# **Assignment 1**

**Project Title: Inventory Management and Delivery Optimization System**

**Client:** BVG India Limited

**Prepared By:** Diksha Rajendra tupe

**Date:** 25 Feb 2025

**Document Version:** 1.0

**1. Executive Summary:**

**Introduction:**  
The company operates multiple manufacturing plants and warehouses across the country and specializes in producing ice-cream and milk products. They aim to develop a software solution that optimizes two core areas: **inventory management** and **expedited product delivery** to customers, enhancing efficiency, minimizing delays, and streamlining their supply chain operations.

**Project Overview**

This project aims to develop a comprehensive software solution for managing the inventory and optimizing the delivery operations of BVG India Limited, a company specializing in ice cream and milk products. The solution will improve the accuracy of inventory management across various manufacturing plants and warehouses and will facilitate quick and efficient delivery to customers.

**Business Goals**

* **Efficient Inventory Management:** Centralize and automate inventory tracking and control for real-time visibility across all manufacturing plants and warehouses.
* **Optimized Delivery:** Enable faster, cost-effective product delivery to end customers, reducing delivery times and maximizing customer satisfaction.

**Business Objectives**

* Real-Time Inventory Updates: Track stock levels across locations to prevent shortages or excess inventory.
* Demand Forecasting: Predict demand based on historical sales data to improve inventory and resource planning.
* Automated Order Fulfillment: Streamline order processing to assign orders to the nearest warehouse or plant for rapid dispatch.
* Enhanced Customer Satisfaction: Achieve fast and reliable delivery to improve customer loyalty and retention.

**Business Rules**

* **Inventory Replenishment:** Minimum stock levels are maintained in each warehouse to avoid stock outs.
* **Order Allocation**: Orders are fulfilled from the closest warehouse to the delivery location to reduce lead times.
* **Delivery Priority:** Perishable items (e.g., milk products) are prioritized for the quickest possible delivery.
* **Demand-Based Manufacturing:** Production schedules adapt based on demand forecasts to avoid overproduction.
* **Data Compliance:** All customer and inventory data must comply with applicable regulatory requirements.

**Background**

The company has been experiencing delays in order processing due to scattered inventory management systems and challenges with delivery timing. The current systems are not integrated, resulting in inventory visibility issues, order fulfillment delays, and inconsistent customer experiences. A centralized solution will streamline these processes, improve efficiency, and support business growth.

**Project Overview**

This project aims to develop a comprehensive software solution for managing the inventory and optimizing the delivery operations of BVG India Limited, a company specializing in ice cream and milk products. The solution will improve the accuracy of inventory management across various manufacturing plants and warehouses and will facilitate quick and efficient delivery to customers.

**Project Objective**

To design and implement a software solution that centralizes inventory management and optimizes delivery schedules, leveraging real-time data analytic to enhance operational efficiency and improve customer satisfaction.

**2. Project Scope**

**In-Scope Functionality**

* **Inventory Tracking:** Centralized real-time tracking of inventory across all plants and warehouses.
* **Demand Forecasting:** Analytic-based demand prediction to guide production and replenishment.
* **Order Fulfillment Optimization:** Automate order allocation to the closest warehouses for delivery.
* **Reporting and Analytic:** Generate insights on inventory turnover, order fulfillment times, and delivery performance.
* **Customer Notifications:** Automated updates for order status, shipment tracking, and estimated delivery times.

**Out-of-Scope Functionality**

* **Marketing and Sales Modules:** Sales campaigns, customer relationship management, and marketing tools**.**
* **Accounting and Finance Integration:** Comprehensive finance management or billing/invoicing functionality.
* **Customer Returns Management:** Handling product returns and refunds.

**3. Assumptions**

* All warehouses and plants have network access and necessary hardware to integrate with the new system.
* Employees in manufacturing and warehousing have basic computer proficiency and can be trained in the software.
* Data on current inventory, sales, and distribution is accurate and up-to-date for migration to the new system.

**4. Constraints**

* **Budget Limitations:** Budget constraints may limit the scope and functionality of the initial release.
* **System Integration:** Compatibility with existing hardware and software infrastructure at various locations.
* **Data Privacy Compliance:** Ensuring compliance with regulatory data protection standards, particularly for customer information.

**5. Risks**

1. **Business Risks**

* **Disruption of Operations:** Transitioning to the new system may cause short-term disruptions.
* **Employee Resistance:** Potential resistance from staff accustomed to existing processes.
* **Customer Dissatisfaction:** Any issues in the initial roll out could impact delivery times and customer satisfaction.

1. **Requirements Risks**

* **Changing Requirements:** Evolving business needs or market dynamics may alter the project’s objectives or requirements.
* **Incomplete Requirements:** The complexity of inventory and logistics might result in incomplete or misunderstood requirements.

1. **Other Risks**

* **Technical Risks:** Unforeseen technical issues during implementation, such as integration challenges.
* **Data Migration:** Risk of data loss or inconsistencies during the migration of legacy data.
* **Supply Chain Risks:** External factors affecting supply chain (e.g., transportation delays or resource shortages) may impact the project’s outcomes.
* **Data Privacy and Security Risks**: Unauthorized access to sensitive inventory or customer data could lead to data breaches and legal consequences.
* **User Adoption and Training Risks**: Employees may resist adopting the new system due to lack of familiarity or preference for existing processes. Insufficient training resources or time could lead to a steep learning curve and errors in early usage.
* **Regulatory and Compliance Risks**: Changes in regulatory standards (e.g., data protection laws, industry regulations) may necessitate modifications in the system.
* **Vendor Dependency Risks**: Dependence on external vendors or third-party providers (e.g., for software, hardware, or logistics) may introduce delays if they experience issues.
* **Performance Risks**: System may not meet performance expectations (e.g., speed, uptime) under high transaction volumes. Inefficient system architecture could lead to slow data processing, affecting inventory tracking and order fulfillment.
* **Maintenance and Update Risks**: Post-implementation, there may be unforeseen maintenance requirements that could interrupt operations. Delays in regular updates or failure to address bugs promptly could affect system performance.
* **Project Scope Creep Risks**: Additional requirements may emerge after the project begins, potentially expanding the project’s scope and budget. Uncontrolled scope creep may lead to timeline delays and impact project quality.

