGINEERING (303105254)
B-Tech 2nd YEAR

ONLINE FOOD DELIVERY

PRACTICAL-1

AIM : Project Defination and Objective of modules and perform Requirement Engineering process.

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

Defination:

It is a process where business delivery customers order through an online platform such as mobile applications, websites, social media to the address intended by the customers. It refers to the interaction of ordering food and prepared meals online. Orders are typically placed through an app ,website and delivery times vary.

Objective:

The primary objective of the online food delivery system project is to create a seamless and efficient platform that connects customers with a variety of restaurants, enabling them to browse menus, place orders, and receive food deliveries at their doorstep. The system aims to enhance the overall food ordering experience for users by leveraging technology to streamline the process, improve accessibility, and provide a convenient solution for both customers and restaurants.

FUNCTIONS:

- Module-1: Restaurant Management
- Module-2: Delivery Management
- Module-3: Business Operations and Analytics

Module-1:

Restaurant Management:

The Admin (Restaurant Manager) oversees restaurant-related activities, including registration, menu configuration, and availability settings. The admin verifies and manages menus, handles user interactions, and manages order processing. They assign delivery personnel, confirm order completion, and address cancellations and refunds. Feedback from users is managed, and the admin implements promotions and analytics for strategic business operations. This role ensures the smooth operation of the restaurant ecosystem within the online food delivery system.siness Operations and Analytics. The restaurant shows as follows

1.Name of the restaurant
2.Area of the retaurent
3.What type of food and which food they are serving
4.Delivery Timings and Limits

Module-2:

Delivery Management:

The Admin (Delivery Manager) optimizes delivery logistics, ensuring timely and accurate order completion. They employ real-time communication channels to keep users informed about their delivery status. Admin resolves any issues, such as address discrepancies or delays, ensuring a positive user experience. Additionally, the manager analyzes delivery data to enhance overall efficiency and streamline the delivery process further. This role is integral to providing a reliable and customer-centric online food delivery service.

Module-3:

Business Operations and Analytics:

The Admin (Operations Analyst) strategically manages business operations, implementing targeted promotions and discounts to attract and retain customers. Through in-depth analytics, they assess sales performance, analyze customer behavior patterns, and identify popular menu items. Admin leverages these insights to make informed decisions that optimize revenue and enhance the overall user experience. This role is instrumental in maintaining a competitive edge in the market and fostering sustained growth for the online food delivery system.

REQUIREMENTS:

Conducting online food delivery operations necessitates careful consideration of various factors to ensure efficiency, reliability, and customer satisfaction. Drawing parallels with the requirements for online elections, the following considerations are crucial:

1. Security:

- Secure Infrastructure: Ensure a robust and secure platform for handling online orders and transactions.

Encryption: Implement encryption protocols to safeguard customer and transaction data.

Authentication: Establish secure authentication mechanisms for users, restaurants, and delivery personnel.

Auditability: Incorporate audit trails to track and verify order processing, ensuring transparency.

2. Accessibility:

- User-Friendly Interface: Design an intuitive and user-friendly interface for seamless order placement.

Multi-Language Support: Provide multi-language support to cater to a diverse user base.

Device Compatibility: Ensure compatibility across various devices, including PCs, smartphones, and tablets, for convenient access.

3. Privacy:

- Anonymity: Protect user privacy by ensuring order details are confidential.

Data Protection: Implement robust data protection measures to secure user information.

Limited Data Retention: Adhere to data retention policies, retaining only necessary information.

Verification and Auditability: Establish mechanisms for verifying order accuracy and providing customers with confirmations.

4. Contingency Plans:

- Backup Systems: Implement redundant systems to ensure continuous order processing in case of technical failures.

Response Protocols: Define clear protocols to address technical issues, disruptions, or cyber threats during order fulfillment.

Communication Plan: Develop a comprehensive communication plan to inform users and stakeholders about disruptions and the steps taken to address them.

By adapting these requirements, an online food delivery system can prioritize security, accessibility, privacy, and contingency planning, ensuring a reliable and transparent service for both users and stakeholders.

PARUL UNIVERSITY
FACULTY OF ENGINEERING
SOFTWARE ENGINEERING (303105254)
B-Tech 2nd YEAR

ONLINE FOOD DELIVERY

PRACTICAL-2

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

AIM : Identify sustainable design and implementation model from different software engineering models

Agile software:-

Agile software development, known for its iterative and flexible approach, is well-suited for online food delivery systems. Applying Agile principles enables developers to adapt to changing requirements, collaborate with stakeholders, and incorporate continuous customer feedback. In the context of food delivery, the Agile model supports quick and incremental releases, ensuring timely delivery of key features and allowing for continuous improvement based on user input. The customer-centric focus of Agile aligns with the dynamic demands of the food industry, addressing unique challenges and enhancing the responsiveness of the online food delivery system.

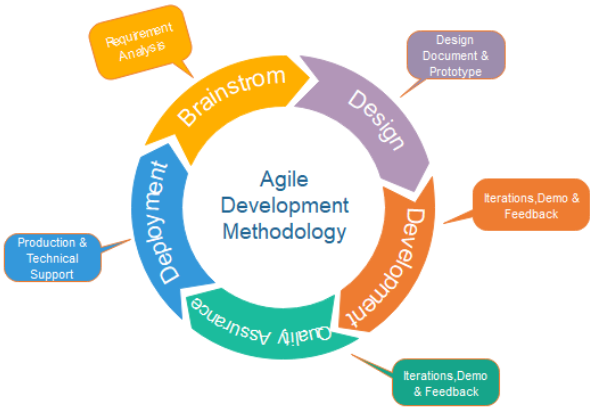


Fig. Agile Model

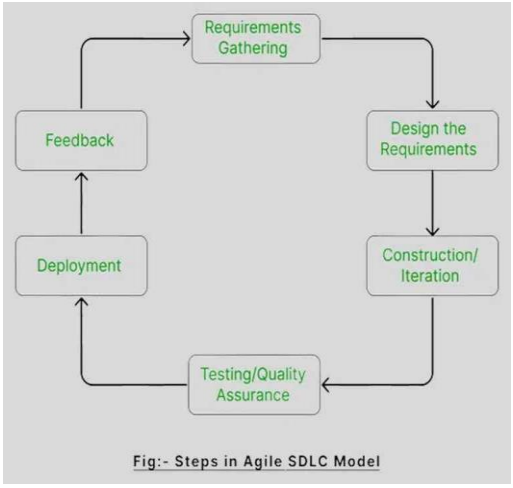


Fig:- Steps in Agile SDLC Model

JUSTIFICATION FOR CHOOSING AGILE MODEL FOR MY PROJECT: -

1.Learning Experience: Agile emphasizes collaboration, adaptability, and continuous improvement. This aligns well with the educational goals of learning through hands-on experience and iterative development.

2.Flexibility: As a student, the project requirements may evolve, and the Agile Model allows for flexibility in accommodating changes even late in the development process. This flexibility is valuable for refining the project based on feedback and new insights.

3.Incremental Progress: Agile encourages incremental development, allowing me to deliver functional components of the system in stages. This approach is suitable for a student project with limited time and resources.

4.Continuous Feedback: Agile promotes regular interactions with stakeholders (in this case, teachers or classmates), facilitating continuous feedback. This iterative feedback loop is beneficial for making improvements and ensuring the project meets expectations.

5.Adaptation to Learning: Since the primary goal is educational, the Agile Model allows me to adapt and learn from each iteration, making adjustments based on lessons learned in the process.

JUSTIFICATION FOR NOT CHOOSING OTHER MODELS:

1. Waterfall Model:

The Waterfall model's rigid sequential structure may not suit the dynamic nature of online food delivery, where requirements and customer preferences can evolve rapidly. Its lack of flexibility makes it challenging to accommodate changes once the development process has started.

2. Spiral Model:

The Spiral model, while addressing risk through iterative development, tends to be more complex and resource-intensive. Given the often-limited resources and scope of a student project, a simpler model may be more practical.

3. V Model:

The V Model's strict correlation between development phases and testing may lead to delays in receiving feedback, hindering the agile adjustments needed for an evolving online food delivery system. Its linear nature may not align well with the need for continuous adaptation.

4. Incremental Model:

While the Incremental model promotes partial implementation and testing, it may result in integration issues if not carefully planned. The dynamic nature of online food delivery may require a more adaptive and collaborative approach, which Agile better provides.

ONLINE FOOD DELIVERY

PRACTICAL-3

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

AIM: Prepare Software Requirement Specification (SRS) for the selected module.

Software Requirement Specification:

The Software Requirements Specification (SRS) is a comprehensive document guiding the development of a software system. It starts with an introduction, outlines functional and non-functional requirements, details system architecture and external interfaces, and describes the user interface and database design. Testing and security requirements are specified, and documentation needs are outlined. The SRS serves as a foundational reference, ensuring a common understanding among stakeholders and guiding the development team throughout the software development lifecycle.

Functional requirements

Requirements:

1.Registrations:

1.Restaurents

2.Delivery boys

3.Customers

1.1 Admin Register

Input = User id and password

Output = Save the details

1.2 User register

Input = user id and password

Output = login

1.3 Booking the order

Input = Enter the food item

Select area

Select hotel

Output = order placed

1.4 Order cancellation

Input = select reason for cancellation

Output = order cancelled

Output = Give feedback

2.Non function Readability:

2.1. Reliability:

Ensure that the system consistently provides accurate and reliable information to users.

Minimize service downtime and disruptions to maintain continuous availability for order placement and tracking**.**

2.2 Scalability:

Design the system to handle variable loads and user traffic, especially during peak hours.