**6. Business Process Overview**

**Legacy System (AS-IS)**

1. **Inventory Management**:
   * **Manual Tracking**: Inventory levels are tracked manually or through disparate systems across each warehouse or plant, resulting in inconsistent data.
   * **Limited Visibility**: There’s no centralized system, leading to delays in stock level updates and lack of real-time inventory visibility.
   * **Stock Replenishment Delays**: Orders for replenishing stock are manually generated, often based on outdated data, leading to overstocking or under stocking.
   * **Error-Prone Processes**: High dependency on manual entry increases errors in data, making inventory records unreliable.
2. **Order Fulfillment and Delivery**:
   * **Non-Optimized Order Routing**: Orders are processed and dispatched without optimized routing, leading to increased delivery times and transportation costs.
   * **Lack of Real-Time Tracking**: There’s no way to track orders in real-time, so customers have limited visibility into delivery status.
   * **Customer Communication**: Limited automated communication with customers about order status or delivery estimates.
   * **Delivery Delays**: Due to non-optimized delivery routes and lack of streamlined processes, delivery times are often delayed.
3. **Data Analysis and Reporting**:
   * **Inconsistent Reporting**: Data is manually aggregated for reporting, leading to inconsistencies and limited insight into sales trends, inventory turnover, and delivery performance.
   * **Limited Demand Forecasting**: Forecasting is based on historical sales data without advanced analytic, leading to inaccurate demand predictions.

**Proposed Recommendations (TO-BE)**

1. **Inventory Management System**:
   * **Centralized Inventory Control**: Implement a unified system to manage inventory levels across all warehouses and plants, offering real-time visibility of stock.
   * **Automated Replenishment**: Automated stock replenishment triggers based on minimum stock thresholds to avoid stock outs or excess inventory.
   * **Bar code and RFID Tracking**: Utilize bar code or RFID technology to minimize manual entry errors and accurately track inventory at each stage.
   * **Demand Forecasting Analytic**: Integrate predictive analytic for better demand forecasting based on historical data, seasonality, and market trends.
2. **Order Fulfillment and Delivery Optimization**:
   * **Automated Order Allocation**: Assign orders to the closest warehouse or plant to the delivery location, reducing shipping time and transportation costs.
   * **Real-Time Order Tracking**: Implement real-time tracking for orders from dispatch to delivery, enhancing visibility for both internal staff and customers.
   * **Customer Notifications**: Provide automated updates for customers on order status, estimated delivery times, and real-time location tracking of shipments.
   * **Optimized Route Planning**: Use route optimization software to plan delivery routes, reducing delivery times and improving fuel efficiency.
3. **Data Analytic and Reporting**:
   * **Advanced Reporting and Insights**: Generate detailed reports on inventory, sales, and delivery performance, allowing for proactive decision-making.
   * **Customer Demand Patterns**: Analyze customer demand patterns to adapt production and inventory based on accurate forecasting models.
   * **KPI Tracking**: Define and track key performance indicators (KPIs) for inventory turnover, order processing time, and delivery speed.

**7. Project Scope:**

**In-Scope**

1. **Inventory Management Module**:
   * Track and manage inventory across manufacturing plants and warehouses.
   * Set reorder levels and notify warehouse staff for replenishment.
   * Track expiration dates for perishable goods.
   * Generate real-time inventory reports and analytic.
2. **Delivery Optimization Module**:
   * Plan optimal delivery routes based on order details and location.
   * Provide real-time delivery tracking and ETA for customers.
   * Generate delivery performance reports.

**Out of Scope**

* Integration with external logistics providers or third-party delivery platforms.
* Management of non-inventory resources, such as machinery or personnel outside the delivery and warehousing teams.

**Non - Functional Scope**

* **Scalability**: Ability to handle increasing locations, delivery zones, and orders.
* **Security**: Protect sensitive data (inventory levels, customer information) and secure system access.
* **Performance**: Real-time updates and low latency in data processing.
* **Usability**: Intuitive interface for easy navigation and user training.
* **Availability**: System uptime of 99.9% with minimal downtime during maintenance.

**8. Key Stakeholders:**

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| Project Sponsor | Provides overall project funding. |
| Business Analyst | Gathers requirements and defines scope. |
| Development Team | Builds and implements the software solution. |
| QA Team | Conducts testing to ensure quality. |
| IT Support | Maintains infrastructure and provides support. |
| End Users | Warehouse and delivery personnel, inventory managers. |

**RASCI Chart:**

Codes Used in RASCI Chart

* R = Responsible
* A = Accountable
* S = Supportive
* C = Consulted
* I = Informed

**RASCI Chart:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Project Sponsor** | **Project Manager** | **Business Analyst** | **Development Team** | **QA Team** | **End Users** |
| Requirements Gathering | C | A | R | S | I | I |
| Document Review | C | A | R | S | I | I |
| Approval | A | C | R | I | I | I |
| Development | I | A | S | R | I | I |
| Testing | I | A | S | S | R | I |
| Implementation | I | A | S | R | S | I |

**9. Key Business Requirements:**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Name** | **Description** |
| BR-01 | Real-time Inventory Tracking | Track inventory levels in real-time across all locations to maintain up-to-date stock visibility. |
| BR-02 | Automated Reorder Alerts | Trigger automatic alerts when stock levels fall below a defined threshold, ensuring timely reordering to prevent stock outs. |
| BR-03 | Inventory Categorization | Organize inventory items by type (e.g., raw materials, finished goods, perishable items) for better management and quick identification. |
| BR-04 | Expiry and Spoilage Notifications | Send automated notifications for items nearing expiry to reduce wastage and ensure freshness. |
| BR-05 | Centralized Inventory Dashboard | Provide a centralized dashboard to give a comprehensive view of stock levels, movements, and reorder status across all locations. |
| BR-06 | Demand Forecasting | Use data analytic to predict future demand, allowing proactive adjustments to inventory levels based on sales trends and seasonal fluctuations. |
| BR-07 | Delivery Route Optimization | Enable optimized route planning for delivery vehicles to reduce transportation time and costs, enhancing delivery efficiency. |
| BR-08 | Customer Delivery Notifications | Send real-time delivery updates, including estimated arrival times, to customers for improved service and satisfaction. |
| BR-09 | Performance Analytic | Monitor and analyze key performance indicators, such as inventory turnover and delivery times, to identify areas for operational improvements. |
| BR-10 | ERP System Integration | Integrate the software with the existing ERP system to facilitate data sharing between inventory, sales, and other business functions. |
| BR-11 | Stock Movement History | Maintain a history of stock movement (e.g., received, issued, transferred) to provide a clear audit trail and track item life cycle. |
| BR-12 | Resource Allocation for Delivery | Assign delivery resources optimally based on order volume, location, and priority, improving resource efficiency and delivery reliability. |
| BR-13 | Order Prioritization | Allow high-priority orders to be flagged for expedited processing and delivery to meet urgent customer requirements. |
| BR-14 | Reporting and Analytic | Generate automated and customization reports on stock status, order history, and delivery performance for data-driven decision-making. |
| BR-15 | Scalability | Design the system architecture to support scaling up with increased inventory volume, additional locations, and expanded delivery zones as the business grows. |

**10. Key Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Name** | **Description** |
| FR-01 | Login Authentication | Implement a login system with role-based authentication to control user access and ensure data security. |
| FR-02 | Real-time Inventory Update | The system should automatically update inventory levels in real-time when items are added, removed, or transferred. |
| FR-03 | Automated Stock Alerts | Generate automatic alerts to notify relevant staff when stock levels fall below the predefined reorder threshold. |
| FR-04 | Product Categorization | Allow users to categorize products by type, shelf-life, and storage location for better management and quick retrieval. |
| FR-05 | Expiry Alerts | Send automated alerts for items nearing their expiry dates to allow timely action and reduce spoilage. |
| FR-06 | Reorder Level Configuration | Enable users to set and adjust reorder levels for various products at different locations, based on demand patterns. |
| FR-07 | Route Optimization | Calculate and suggest optimal delivery routes based on real-time traffic, distance, and priority to improve delivery efficiency. |
| FR-08 | Real-Time Delivery Tracking | Provide a map-based tracking interface to show real-time progress of delivery vehicles and estimated time of arrival (ETA). |
| FR-09 | Order Status Update | Automatically update the order status in real-time, reflecting stages such as processing, in transit, and delivered. |
| FR-10 | Customer Notifications | Send order status and ETA updates to customers via SMS or email, enhancing transparency and customer satisfaction. |
| FR-11 | Demand Forecast Report Generation | Generate and display demand forecasts based on historical data, helping to maintain optimal inventory levels. |
| FR-12 | Order Prioritization | Provide an option to prioritize certain orders, marking them as high-priority to expedite processing and delivery. |
| FR-13 | Inventory Movement History | Record and display the full movement history of inventory items, including receipts, issues, and transfers, for auditing and tracking purposes. |
| FR-14 | Analytic Dashboard | Provide an analytic dashboard showing visualized KPIs like inventory turnover, stock movement, and delivery efficiency for data-driven decisions. |
| FR-15 | Data Export | Enable users to export data and reports in formats such as CSV and Excel for additional analysis and record-keeping. |

**11. Development Plan:**

* **Project Duration**: 6 months
* **Team**: Project Manager, Business Analyst, Software Engineers, QA Analysts, UX/UI Designers, Database Administrator.

**Phase timeline**:

* **Total Duration:** Approximately 6 months.

|  |  |  |
| --- | --- | --- |
| **Phase** | **Description** | **Timeline** |
| Requirements Gathering | Gather and finalize business requirements. | 4 weeks |
| System Design | Design system architecture, database, and modules. | 3 weeks |
| Development | Front end, back end, and API development. | 8 weeks |
| Testing | Functional, non-functional, and security testing. | 4 weeks |
| Deployment and Training | Deploy system and provide user training. | 2 weeks |
| Go-Live and Support | System goes live, and initial support is provided. | 1 week |

**12. Resource Plan:**

**Human Resources:-**

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibilities** | **Duration** |
| Project Manager | Oversee the project, timelines, and budgets. | Full Project |
| Business Analyst | Requirements gathering and client interactions. | Full Project |
| Developers (Front end) | UI/UX design and implementation. | 8 weeks |
| Developers (Back end) | Back end logic, API development. | 8 weeks |
| Database Administrator | Database design, maintenance, and optimization. | Full Project |
| QA Engineers | Testing and quality assurance. | 4 weeks |
| Training Specialist | Provide training to end-users. | 2 weeks |