Ensure that the platform scales seamlessly to accommodate growth in the user base and transaction volume.

2.3. Robustness:

Implement error-handling mechanisms to gracefully handle unexpected situations, preventing system crashes or data loss.

Regularly conduct stress testing to identify potential weaknesses and enhance the system's robustness.

2.4. Performance Efficiency:

Optimize system performance to ensure quick response times for order processing and real-time tracking.

Monitor and manage system resources efficiently to prevent performance bottlenecks.

2.5. Data Security:

Employ robust security measures to safeguard user data, ensuring confidentiality and integrity.

Implement encryption protocols for data transmission and storage to mitigate security risks.

2.6. Availability:

Guarantee high system availability by minimizing planned downtime for maintenance activities.

Implement redundant systems and backup mechanisms to ensure uninterrupted service.

2.7. Fault Tolerance:

Design the system to gracefully handle faults and recover quickly from unexpected failures.

Implement redundancy and failover mechanisms to maintain essential functionalities in case of system issues.

2.8. Load Balancing:

Employ load balancing techniques to distribute user requests evenly across servers.

Ensure optimal resource utilization and prevent overloading of specific components

3.External requirements:

3.1 Software Interface

3.1.1 Programming Language:

Python: Utilize Python for its versatility, readability, and a wide range of libraries suitable for web development.

3.1.2. Web Framework:

Django: Leverage Django as the web framework for its robust features, including an ORM, built-in authentication, and an admin interface.

3.1.3. Database Management:

PostgreSQL: Choose PostgreSQL as the relational database management system due to its reliability and support for complex data queries.

3.1.4. Frontend Development:

HTML, CSS, JavaScript: Use standard web technologies for frontend development, ensuring a responsive and interactive user interface.

Bootstrap: Implement the Bootstrap framework to enhance the visual appeal and responsiveness of the user interface.

3.1.5. Payment Integration:

Stripe API: Integrate the Stripe payment gateway using its Python API for secure and seamless transactions.

3.1.6. Real-Time Updates:

Django Channels: Employ Django Channels for real-time updates, especially for features like live order tracking.

3.1.7. Geolocation Services:

Google Maps API: Integrate the Google Maps API for location-based services, facilitating real-time order tracking and accurate restaurant locations.

3.1.8. Version Control:

Git: Use Git for version control, allowing collaborative development and easy management of code changes.

3.1.9. Deployment:

Docker: Utilize Docker for containerization, simplifying the deployment process and ensuring consistency across different environments.

Heroku: Deploy the application on Heroku for its ease of use and scalability.

3.1.10. Testing:

pytest: Implement the pytest framework for unit testing to ensure the reliability and correctness of code.

3.1.11. Security:

SSL/TLS: Enforce secure communication through SSL/TLS protocols to protect user data during transmission.

Django Security Best Practices: Adhere to Django's security best practices to mitigate vulnerabilities.

3.1.12. Documentation:

Sphinx: Use Sphinx for documenting the codebase, APIs, and project structure to facilitate collaboration and future maintenance.

3.2: Hardware Interface: -

3.2.1. Server Hardware:

Processor (CPU): A multi-core processor (e.g., quad-core or higher) with sufficient processing power to handle concurrent user requests efficiently.

RAM: At least 8GB of RAM to ensure smooth operation and responsiveness, with the ability to scale based on the system's demands.

Storage: SSD storage for faster data retrieval and improved system performance compared to traditional HDD.

3.2.2. Networking:

Network Interface Cards (NICs): Gigabit Ethernet NICs to support high-speed data transfer between the server and clients.

Load Balancer: If utilizing multiple servers, a load balancer to evenly distribute incoming traffic and improve system reliability.

3.2.3. Security:

Firewall: Hardware firewall to control incoming and outgoing traffic, enhancing system security.

Intrusion Detection System (IDS): A dedicated IDS to monitor and detect potential security threats.

SSL Acceleration Card: If necessary, an SSL acceleration card can offload SSL/TLS encryption/decryption tasks, improving overall performance.

3.2.4. Backup and Redundancy:

Backup Storage: Sufficient backup storage space to regularly back up critical data and configurations.

Redundant Power Supply: Redundant power supplies to ensure continuous operation in case of power failures.

3.2.5. Monitoring:

Server Monitoring Tools: Utilize monitoring tools (e.g., Nagios, Prometheus) to track hardware health, performance metrics, and potential issues.

3.3: User Interface: -

3.3.1. End-User Devices:

Desktops/Laptops: Ensure compatibility with a range of desktop and laptop devices, with considerations for various screen sizes and resolutions.

Smartphones/Tablets: Optimize the user interface for mobile responsiveness, supporting different devices and screen orientations.

3.3.2. Operating Systems:

Windows, macOS, Linux: Ensure compatibility with major desktop operating systems.

iOS, Android: Optimize for mobile operating systems to provide a seamless experience on smartphones and tablets. #@keeep it softwear requrddsflk

3.3.3. Browsers:

Google Chrome, Mozilla Firefox, Safari, Microsoft Edge: Support popular web browsers to ensure a consistent experience across different platforms.

Mobile Browsers: Optimize for mobile browsers, including Chrome and Safari, for a smooth experience on smartphones.

3.3.4. Input Devices:

Mouse and Keyboard (Desktop): Design the user interface to support traditional desktop input devices.

Touchscreen (Mobile): Implement touch-friendly controls and gestures for mobile devices.

3.3.5. Screen Resolution:

Responsive Design: Design the interface to be responsive, adapting to different screen sizes and resolutions.

High-DPI Displays: Optimize graphics and images for high-DPI (dots per inch) displays for better clarity.

3.3.6. Graphics and Multimedia:

Graphics Card Compatibility: Ensure compatibility with a range of graphics cards for displaying multimedia content.

Audio Compatibility: Support audio output for features such as order confirmation sounds or voice navigation.

ONLINE FOOD DELIVERY

PRACTICAL-4

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

AIM: Develop Software project management planning (SPMP) for the specified module.

Software project management planning:

The software project management planning phase involves defining the project scope, objectives, and assembling a cross-functional team with clear roles. Stakeholders are identified, and potential risks are assessed with mitigation plans. A realistic project timeline, resource allocation, and budget are established, along with quality assurance standards. A communication plan, change management processes, and comprehensive documentation are implemented. Project management tools aid in efficient task management and collaboration. Legal and compliance considerations are factored in, and closure criteria are established. Regular reviews ensure the plan aligns with project goals, with a kick-off meeting setting expectations for the team.

7 Principles of SPMP:

1. Compartmentalization:

Facilitates focused management and development of each part, making it easier to track progress and address issues within specific areas.

2. Interdependency:

Understanding and managing interdependencies is crucial to ensure that tasks are completed in the correct sequence and that changes to one part do not adversely affect others.

3. Time Allocation:

Helps in creating realistic schedules, managing resources efficiently, and meeting project deadlines by ensuring that each task is given an appropriate amount of time.

4. Effort Validation:

Ensures that the project team is working on tasks that contribute to the project's success and that resources are used effectively.

5. Defined Responsibilities:

Promotes accountability, prevents confusion, and ensures that everyone understands their specific contributions to the project.

6. Defined Outcomes:

Provides a clear vision of what needs to be achieved, helping the team stay focused on the end goals and facilitating effective evaluation of project progress.

7. Defined Milestones:

Milestones serve as progress indicators, helping to measure and celebrate achievements, while also allowing for course correction if needed.PS Techniques: -write full forms

Critical Path:

Identifying the critical path involves determining the longest sequence of dependent tasks, which will determine the project's overall duration.

1. Task Sequence:

System Design → Database Development → Front-end Coding → Back-end Coding → Testing → Deployment

2. Critical Path Duration:

In developing our online food delivery system, critical tasks span various phases. Initiating with the User Function Model (16 days), followed by the 17-day Backup Server Model, subsequent phases shape the user interface and functionality in 220 days. Key components include Menu Model (7 days), HTTP API Model (6 days), List View Model (7 days), Integrating and Finalizing (7 days), and 18-day Debugging I for system robustness. UI Design Session II (10 days), Database Design Session II (12 days), Drivers Model (12 days), Implement Server (14 days), and Implement API (14 days) contribute to backend functionality. The Order Management Model streamlines processing (20 days). Later phases involve Database Testing (5 days), Backup Server Testing (4 days), Integrating and Finalizing (7 days), Testing and Debugging II (18 days), and a 3-day Publishing phase for deployment. This interconnected critical path defines the project's overall success.

Program Evaluation and Review Technique (PERT):

PERT involves estimating three-time durations for each task: optimistic, most likely, and pessimistic. The expected time (TE) is then calculated as a weighted average.