**Technical Resources:**

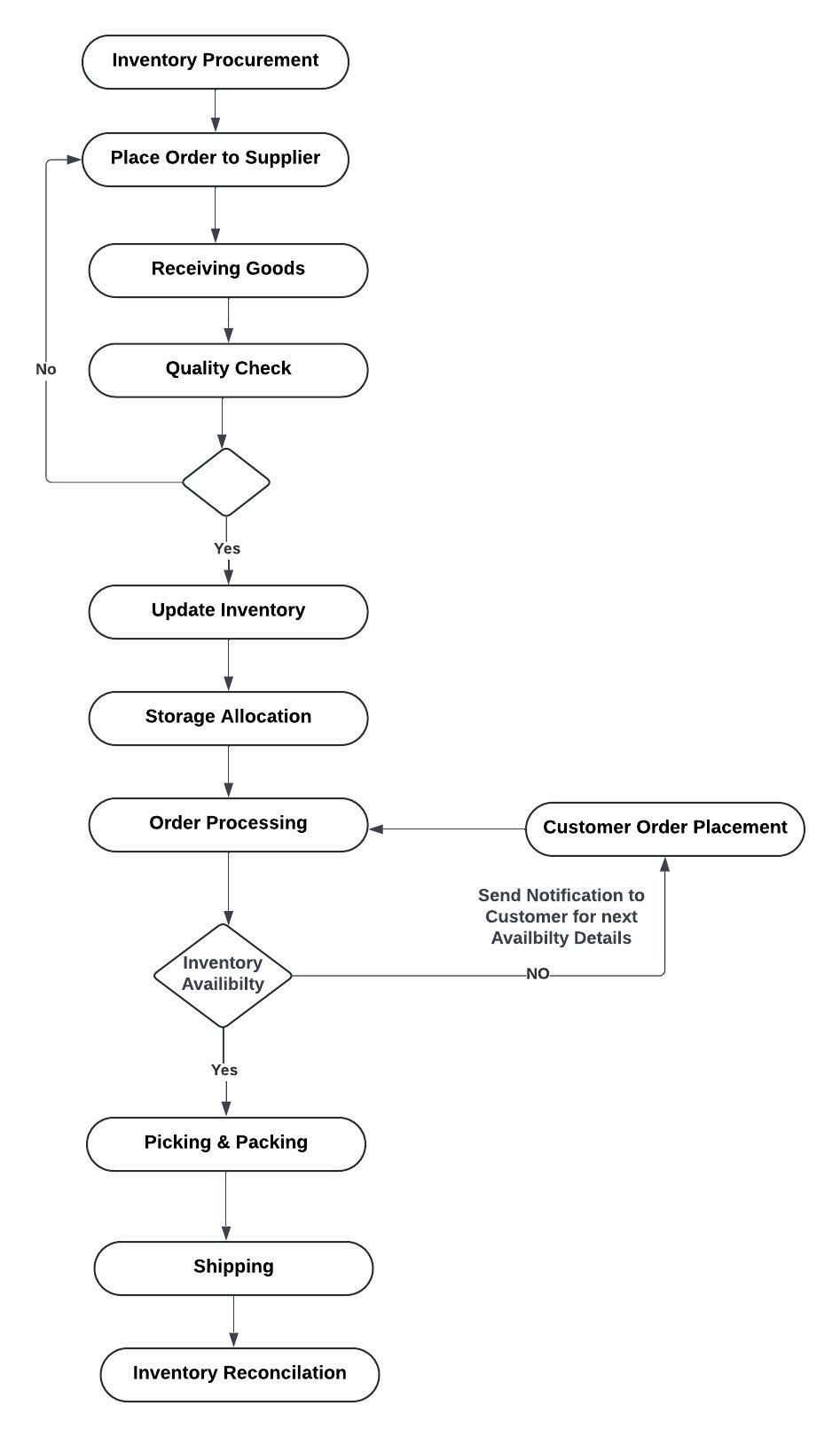
* **Software**: Database management system, development framework, API testing tools.
* **Hardware**: Servers for data processing and storage, backup systems.

**13. Risks and Mitigation**

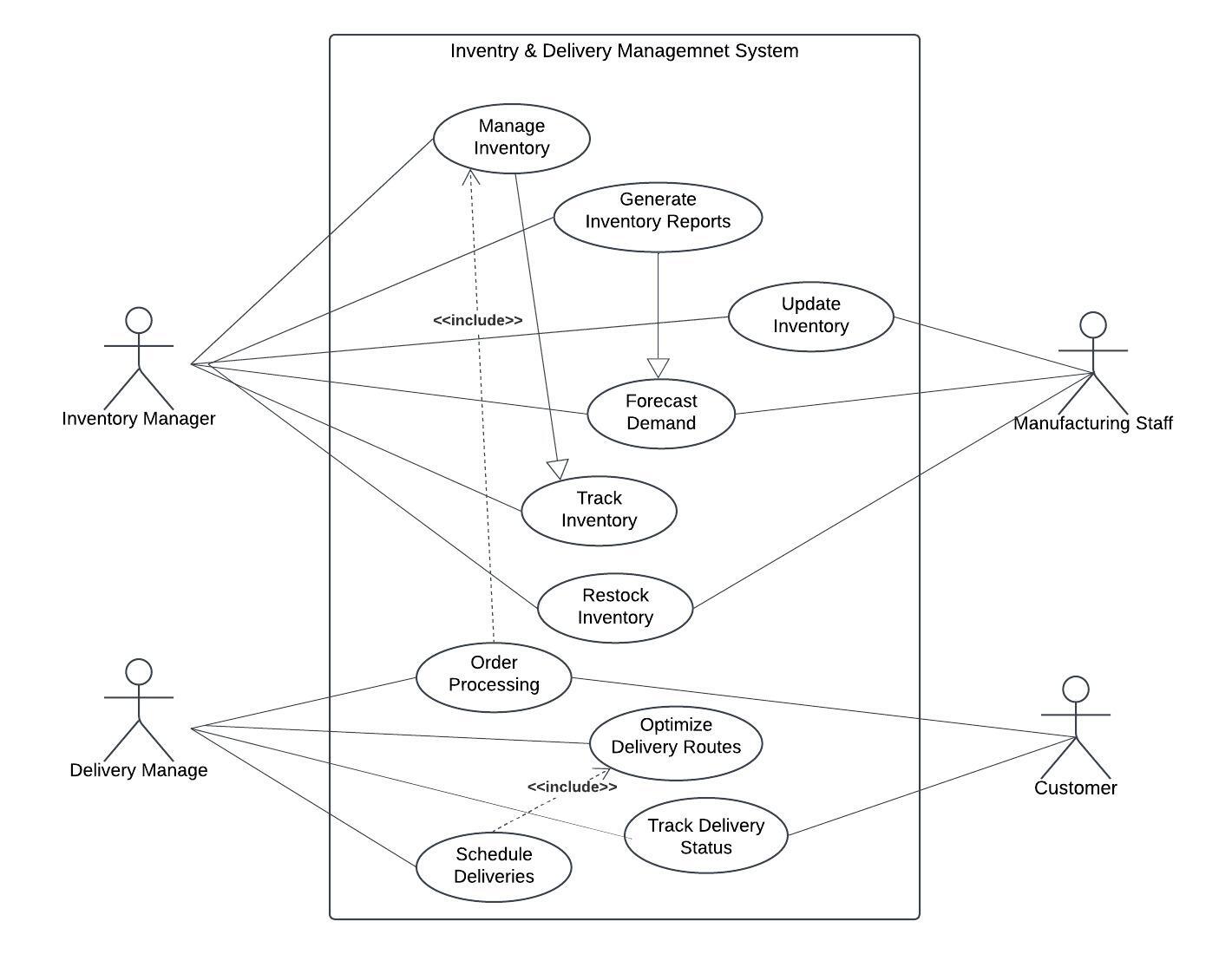
|  |  |  |
| --- | --- | --- |
| **Risk** | **Impact** | **Mitigation** |
| Requirement Misalignment | High | Regular client reviews |
| Data Security Concerns | High | Implement strict security controls |
| Delays in Development | Medium | Agile development for quick adjustments |
| User Adoption | Medium | Comprehensive training and on boarding |

**14. Approval:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Signature** | **Date** |
| Rahul Kirar | Project Sponsor | RahulK | 02/04/2024 |
| Ankita Chandravanshi | Business Analyst | AnkitaC | 02/04/2024 |

**15.Inventory Management System:**

**16. Use case Diagram:**



**Assignment 2**

**Introduction Letter to the Client**

**Dear BVG Team,**

I hope this message finds you in great spirits. My name is Diksha tupe, and I have been assigned as the Business Analyst to work with you on this project. I am genuinely excited to partner with your team and embark on a collaborative journey to identify, understand, and address your unique business needs. My role is to serve as your strategic liaison, facilitating clear communication, thorough analysis, and tailored solutions to ensure our work together is not only aligned with your objectives but also highly impact.

My first priority is to understand your business model, workflows, and the specific challenges you encounter daily. As we dive into this discovery phase, I will work closely with your team to gather insights into your operations, identify pain points, and establish a well-defined set of project goals. This initial work is essential for laying the groundwork for a solution that is effective, intuitive, and, most importantly, aligned with your vision for the future.

Our project’s objectives include designing a system that will address your current needs comprehensively. From enhancing efficiency and accuracy to automating repetitive tasks, my focus will be on ensuring that the solution is both robust and scalable. With these goals in mind, I will continuously analyze data, assess requirements, and validate processes to make sure every step of development resonates with your priorities.

I am here to facilitate and streamline the process, ensuring that we keep communication open, transparent, and adaptable. Please feel free to reach out to me at any time with questions, insights, or feedback, as your input is invaluable to our shared success. I look forward to a productive partnership with BVG, one marked by proactive collaboration and a shared commitment to excellence.

Warm regards,  
Diksha Tupe  
Business Analyst

**BRD and SRS for a Ticketing System**

**BRD: Ticketing System**

**1. Document Overview**

**Project Title:** Ticketing System  
**Prepared by:** Diksha Rajendra Tupe  
**Date:** 25 Feb 2025  
**Version:** 2.0

**2. Executive Summary:**

**Business Goals**

* **Efficient Ticket Management:** Streamline ticket creation, tracking, and resolution for faster response times.
* **Enhanced Customer Satisfaction:** Provide a system that enables quick support for customer inquiries and issues.
* **Data-Driven Insights:** Enable reporting on ticket trends and response metrics for improved operational efficiency.

**Business Objectives**

* **Automate Ticket Assignment**: Automatically assign tickets based on priority and availability.
* **Real-Time Status Updates:** Provide live tracking and updates for both customers and agents.
* **Analytic and Reporting:** Offer insights into ticket volume, resolution time, and agent performance.

**Business Rules**

* **Ticket Assignment:** Tickets are automatically routed based on predefined criteria (e.g., priority, department).
* **Priority Levels:** Each ticket is assigned a priority level (Low, Medium, High) based on predefined business rules.
* **SLA Compliance:** Each priority level has an associated SLA (e.g., High priority must be resolved within 4 hours).

**Background**

Currently, the organization uses email and spreadsheets to track support tickets, which leads to inefficiencies, delayed responses, and poor visibility into ticket status. This system aims to address these issues by providing a centralized, automated ticketing solution.

**Project Objective**

To implement a ticketing system that centralizes ticket management, automates ticket assignment, and provides real-time tracking and reporting to improve customer support efficiency and satisfaction.

**3. Key Stakeholders:**

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Role** | **Interest** |
| **Project Sponsor** | Provides funding and strategic direction | Alignment with organizational goals, ensuring ROI |
| **Project Manager** | Oversees project execution | Manages resources, timelines, and risks |
| **Business Analyst (BA)** | Conducts requirements gathering, defines scope | Ensures clear and actionable requirements are documented |
| **IT/Technical Team** | Develops and maintains technical aspects | Ensures technical feasibility, integration, and scalability |
| **Customer Support/Help Desk Team** | Main users of the system for issue tracking | Requires user-friendly system to enhance productivity and customer satisfaction |
| **Quality Assurance (QA) Team** | Tests system for bugs and requirement alignment | Ensures reliable, error-free system |
| **Data Security & Compliance Officer** | Oversees data protection and compliance | Ensures adherence to security and compliance standards |
| **Training and Support Team** | Provides user training and ongoing support | Ensures system usability for end-users and provides resources for training |
| **End-Users (Customers)** | Submits tickets or inquiries | Needs an efficient way to raise and track issue resolutions |
| **Third-Party Vendors/Software Providers** | Supplies software or integration | Ensures smooth integration and functionality with other system components |
| **Senior Management/Executive Team** | Evaluates project’s impact on business goals | Requires updates on project progress, ROI, and alignment with company objectives |
| **Legal and Compliance Advisors** | Advises on regulatory and compliance needs | Ensures system adheres to legal and regulatory standards |