1. Task Durations:

System Design: Optimistic (8 days), Most Likely (10 days), Pessimistic (12 days)

Database Development: Optimistic (12 days), Most Likely (15 days), Pessimistic (18 days)

Front-end Coding: Optimistic (18 days), Most Likely (20 days), Pessimistic (22 days)

Back-end Coding: Optimistic (22 days), Most Likely (25 days), Pessimistic (28 days)

Testing: Optimistic (12 days), Most Likely (15 days), Pessimistic (18 days)

Deployment: Optimistic (8 days), Most Likely (10 days), Pessimistic (12 days)

2. PERT Calculation for Each Task:

PERT TE = (Optimistic + 4 Most Likely + Pessimistic) / 6

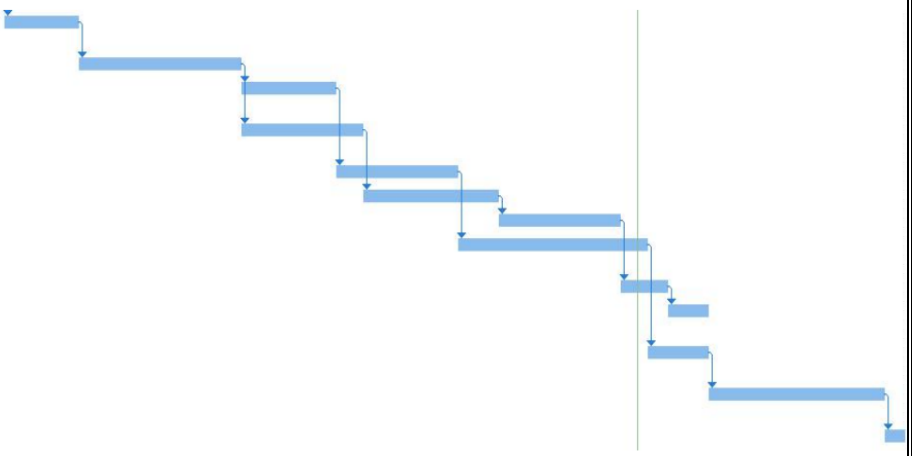
3. Expected Project Duration:

Summing up the expected durations of all tasks gives the overall expected project duration.

Gantt Chart:

A Gantt chart visually represents the project schedule over time.

11		Intergrating and Finalizing	7 day
12		Debugging I	18 da
13		UI Design Session II	10 da
14		Databese Design Session II	12 da
15		Drivers Model	12 da
16		Implement Server	14 da
17		Implement API	14 da
18		Order Management Model	20 da
19		Database Testing	5 day
20		Backup Server Testing	4 day
21		Intergrating and Finalizing	7 day
22		Testing and Debuging II	18 da
23		Publishing	3 day



ONLINE FOOD DELIVERY

PRACTICAL – 06

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

AIM : Prepare System Analysis and System Design of identified Requirement specification using structure design as DFD with data dictionary and Structure chart for the specific module.

DATA FLOW DIAGRAM :

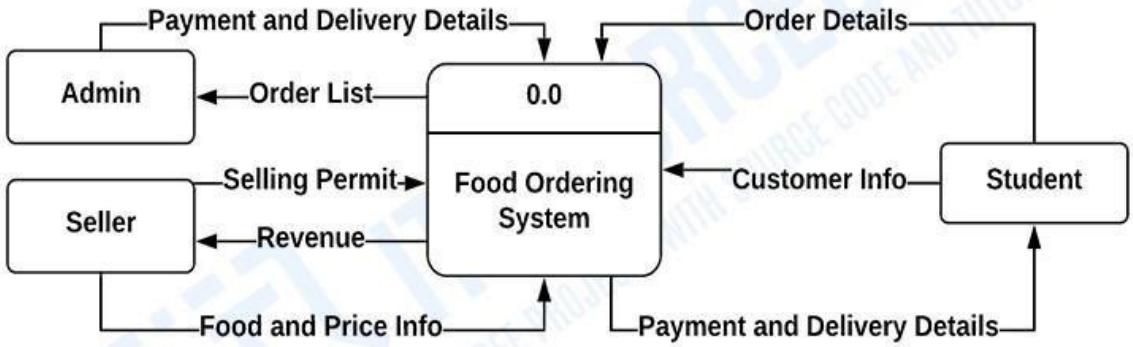
Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.Food Ordering System is actually a type of software that allows the manager of restaurants to manage and accept the placed orders over the Internet or in the restaurant. Let us understand the working of the food ordering system by DFD are mainly 3 types.

1.level 0
2.level 1

3.level 2

Level 0 :

Level 0 is also called as fundamental system model. At this level, the Input and Output of the system are shown. The system is designed and established across the world with input and output at this level.



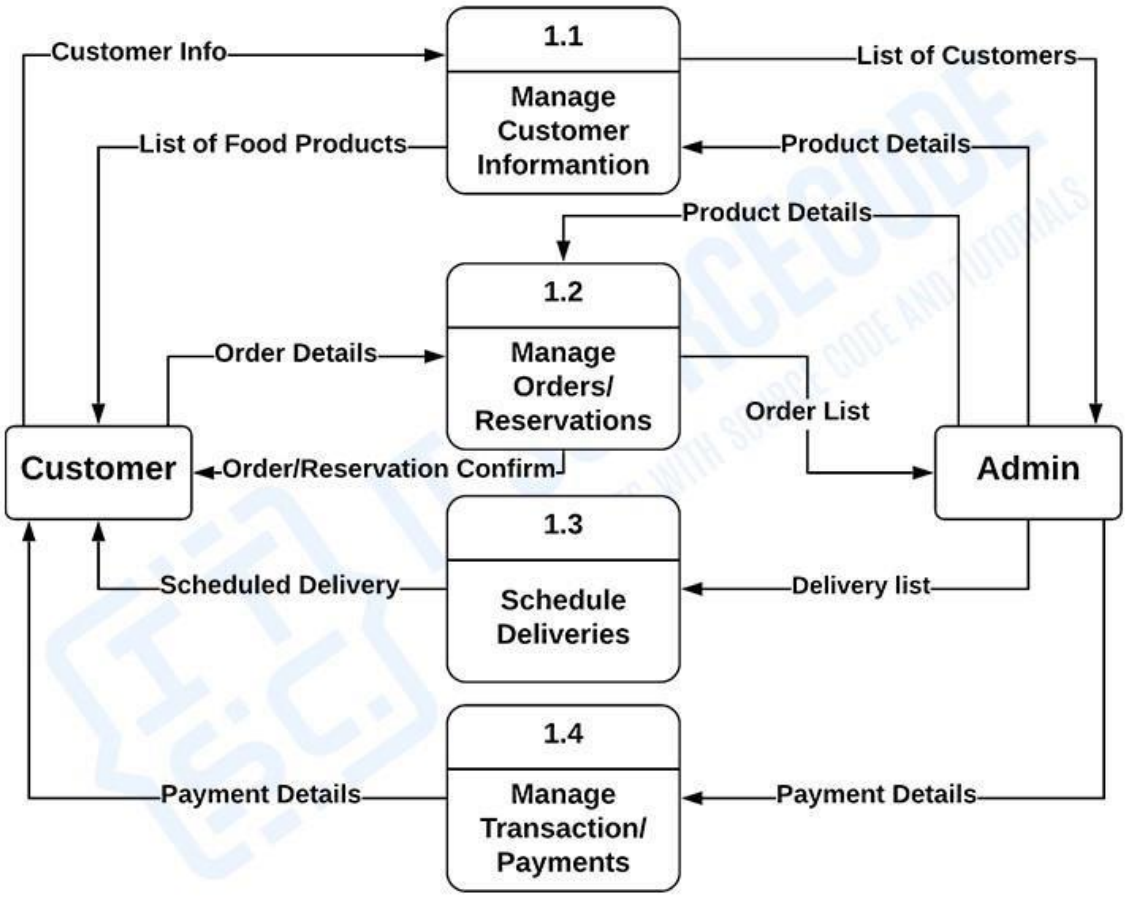
Level 1 : For processing the order, process 1.0 is responsible. For food, the housekeeping activities involved are represented by processes 2.0, 3.0, and 4.0. The detailed information about daily sold items should be available to create and report management and the list of items that are available 'in-stock' should be kept by maintaining the inventory data.

The "detonated view" of the context diagram is **Food Ordering System DFD Level 1**.

Its function is to deepen the concept derive from the context diagram.

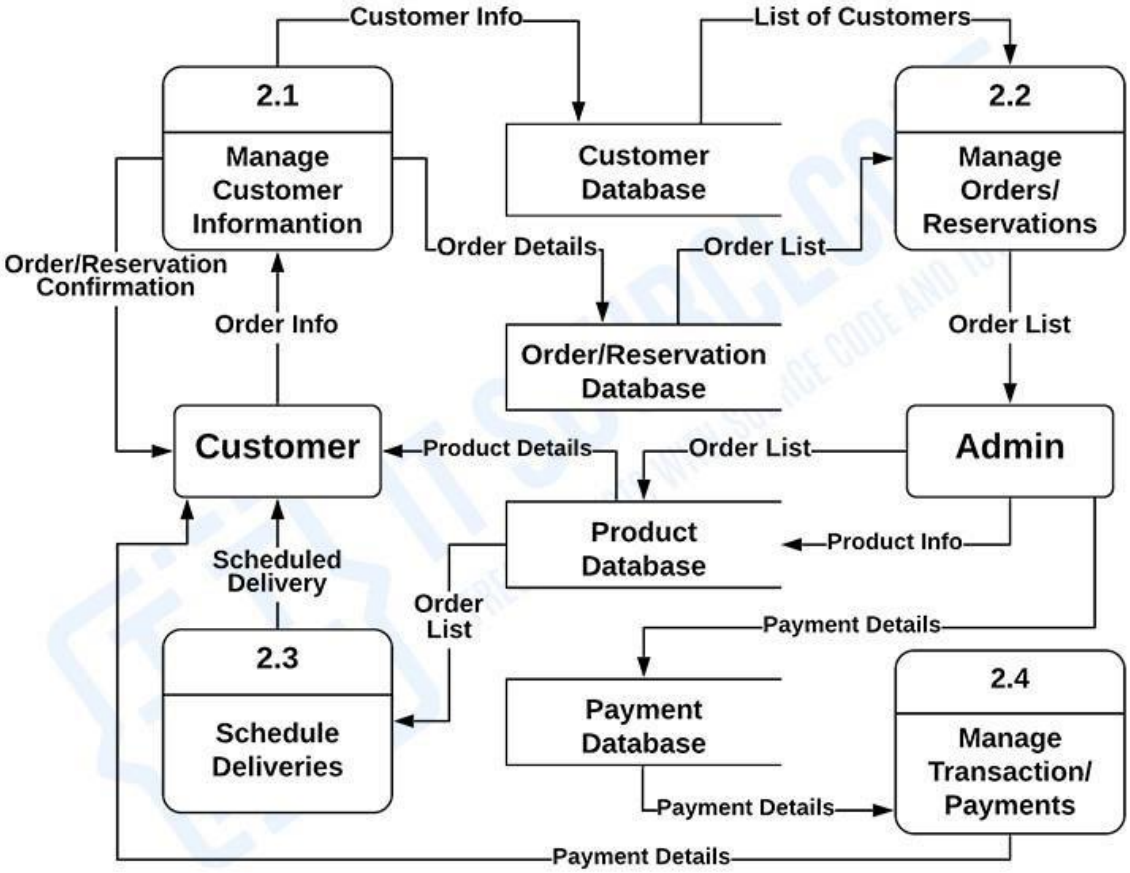
Specifically, level 1 shows the broader details of Food Ordering System DFD Level 0.