**4. RASCI Chart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task** | **Responsible** | **Accountable** | **Support** | **Consulted** | **Informed** |
| Requirements Gathering | Business Analyst | Project Manager | Development Team | Stakeholders | Client |
| System Design | Architect | Project Manager | Development Team | Business Analyst | Client |
| Development | Development Team | Project Manager | Testing Team | Business Analyst | Client |
| Testing | Testing Team | QA Lead | Development Team | Business Analyst | Client |
| Deployment | DevOps Team | Project Manager | IT Support | Business Analyst | Client |
| Training | Training Team | Project Manager | Business Analyst | Stakeholders | Client |

**5. Project Scope**

**In-Scope Functionality**

* **Ticket Creation:** Allow customers to submit tickets via multiple channels (web, email).
* **Ticket Tracking:** Provide real-time status updates for each ticket.
* **Ticket Assignment:** Automatically route tickets to the appropriate department or agent.
* **Reporting and Analytic:** Generate reports on metrics like ticket resolution time and volume.
* **User Management:** Manage user roles and permissions for access control.

**Out-of-Scope Functionality**

* **Knowledge Base:** Implementing a self-help portal or FAQ for customers.
* **Billing and Invoicing:** Managing payments or customer billing within the ticketing system.
* **Live Chat Support:** Real-time chat support for immediate issue resolution.

**6. Assumptions**

* The client will provide access to existing ticketing data if migration is required.
* Users will have the necessary training and device compatibility to use the ticketing system.
* The system will operate within a secure network environment.

**7. Constraints**

* **Budget Constraints:** Limited budget may restrict additional feature enhancements.
* **Data Privacy Requirements:** System must adhere to data privacy standards to protect customer information.
* **Integration:** Must integrate with existing CRM and email systems for seamless communication.

**8. Risks**

**Business Risks**

**Adoption Risk:** Users may resist transitioning from legacy systems to the new solution.

* **Training Requirements:** Insufficient training may hinder user adoption.
* **User Preference for Legacy Systems:** Users may be more comfortable with existing systems, leading to resistance.
* **Loss of Productivity During Transition:** Learning curves and adjustments to new processes may reduce productivity temporarily.

**Operational Disruption:** Initial roll out may disrupt existing support workflows.

* Unanticipated Downtime: Downtime during deployment or system updates could impact daily operations.
* Workflow Adjustments: Existing workflows may require adaptation, causing temporary operational challenges.
* Dependency on External Systems: Integration with other systems (CRM, ERP) could be impacted, disrupting workflows.

**Data Privacy and Compliance:** New systems may introduce risks related to data privacy and regulatory compliance.

* **Regulatory Compliance**: Non-compliance with industry standards (e.g., GDPR) could lead to legal issues.
* **Data Security Concerns:** New data handling processes may introduce vulnerabilities.

**Requirements Risks**

**Changing Requirements:** Adjustments to requirements due to evolving business needs.

* Scope Creep: Frequent additions or modifications to requirements can lead to scope expansion.
* Stakeholder Alignment: Different departments or teams may have conflicting priorities or needs.
* **Rapid Market Changes:** Industry or market changes may require unexpected adjustments to requirements.

**Incomplete Requirements:** Misunderstood or missing requirements due to the complexity of current processes.

* **Miscommunication Among Stakeholders:** Lack of clear communication can result in overlooked needs.
* **Ambiguity in Requirements Documentation:** Vague or unclear requirements can cause confusion during development.
* **Dependency on Subject Matter Experts:** Limited access to experts may lead to incomplete understanding of key processes.

**Testing Limitations:** Risk of insufficient testing, which may affect requirement validation.

* **Limited Testing Scope:** Not all scenarios may be thoroughly tested, leading to potential issues post-deployment.
* **Availability of Test Data:** Unavailable or inaccurate test data may lead to incorrect requirement validation.

**Other Risks**

**Technical Risk:** Unforeseen technical challenges during development or integration.

* **Integration Complexities:** Challenges in integrating with third-party software could delay the project.
* **System Compatibility:** New systems or technologies may be incompatible with existing infrastructure.
* **Reliance on Emerging Technologies:** Utilizing new or unproven technologies may introduce unexpected issues.

**Data Migration Risks:** Potential for data loss or inconsistencies during migration from legacy systems.

* **Data Accuracy Issues:** Inaccurate mapping or cleansing may lead to data inconsistencies post-migration.
* **Legacy Data Quality:** Poor-quality data from the legacy system may complicate the migration process.
* **Incomplete Migration:** Some data may be omitted during migration due to formatting or compatibility issues.

**Scalability and Performance Risks:** The system may face performance issues as data or user volume grows.

* High Traffic Loads: Increased demand on the system could degrade performance if not properly optimized.
* Infrastructure Limitations: Existing infrastructure may be insufficient to support future growth.

**Vendor Dependence:** Dependence on third-party vendors may introduce additional risks.

* **Vendor Reliability:** Delays or failures by third-party providers could impact project timelines.
* **Cost Overruns:** Changes or additional requirements from vendors could increase project costs.

**9. Business Process Overview**

**Legacy System (AS-IS)**

* **Ticket Submission:** Customers submit tickets via email, which is manually tracked by support teams.
* **Ticket Assignment:** Manual assignment based on availability and ticket content.
* **Tracking and Updates:** Agents manually update ticket statuses in a spreadsheet, causing delays and inconsistencies.
* **Reporting:** Minimal reporting capabilities, relying on manual data extraction and analysis.

**Proposed Recommendations (TO-BE)**

* **Automated Ticketing Portal:** A centralized system where customers can directly create and track tickets.
* **Automated Ticket Routing:** Automated ticket assignment based on ticket content, agent workload, and priority.
* **Real-Time Tracking and Notifications:** Automated status updates for customers and agents.
* **Enhanced Reporting and Analytic**: Real-time insights and customization reporting for management.

**10. Business Requirements**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Name** | **Description** |
| BR-001 | User Registration | Allow users to register and create accounts within the system. |
| BR-002 | Ticket Submission | Enable users to submit support tickets through a user-friendly interface. |
| BR-003 | Ticket Tracking | Provide users with the ability to track the status of their submitted tickets. |
| BR-004 | Priority Assignment | Allow for prioritization of tickets (e.g., high, medium, low priority) based on urgency and impact. |
| BR-005 | Ticket Categorization | Support categorization of tickets (e.g., technical support, billing, general inquiry). |
| BR-006 | Ticket Status Updates | Ensure users receive real-time status updates on the progress of their tickets. |
| BR-007 | Notifications | Send automated email or SMS notifications for ticket submissions, updates, and resolutions. |
| BR-008 | Agent Assignment | Enable automatic or manual assignment of tickets to support agents based on expertise. |
| BR-009 | SLA Tracking | Track and monitor ticket resolution times according to predefined SLAs to ensure compliance. |
| BR-010 | User Dashboard | Provide a dashboard for users to view all submitted tickets, statuses, and recent interactions. |
| BR-011 | Reporting and Analytic | Generate reports and analytic on ticket volumes, resolution times, and agent performance. |
| BR-012 | Escalation Management | Provide a mechanism for users to escalate tickets that are not resolved within SLA limits. |
| BR-013 | Knowledge Base Integration | Integrate a knowledge base to help users find solutions to common issues before submitting a ticket. |
| BR-014 | Multi-Channel Support | Allow ticket submissions through multiple channels (e.g., website, email, mobile app). |
| BR-015 | Customer Feedback Collection | Enable users to provide feedback on ticket resolution and overall support experience. |

**11. Functional and Non – Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Name** | **Description** |
| FR-001 | User Authentication | The system should authenticate users upon login using username and password. |
| FR-002 | Ticket Creation | Users should be able to create and submit new tickets through a web form. |
| FR-003 | Priority Selection | Users should be able to select the priority level when submitting a ticket. |
| FR-004 | Assign Ticket to Agent | The system should assign tickets to agents automatically or allow manual assignment by managers. |
| FR-005 | Ticket Status Update | Agents should be able to update ticket statuses (e.g., Open, In Progress, Resolved). |
| FR-006 | Email Notifications | The system should send email notifications to users for ticket submissions, updates, and closures. |
| FR-007 | SLA Timer | The system should track SLA times for each ticket based on priority level and ticket type. |
| FR-008 | Escalation Workflow | Tickets should automatically escalate if SLAs are breached. |
| FR-009 | Dashboard for Users | Users should have access to a dashboard showing all their active and closed tickets. |
| FR-010 | Reporting Dashboard | Admins should be able to generate reports on ticket metrics like resolution time and volume. |
| FR-011 | Knowledge Base Access | Users should have access to a knowledge base to search for solutions before creating a ticket. |
| FR-012 | Feedback Collection | The system should allow users to provide feedback on ticket resolution. |
| FR-013 | Ticket Search and Filter | Users and agents should be able to search and filter tickets by various criteria. |
| FR-014 | Multi-Channel Ticket Submission | The system should support ticket submission via web, email, and mobile app. |
| FR-015 | Role-Based Access Control | The system should enforce role-based permissions for different user types (e.g., Admin, Agent, User). |
| NFR-001 | System Availability | The system should be available 99.9% of the time to ensure constant user access. |
| NFR-002 | Performance | The system should respond to user actions within 3 seconds. |
| NFR-003 | Scalability | The system should support up to 10,000 concurrent users. |
| NFR-004 | Security Compliance | The system should comply with GDPR and data protection regulations. |
| NFR-005 | Data Backup | The system should perform daily backups to prevent data loss. |
| NFR-006 | User-Friendly Interface | The interface should be intuitive and easy to navigate for all users. |
| NFR-007 | Reliability | The system should have a 99% uptime rate, with minimal disruptions or downtime. |
| NFR-008 | Compatibility | The system should be compatible with major browsers and mobile operating systems. |
| NFR-009 | Data Integrity | Data should be accurate and consistent across all modules of the system. |
| NFR-011 | Load Handling | The system should handle peak load without degradation in performance. |

**11. Appendices**

List of Acronyms

* **SLA:** Service Level Agreement
* **CRM:** Customer Relationship Management
* **TO-BE:** Proposed Recommendations
* **AS-IS:** Current State

Glossary of Terms

* **Ticket:** A record of a customer inquiry or issue requiring support.
* **Agent:** A staff member responsible for handling and resolving customer tickets.

Related Documents

* Project Charter
* Requirements Specification
* Project Plan

**Software Requirements Specification (SRS) for Ticketing System**

**1. Introduction:**

This SRS document outlines the requirements for developing a comprehensive **Ticketing System** designed to streamline the entire life cycle of support ticket management. The system aims to create an efficient and organized workflow for handling support requests, enabling users to submit, track, and manage their tickets easily while allowing agents to efficiently view, prioritize, respond to, and resolve incoming requests.

The Ticketing System is intended to improve communication between users and support staff, reduce response and resolution times, and ensure a structured and transparent process. Key functionalities include ticket categorization and prioritization, automated notifications, ticket history tracking, escalations, real-time communication through comments, and analytic reporting. Additionally, the system supports various roles such as **Users**, **Agents**, and **Administrators**, with permissions tailored to each role to ensure secure access to sensitive data.

By implementing these features, the Ticketing System will enhance support service quality, improve user satisfaction, and enable organizations to monitor performance metrics, such as ticket response and resolution times, for continuous improvement. This document provides detailed specifications for the system’s functional and non-functional requirements, along with user stories, use cases, acceptance criteria, and expected outcomes to guide the development and implementation of a robust support ticketing solution.