This is to clarify the paths (flow) of data and its transformation from input to output.



Level 2 DFD for Food Ordering System is also the highest abstraction of the data flow diagram. This level also broadens the idea from the DFD level 1. It includes the subprocesses from level 1 as well as the data that flows.

However, not all of the processes in the project must have sub-processes. Only provide this diagram if needed. As long as your previous diagrams were clear and precise, this level is not required.


Customer Info
List of Customers
2.1
Manage Customer Informantion
Customer Database
2.2
Manage Orders/ Reservations
Order Details
Order List
Order/Reservation Confirmation
Order Info
Order/Reservation Database
Order List
Customer
Product Details
Order List
Admin
Product Database
Product Info
Scheduled Delivery
Order List
Payment Details
2.3
Schedule Deliveries
Payment Database
2.4
Manage Transaction/ Payments
Payment Details
Payment Details

ONLINE FOOD DELIVERY

PRACTICAL – 07

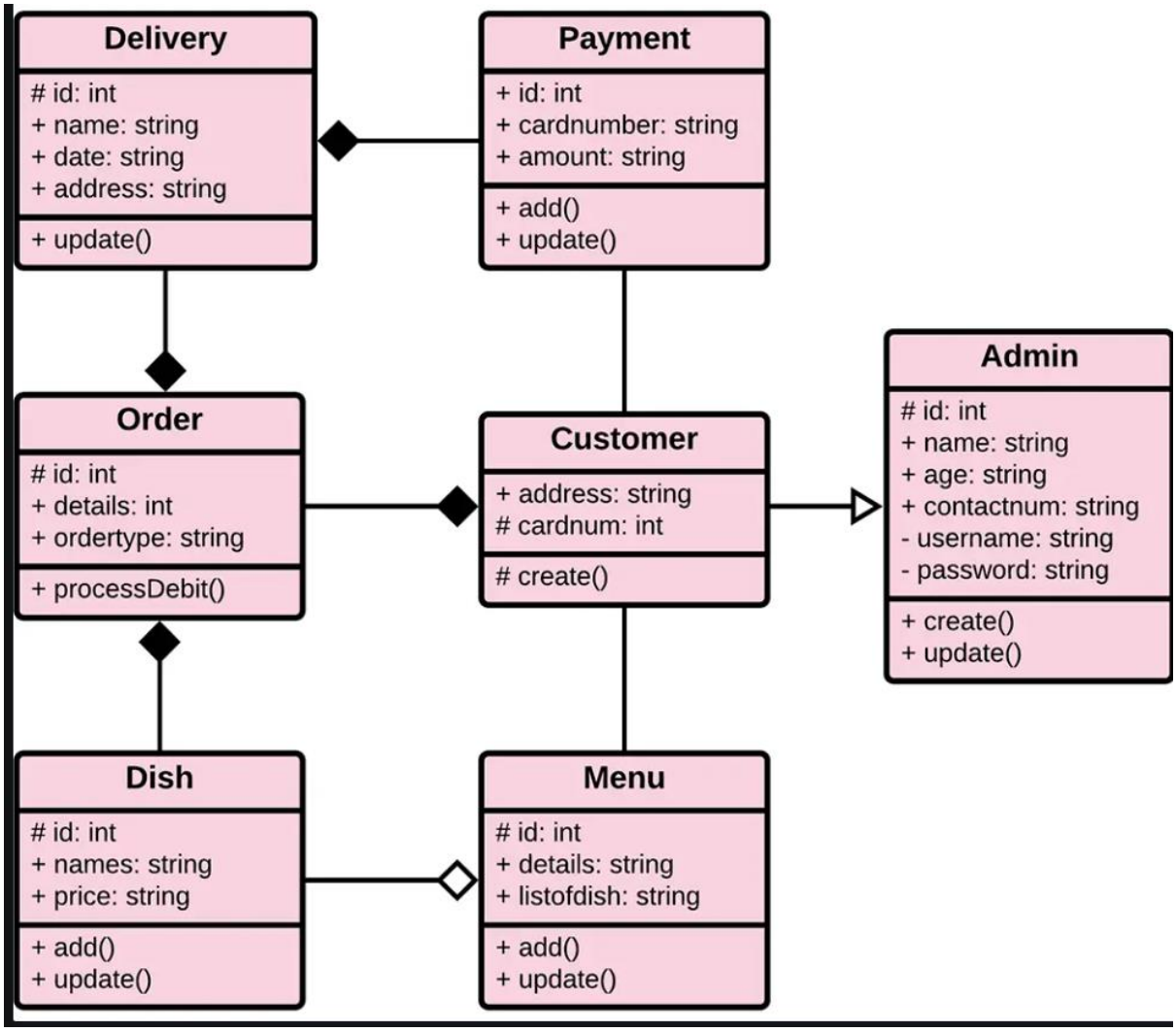
Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

AIM: Designing the module using Object Oriented approach including Use case Diagram with scenarios, Class Diagram and State Diagram, Collaboration Diagram, Sequence Diagram and Activity Diagram.

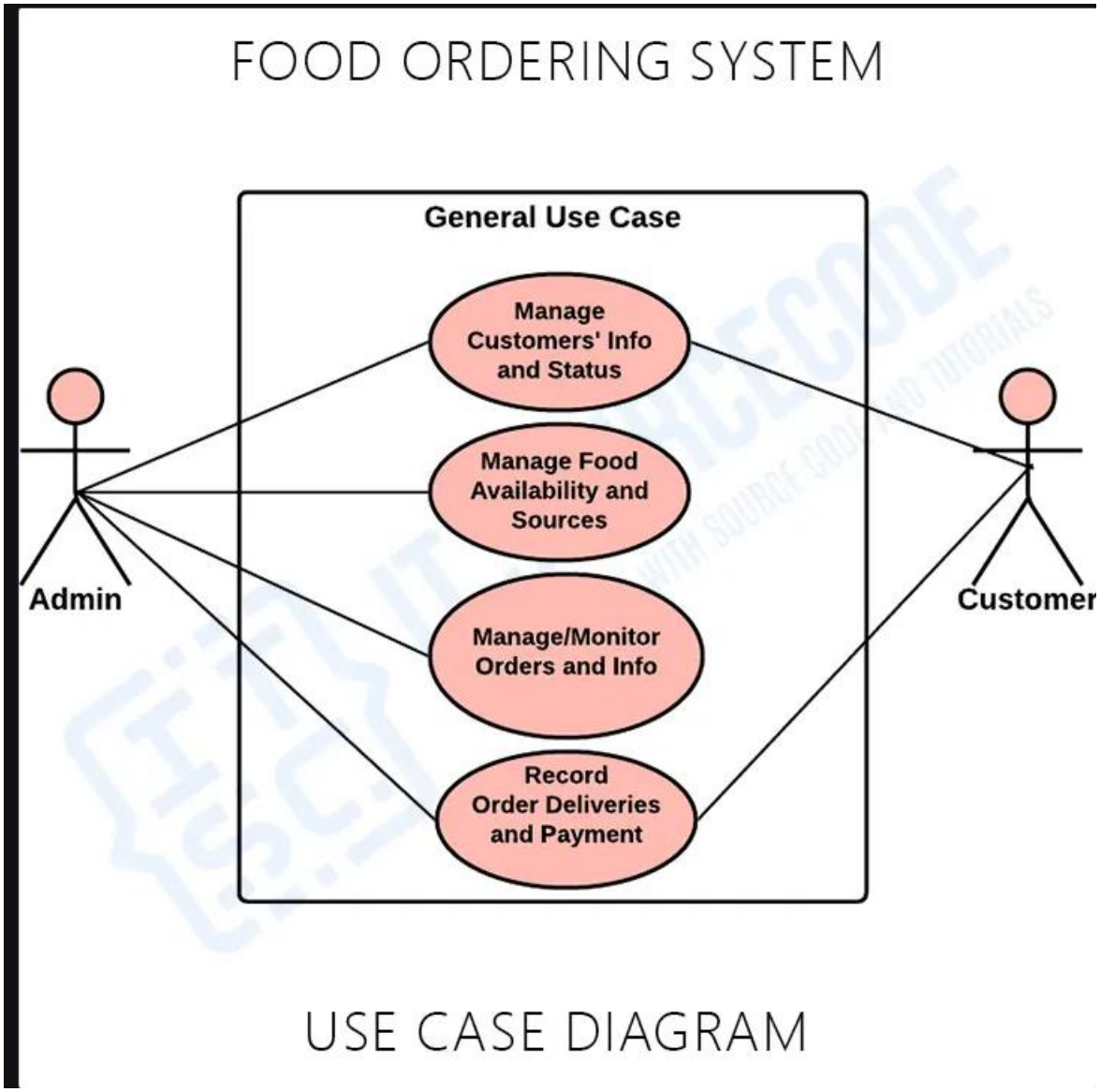
Online Food Ordering System Class Diagram:

Each of the classes will have their attributes in accord to the methods they will use. So the UML Class diagram was illustrated by a box with 3 partitions and the upper part was the name of the class, the middles are the attributes and the bottom is for the methods. The arrows on them represents their relationships in each other



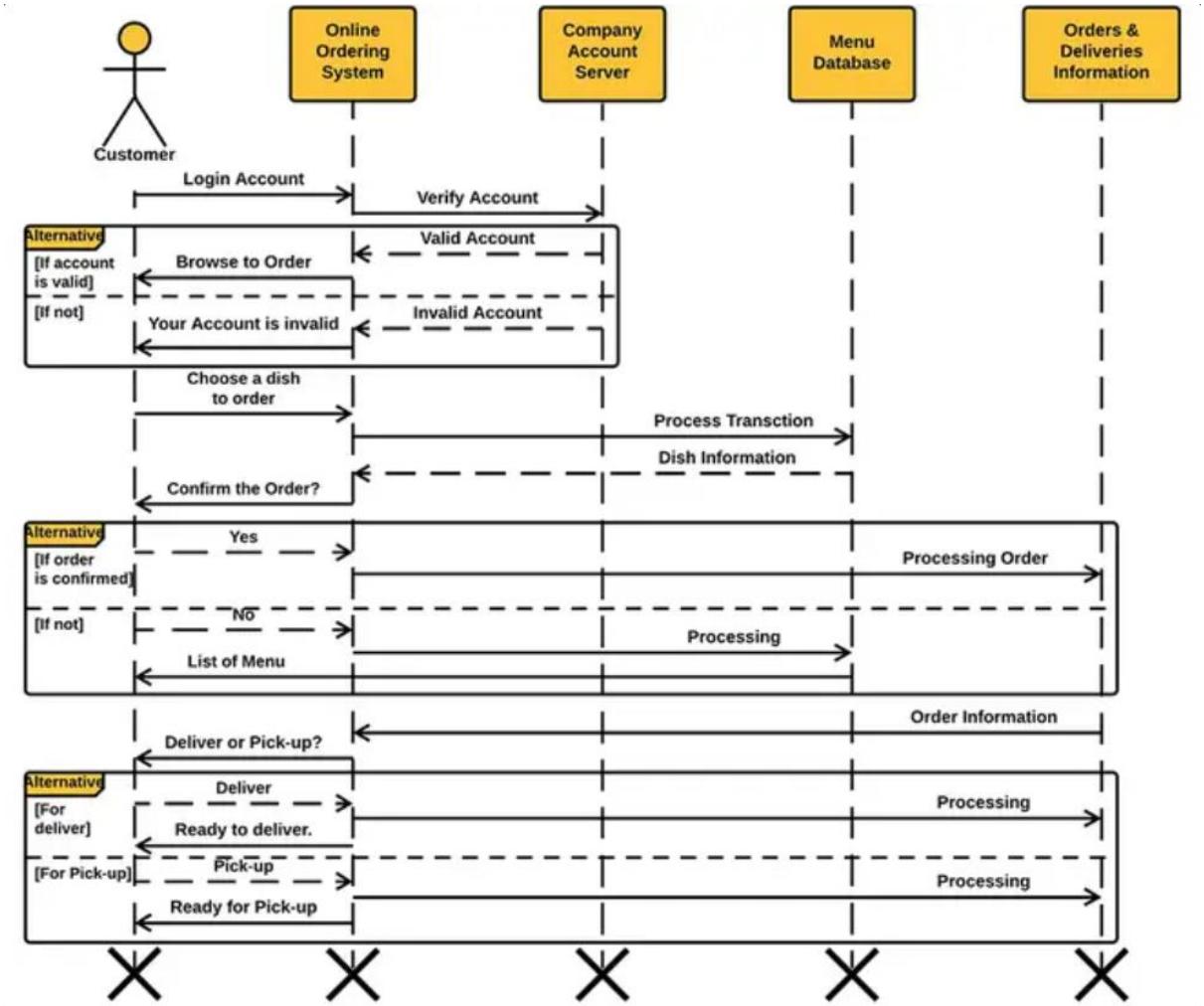
Food Ordering System Use Case Diagram

The use cases in the diagram represents the main processes in an Online Food Ordering system. Then they will be broken down into more specific use cases depending on the included processes of the main use case. Each of these use cases explains how the system handles the actions or scenarios requested by the user.



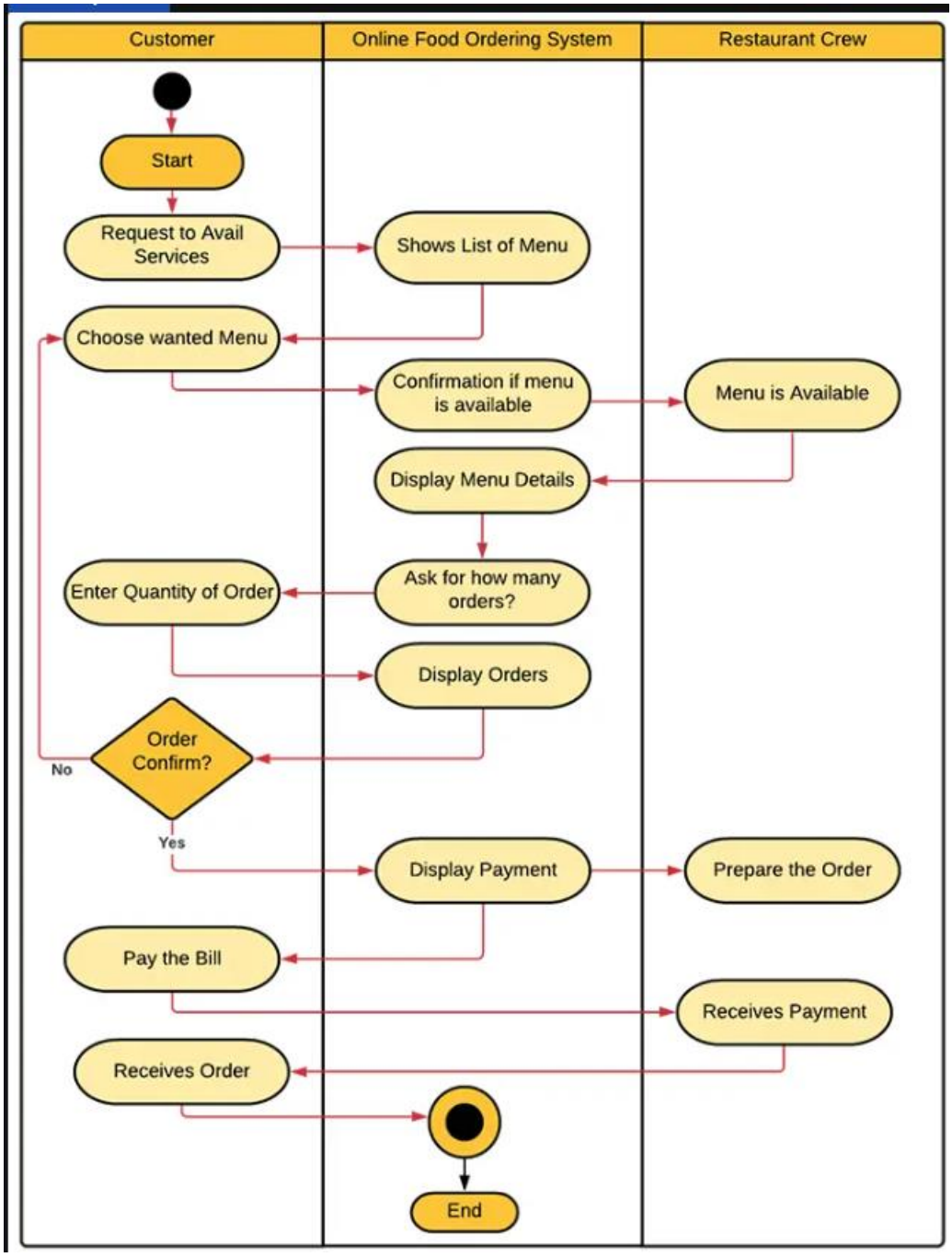
Online Ordering System Sequence Diagram

The designed sequence diagram illustrates the series of events that occurs in Online Food Ordering System. In this illustration, the actors are represented by a stick man and the transactions or classes are represented by objects. It will give you clear explanation about the behavior of an Online Food Ordering System in terms of processing the flow of instructions.



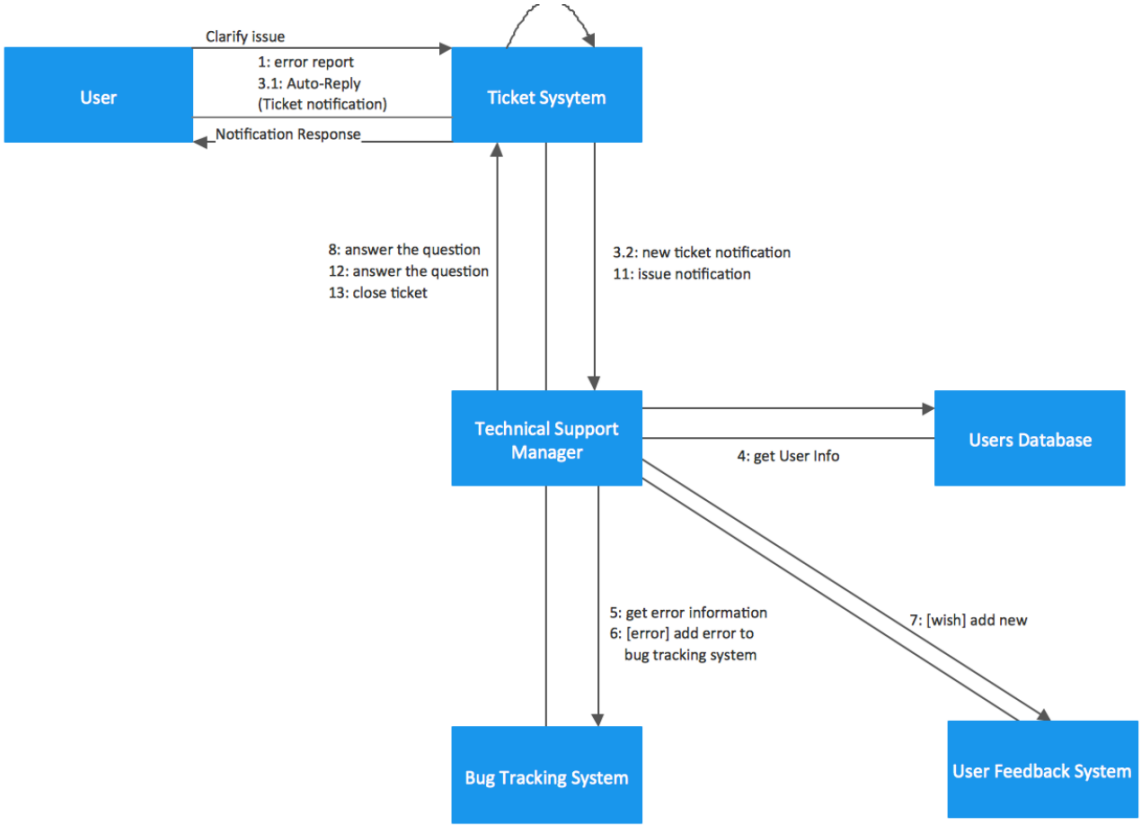
Online Food Ordering System Activity Diagram | UML

Here's the UML activity diagram design of Online Food Ordering System that you can use for your own Final year Project. The UML activity Diagram is used to show the interaction of the user and the system. By creating it, you'll be able to see the flaws of the system and you may avoid it once you apply it to the project development. So it is important to have your diagrams designed first before jumping into its development



Online Food Ordering Collaboration Diagram:

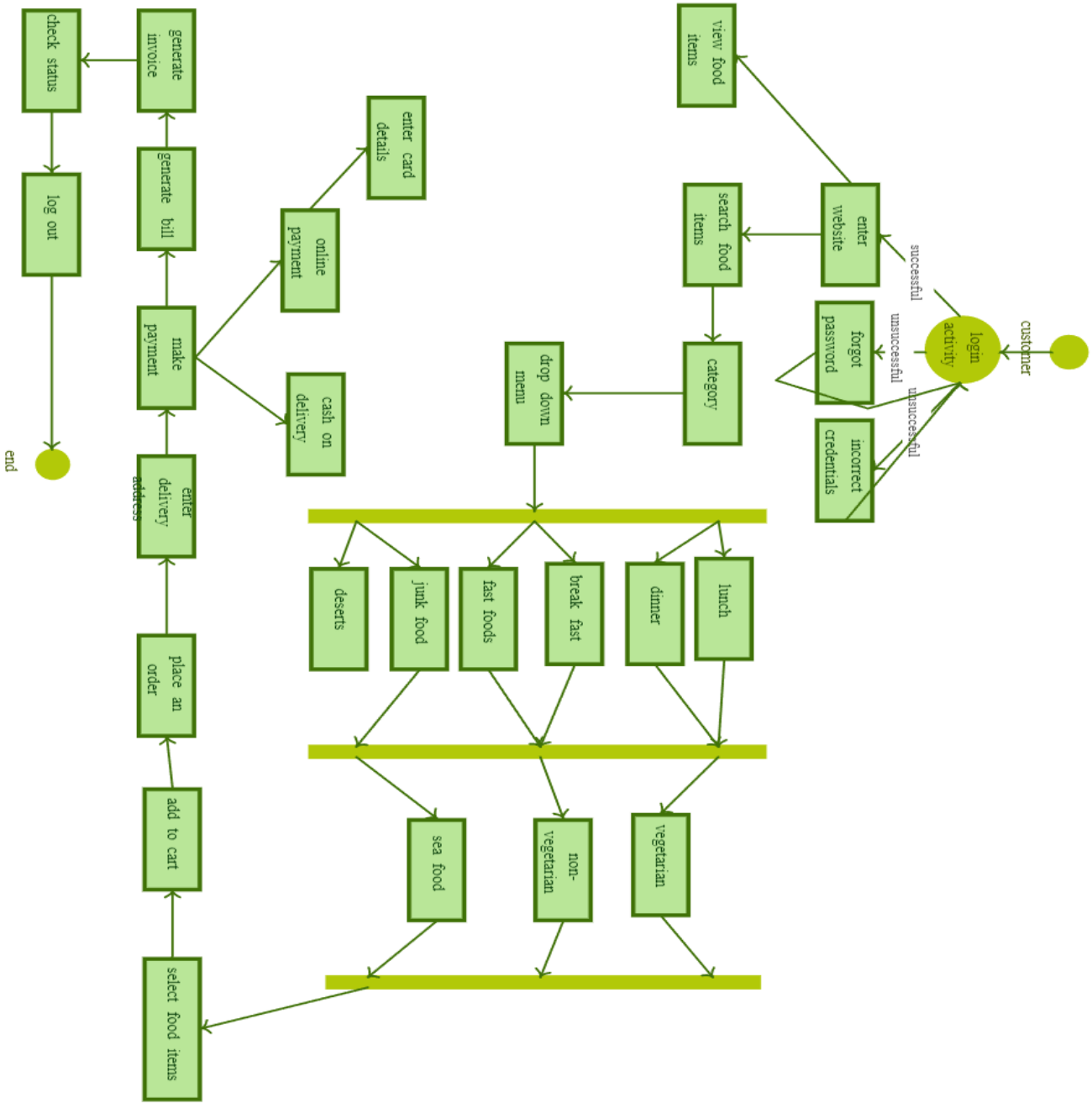
A collaboration diagram, unlike sequence diagrams which focus on message order, shows how objects in a system work together to achieve a function. It depicts objects, their connections, and the messages exchanged between them, helping visualize how objects collaborate to complete a specific use case.



Online Food Ordering State Diagram:

A state diagram, also known as a state machine diagram or state chart diagram, is a visual representation of the behavior of a system in software engineering. It shows the different states that a system can be in and the events or conditions that cause it to transition between those states.

PARUL UNIVERSITY
FACULTY OF ENGINEERING
SOFTWARE ENGINEERING (303105254)
B-Tech 2nd YEAR



ONLINE FOOD DELIVERY

PRACTICAL – 08

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

Aim: Defining Coding Standards and walk through.

CODING STANDARDS:

Different modules specified in the design document are coded in the Coding phase according to the module specification. The main goal of the coding phase is to code from the design document prepared after the design phase through a high-level language and then to unit test this code.

Good software development organizations want their programmers to maintain to some well-defined and standard style of coding called coding standards. They usually make their own coding standards and guidelines depending on what suits their organization best and based on the types of software they develop. It is very important for the programmers to maintain the coding standards otherwise the code will be rejected during code review.

Purpose of Having Coding Standards.

Uniform Appearance and Readability:

A coding standard gives a uniform appearance to code written by different engineers.

It improves readability and makes it easier to understand and maintain the code.

Reducing Complexity and Detecting Errors:

Coding standards help reduce complexity by enforcing clear and concise practices.

They facilitate error detection during code review.

Code Reusability and Efficiency:

Consistent coding practices promote code reuse.

Developers can more easily understand and integrate existing code..

Common Coding Standards:

Limited Use of Globals:

Define which data can be declared global and which cannot.

Standard Headers for Modules:

Headers should include module name, creation date, author, modification history, synopsis, and function details.

Naming Conventions:

Meaningful variable names

(e.g., localData, GlobalData, CONSDATA).

Function names in camel case (e.g., calculateTotalRevenue).

Indentation and Spacing:

Proper indentation enhances readability.

Use spaces consistently (e.g., after commas, within nested blocks).

Why do we need coding standards?

They reduce security concerns and performance issues that might have resulted from poor coding practices;

They help guarantee code quality, making your code easier to read, analyze, and work through. The code also becomes easier to maintain and extend, even by new developers;

They lead to lower code complexity and more elegant design solutions;

Any developer can examine any part of the code, understand it, and change it independently of when and who wrote it.

Draw backs for not following proper coding conventions

Avoid using a coding style that is too difficult to understand: Code should be easily understandable. The complex code makes maintenance and debugging difficult and expensive. Avoid using an identifier for multiple purposes: Each variable should be given a descriptive and meaningful name indicating the reason behind using it

Reduced engineers motivation

Increase Development time

Complex code base structure

Google code of Conduct:

The Google Code of Conduct is one of the ways we put Google's values into practice. It's built around the recognition that everything we do in connection with our work at Google will be, and should be, measured against the highest possible standards of ethical business conduct. We set the bar that high for practical as well as aspirational reasons: Our commitment to the highest standards helps us hire great people, build great products, and attract loyal users. Respect for our users, for the opportunity, and for each other are foundational to our success, and are something we need to support every day.

So please do read the Code and Google's values, and follow both in spirit and letter, always bearing in mind that each of us has a personal responsibility to incorporate, and to encourage other Googlers to incorporate, the principles of the Code and values into our work. And if you have a question or ever think that one of your fellow Googlers or the company as a whole may be falling short of our commitment, don't be silent. We want – and need – to hear from you.

Coding review :

Code review (sometimes referred to as peer review) is a software quality assurance activity in which one or more people check a program, mainly by viewing and reading parts of its source code, either after implementation or as an interruption of implementation. At least one of the persons must not have authored the code. The persons performing the checking, excluding the author, are called "reviewers".

Although direct discovery of quality problems is often the main goal]code reviews are usually performed to reach a combination of goals:

Better code quality – Improve internal code quality and maintainability (such as readability, uniformity, and understandability)

Finding defects – Improve quality regarding external aspects, especially correctness, but also find issues such as performance problems, security vulnerabilities, and injected malware

Learning/Knowledge transfer – Help transfer codebase knowledge, solution approaches, and quality expectations, both to the reviewers and the author

Increase sense of mutual responsibility – Increase a sense of collective code ownership and solidarity

Finding better solutions – Generate ideas for new and better solutions and ideas that transcend the specific code at hand

Review types:

Management reviews

Technical reviews

Inspections

Walk-throughs

Audits

Code Inspection:

Historically, the first code review process that was studied and described in detail was called "Inspection" by its inventor, Michael Fagan. This Fagan inspection is a formal process which involves a careful and detailed execution with multiple participants and multiple phases. Formal code reviews are the traditional method of review, in which software developers attend a series of meetings and review code line by line, usually using printed copies of the material. Formal inspections are extremely thorough and have been proven effective at finding defects in the code under review.

Code Walk-throughs:

In recent years many industry teams have introduced a more lightweight type of code review in which the scope of each review is based on the changes to the codebase performed in a ticket, user story, commit, or some other unit of work. Furthermore, there are rules or conventions that embed the review task into the development process (e.g., "every ticket must be reviewed"), commonly as part of a pull request, instead of explicitly planning each review. Such a review process is called "regular, change-based code review". There are many variations of this basic process. A survey among 240 development teams from 2017 found that 90% of the teams use a review process that is based on changes (if they use reviews at all), and 60% use regular, change-based code review. Also, most large software corporations such as Microsoft Google, and Facebook follow a change-based code review process.

ONLINE FOOD DELIVERY

PRACTICAL – 09

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

AIM : Write the test cases for the identified module.

What is tesing :

A test case is a defined format for software testing required to check if a particular application/software is working or not. A test case consists of a certain set of conditions that need to be checked to test an application or software i.e. in more simple terms when conditions are checked it checks if the resultant output meets with the expected output or not. A test case consists of various parameters such as ID, condition, steps, input, expected result, result, status, and remarks.

Resources are needed for testing :

1.Test Environments: Set up test environments that mimic the production environment to conduct various types of testing, including functional, integration, and performance testing.

2. **Test cases :** Develop comprehensive test cases covering various scenarios such as user registration, order placement, payment processing, menu browsing, and delivery tracking.

3.Test data : Prepare test data including sample menus, user profiles, payment details, and delivery addresses to simulate real-world scenarios.

4. Testing tools : Utilize testing tools for automation, load testing, and security testing. Automation tools like Selenium or Cypress can automate repetitive tests, while tools like JMeter or Gatling can help with load testing.

5.Moblile device and emulators : Test the mobile application on various devices and emulators to ensure compatibility across different screen sizes, resolutions, and operating systems.

6.Network infrastructure : Test the application's performance under different network conditions (3G, 4G, Wi-Fi) to ensure smooth operation for users with varying internet speeds

7.Payment gateways : Integrate with payment gateways in a test environment to ensure secure and seamless payment processing. Use sandbox environments provided by payment gateways for testing payment transactions without using real money.

8.Security measures : Conduct security testing to identify and fix vulnerabilities such as SQL injection, cross-site scripting (XSS), and authentication flaws to protect user data and transactions.

9.Localization and internationalization resources : If the platform operates in multiple regions, ensure testing covers different languages, currencies, time zones, and cultural preferences.

10.Feedback mechanism : Implement mechanisms for collecting feedback from testers, users, and stakeholders to continuously improve the platform's usability and functionality.

11.Documentation : Document test cases, test results, and any issues encountered during testing for reference and future improvements.

12.Performance monitoring tools : Implement tools for monitoring application performance in production to detect and address performance issues proactively.

Test cases	Explanation	Result
User registration	Verify that users can successfully register on the platform.	pass
Login functionality	Ensure users can log in with valid credentials.	pass
Menu browsing	Check that users can browse the menu and add items to the cart.	pass
Order placement	Verify that users can place orders successfully.	pass
Order tracking	Ensure users can track their orders in real-time.	pass
Payment processing	Verify that payment processing is secure and reliable.	pass

No	Test case id	Test case objective	Prerequisite	Steps	Input data	Expected result	Actual result	Remarks
1.	TC_UI_001	Verify user location on Homepage	User has location services enabled	Launch the online food delivery app	N/A	User's current location is displayed on the homepage.	Book is issued on the students account.	Test GPS accuracy and location permission handling.
2.	TC_UI_002	Search for restaurant by name	Search functionality Available	1. . Launch the app. 2. Access the search bar. 3. Enter a specific restaurant name.	Restaurant name: "Pizza Palace"	1.List of restaurants matching the specific name is displayed. 2.List of restaurants offering the chosen cuisine is displayed. 3. List of restaurants with relevant keywords in their name or menu is displayed.	Search results show irrelevant restaurants or lack filtering options.	1. Test search functionality for accuracy, comprehensiveness, and ability to handle different search terms. 2.Validate filtering options based on cuisine and other categories.
3.	TC_ORD_001	Place order with registered account	User has a registered account with valid payment method	1.login to the app with a registered account . 2.select a restaurant and menu items . 3.proceed to checkout. 4.enter valid payment . 5.confirm order.	1.User: John Doe (registered account) 2.Payment method: Credit Card (1234****1234)	Order confirmation with estimated delivery time is displayed.	Order fails or confirmation details are incorrect.	Test entire order placement process for registered users. Validate payment processing.

4.	TC_PAY_001	Payment processing with different methods	Multiple payment methods available	1. Login to the app. 2. Add items to cart. 3. Proceed to checkout. 4. Try different payment methods (credit card, debit card, etc.).	Payment Method 1: Credit Card (Valid details) - Payment Method 2: Debit Card (Valid details)	Payment is successful for both methods with confirmation messages.	Payment fails for any method.	Test functionality of various payment processing options
5.	TC_DEL_002	Verify order delivery	Order is placed successfully	Place an order (refer to TC_ORD_001).	N/A	Order is delivered to the specified address within the estimated timeframe.	Order is incomplete, delivered late, or delivered to the wrong addres.	Test delivery logistics and communication with restaurants/delivery personnel.
6.	TC_DEL_003	Report missing item in order	Order is delivered	. Receive an order with missing items.	Missing item: "Fries"	The platform offers a way to report missing items and receive support.	Reporting functionality is unavailable or manufacturing.	Test the system's ability to handle order discrepancies and customer support.

ONLINE FOOD DELIVERY

PRACTICAL – 10

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

Aim: Demonstrate the use of different testing tools with comparison

Objective: Testing an online food ordering system aims to guarantee a seamless user experience. This involves checking various functions like menu browsing, order placement, payment processing, all while ensuring it's user-friendly and secure. By verifying these aspects, you can launch a reliable system that handles customer data safely and efficiently.

Tools for testing:

Selenium:

- **Pros:** Free, open-source, highly flexible for web testing with various programming languages.
- Cons: Requires coding knowledge, lacks built-in reporting features.

Katalon Studio:

- Pros: Free (with paid enterprise version), user-friendly interface for web and mobile testing, supports both scripting and keyword-driven testing.
- Cons: Limited compared to paid tools in some functionalities, may require scripting knowledge for complex tests.

UFT (Micro Focus):

- Pros: User-friendly interface, comprehensive features for web, mobile, and API testing, good integration with other tools.
- Cons: Paid tool, scripting limited to VBScript, potential vendor lock-in.

TestComplete (SmartBear):

- Pros: User-friendly interface, supports various application types (web, mobile, desktop), offers keyword-driven testing and recording.
- Cons: Paid tool, may require scripting for complex tests.

Tricentis Tosca:

- Pros: Comprehensive solution for all testing needs (web, mobile, desktop, API), supports Agile and DevOps workflows, offers model-based testing.
- Cons: Expensive, complex to learn initially.

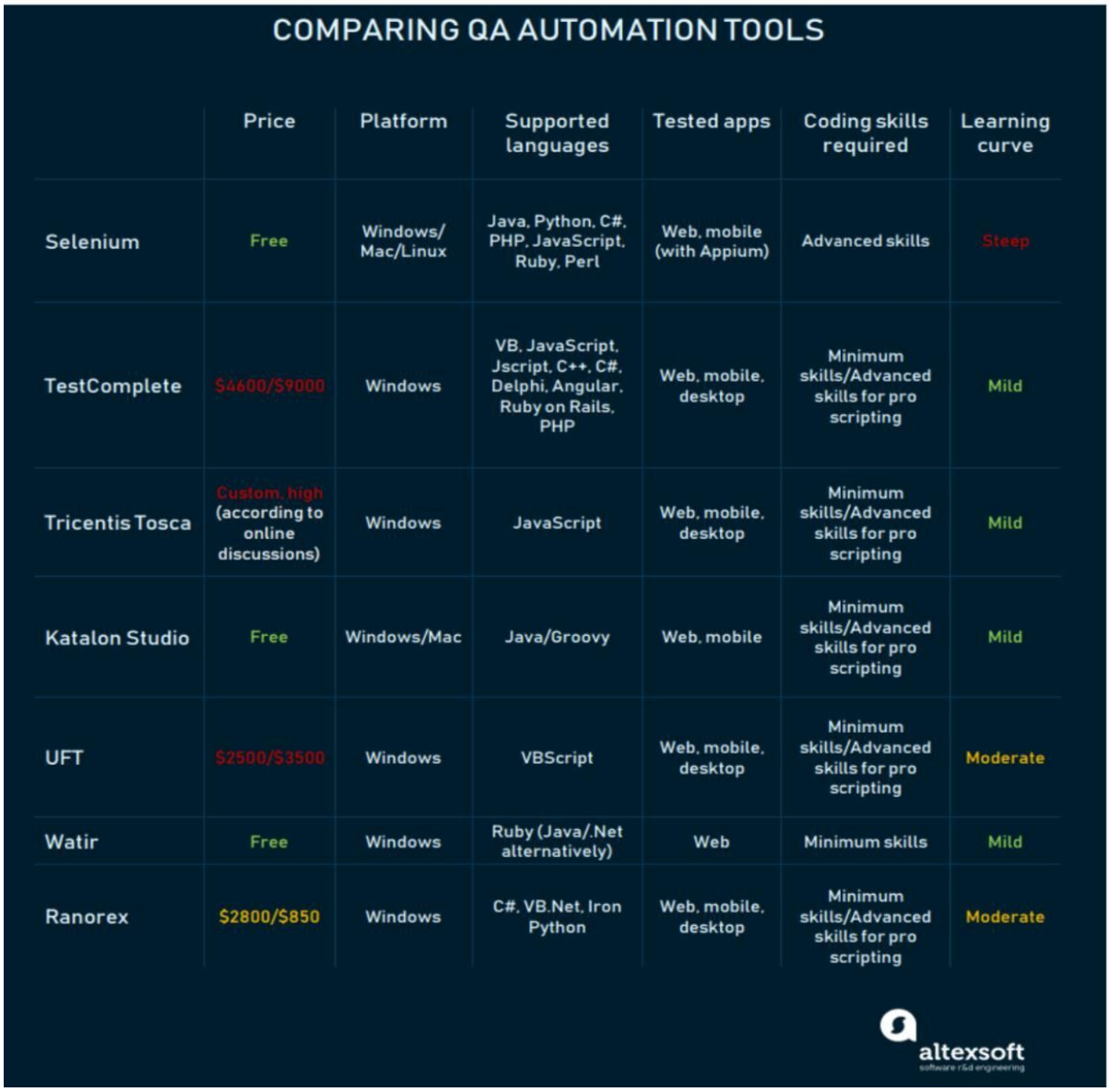
Ranorex:

- Pros: User-friendly interface, supports web, mobile, desktop, and API testing, offers visual recording and codeless automation.
- Cons: Paid tool, may require scripting for complex tests.

Watir:

- Pros: Free, open-source Ruby library for web testing, easier to learn than raw Selenium.
- Cons: Limited compared to other frameworks, less actively maintained.

Comparison Table:



Steps to install Selenium IDE:

Installation:

- Download the setup from the official website: https://www.selenium.dev/

Check Chrome Version:

- Open Chrome Browser -> Help -> About Google Chrome

Download ChromeDriver:

- Based on your Chrome version, download the compatible ChromeDriver version from https://chromedriver.chromium.org/downloads. In this example, we'll download version 75 of chromedriver.exe.

Copy ChromeDriver:

- Download the .exe file for your operating system and copy it to a local directory.

Set Path (C:\webdriver\chromedriver.exe):

- The path to your chromedriver.exe will be used in your program.

Selenium Setup with ChromeDriver

- Now that ChromeDriver is set up, we'll use Eclipse to execute our Selenium code. Here's how to create and run Selenium code in Eclipse:

Create a New Maven Project:

- In Eclipse, go to File -> New -> Others -> Maven Project.

Latest testing tools in AI:

Functionize: This tool leverages both AI and machine learning (ML) for automating complex tests across web, mobile, and API applications.expand_more It eliminates the need for repetitive scripting and allows collaboration between development and testing teams.expand_more

TestCraft: This is another AI-powered tool that focuses on intelligent test automation.expand_more It uses AI to identify UI elements, generate test data, and heal broken tests, improving efficiency and reducing maintenance needs.expand_more Applitools: This tool utilizes AI for visual testing, ensuring applications render correctly across different browsers and devices.expand_more It goes beyond traditional screenshot comparisons by identifying and highlighting visual inconsistencies.expand_more

Mabl: This platform offers AI-powered codeless test automation.expand_more It allows testers to create tests through a user-friendly interface without needing to write code.expand_more Mabl utilizes AI to learn user behavior and adapt tests over time.expand_more

Testim.io: This is an AI-powered test automation tool with a focus on web and mobile applications.expand_more It uses AI to suggest improvements to test cases, identify potential issues, and streamline the testing process.expand_more

ONLINE FOOD DELIVERY

PRACTICAL – 11

Group Information

1.Mutthuluri Varun Kumar-2203031240878
2.Teeparthi kasiviswanadh-2203031241289
3.Narra Adarsh Reddy -2203031240923
4.M Amrutha Lahari -2203031240740

Aim : Define security and quality aspects of the identified module.

Security aspects :

1. Payment Security: Ensuring that customers' payment information is encrypted and securely transmitted to prevent unauthorized access or theft.

2. Data Privacy: Protecting customers' personal information such as addresses, phone numbers, and order history from being accessed or misused by unauthorized parties.

3. Secure Transactions: Implementing secure protocols for online transactions to prevent interception or manipulation of orders or payment details.

4. Authentication and Authorization: Employing robust authentication mechanisms to verify the identity of users and ensure that only authorized individuals can access accounts and place orders.

5. Secure Communication: Using secure communication channels to transmit sensitive information between customers, restaurants, and delivery partners to prevent data breaches.

6. Fraud Prevention: Implementing measures to detect and prevent fraudulent activities such as fake orders, stolen credit cards, or identity theft.

7.Cybersecurity: Safeguarding online platforms against cyber threats such as malware, phishing attacks, or denial-of-service (DoS) attacks that could disrupt operations or compromise data integrity.

Quality aspects :

1. Food Quality: Ensuring that the food delivered meets high-quality standards in terms of taste, freshness, and presentation, maintaining customer satisfaction and loyalty.

2. Timeliness: Delivering orders within the promised time frame to meet customer expectations and minimize dissatisfaction due to delays.

3. Order Accuracy: Ensuring that orders are prepared and delivered accurately according to the customer's specifications to avoid errors or discrepancies.

4. Customer Service: Providing responsive and helpful customer support to address inquiries, complaints, or issues promptly and professionally.

5. Hygiene and Safety: Ensuring that food preparation, packaging, and delivery processes adhere to strict hygiene and safety standards to prevent contamination and foodborne illnesses.

6. Customization and Flexibility: Offering options for customizing orders to accommodate dietary preferences, allergies, or special requests, enhancing the overall customer experience.

7. Feedback Mechanisms: Implementing feedback mechanisms such as ratings, reviews, and surveys to gather insights from customers and identify areas for improvement in service quality.

8. Reliability: Consistently delivering a reliable service without disruptions or cancellations, building trust and confidence among customers.

Different security areas that are provided in online food delivery system :

1. **User Authentication and Authorization:**
    - **Secure Login:** Require users to create accounts with strong passwords and implement multi-factor authentication (MFA) for added security. For example, platforms like Uber Eats or DoorDash use password encryption and two-factor authentication to secure user accounts.
    - **Role-Based Access Control (RBAC):** Restrict access to sensitive data and functionalities based on user roles and permissions. For instance, only authorized administrators should have access to backend systems for managing orders and user accounts.

2. **Data Encryption:**
    - **Transport Layer Security (TLS):** Encrypt data transmitted between the user's device and the server to prevent eavesdropping and tampering. Online food delivery services commonly use TLS encryption for secure communication during order placement and payment processing.
    - **End-to-End Encryption**: Implement encryption mechanisms to protect sensitive information such as payment details and personal data throughout the entire transaction process. For example, services like Grubhub employ end-to-end encryption to safeguard user data from interception or unauthorized access.
3. **Data Privacy and Compliance:**
    - **Privacy Policies:** Clearly outline how user data is collected, processed, and stored, and ensure compliance with relevant data protection regulations such as GDPR or CCPA. Online food delivery services typically publish privacy policies detailing their data handling practices.
    - **Data Access Controls:** Limit access to user data to authorized personnel only and establish strict protocols for data handling and storage to maintain privacy and confidentiality.
4. **Platform Security:**
    - **Regular Security Audits:** Conduct periodic security audits and vulnerability assessments to identify and address potential security risks. For instance, platforms may hire third-party security firms to perform penetration testing and identify vulnerabilities in their systems.
    - **Security Updates and Patches:** Promptly apply security patches and updates to address known vulnerabilities and ensure that systems are protected against emerging threats and attacks.
5. **Cybersecurity Measures:**
    - **Intrusion Detection Systems (IDS):** Deploy IDS solutions to monitor network traffic and detect suspicious or malicious activities such as unauthorized access attempts or malware infections.
    - **Firewalls and Antivirus Software:** Utilize firewalls and antivirus software to prevent unauthorized access to network resources and defend against malware and other cyber threats.