**2. Goals:**

* Improve customer satisfaction by providing an organized and responsive system for issue resolution.
* Increase efficiency for support teams by automating ticket assignment, tracking, and notification processes.
* Enhance reporting capabilities to monitor support performance and identify areas for improvement.

**3. Objectives:**

* Allow users to submit and track support tickets.
* Facilitate efficient assignment and management of tickets by support agents.
* Enable categorization and prioritization of tickets based on urgency and type.
* Provide notifications to users and agents regarding ticket status updates.
* Offer reporting and analytic for ticket metrics such as resolution time and agent performance.

**4. Use Case Specifications:**

**Use Case 1: User Registration and Login**

* **User Story ID:** US01
* **Task**: Create an account, verify email, and log in.
* **Priority**: High
* **Value Statement:**
  + As a visitor,
  + I want to register and log in,
  + So that I can create, track, and manage support tickets.
* **Precondition:** User must have a valid email or phone number.
* **Basic Flow:**
  1. User navigates to the registration page.
  2. User enters details like name, email, password, and optionally phone number.
  3. System sends a verification email or OTP.
  4. User verifies email/phone number to activate the account.
  5. User logs in to access the dashboard.
* **Alternate Flow:**
  + If the email/phone verification fails, the user is prompted to retry verification.
* **BV**: 5
* **CP:** 4
* **Acceptance Criteria:**
  + Users can register and log in with verified email/phone.
  + Users are redirected to their dashboard after login.
* **Post condition:** User has access to the ticketing system with an active account.

**Use Case 2: Ticket Submission**

* **User Story ID:** US02
* **Task:** Submit a new support ticket.
* **Priority**: High
* **Value Statement:**
  + As a registered user,
  + I want to submit a support ticket,
  + So that I can receive assistance on an issue.
* **Precondition:** User is logged into the system.
* **Basic Flow:**
  1. User navigates to the "Submit Ticket" page.
  2. User enters ticket details, selects category, priority level, and optionally attaches files.
  3. User submits the ticket.
  4. System generates a Ticket ID and assigns it to an available agent.
* **Alternate Flow:**
  + If required information is missing, the user is prompted to complete the form before submission.
* **BV**: 5
* **CP**: 4
* **Acceptance Criteria:**
  + Users can submit tickets with required details.
  + System provides a confirmation and Ticket ID after successful submission.
* **Post condition:** Ticket is created and assigned, and the user is notified of successful submission.

**Use Case 3:** Ticket Assignment and Response

* **User Story ID:** US03
* **Task**: Assign ticket to agent, respond to user.
* **Priority**: High
* **Value Statement:**
  + As an agent,
  + I want to view and respond to assigned tickets,
  + So that I can assist users with their issues promptly.
* **Precondition:** Agent is logged into the system.
* **Basic Flow:**
  1. Agent logs in to the dashboard and views assigned tickets.
  2. Agent opens a ticket to review details.
  3. Agent responds with a message or requests additional information from the user.
* **Alternate Flow:**
  + If an agent cannot resolve the issue, the ticket is escalated to a specialist.
* **BV**: 4
* **CP**: 4
* **Acceptance Criteria:**
  + Agents can view assigned tickets and respond to users.
  + Tickets can be escalated if needed.
* **Post condition:** User is notified of agent response or ticket escalation.

**Use Case 4:** **Ticket Status Update and Notification**

* **User Story ID:** US04
* **Task**: Update ticket status and notify users.
* **Priority**: Medium
* **Value Statement:**
  + As an agent,
  + I want to update the status of a ticket,
  + So that the user is informed about the ticket’s progress.
* **Precondition:** A ticket exists, and an agent is assigned to it.
* **Basic Flow:**
  1. Agent changes the ticket status to "In Progress," "Resolved," or "Closed."
  2. System sends a notification to the user about the updated status.
* **Alternate Flow:** If the status update fails, the agent receives an error message and retries.
* **BV**: 3
* **CP**: 3
* **Acceptance Criteria:**
  + Status changes are reflected in the system.
  + Notifications are sent to users for each status update.
* **Post condition**: The user is informed of the updated ticket status.

**Use Case 5: Ticket Closure and Feedback Collection**

* **User Story ID:** US05
* **Task**: Close ticket and collect feedback.
* **Priority**: Medium
* **Value Statement:**
  + As an agent,
  + I want to close tickets after resolution and collect feedback,
  + So that we can gauge user satisfaction.
* **Precondition**: Ticket is resolved.
* **Basic Flow:**
  1. Agent marks the ticket as "Closed."
  2. System sends a feedback form link to the user.
  3. User completes the feedback form.
* **Alternate Flow:**
  + If the user does not complete the feedback, the system sends a reminder.
* **BV**: 4
* **CP**: 3
* **Acceptance Criteria:**
  + Ticket is marked as "Closed" after resolution.
  + Users receive a feedback request, and responses are recorded.
* **Post condition:** Ticket is fully closed, and user feedback is collected.

**Use Case 6: Ticket Categorization and Priority Assignment**

* **User Story ID:** US06
* **Task**: Assign category and priority to tickets.
* **Priority**: Medium
* **Value Statement:**
  + As a user,
  + I want to select a category and priority level when submitting a ticket,
  + So that my issue is assigned to the appropriate team and resolved promptly.
* **Precondition**: User is logged in and ready to submit a ticket.
* **Basic Flow:**
  1. User navigates to the "Submit Ticket" form.
  2. User selects a category (e.g., Technical, Billing, General).
  3. User selects a priority level (e.g., High, Medium, Low).
  4. System assigns the ticket to the relevant team based on category and priority.
* **Alternate Flow:**
  + If a category or priority is not selected, the system prompts the user to make a selection.
* **BV**: 4
* **CP**: 3
* **Acceptance Criteria:**
  + Users can categorize and prioritize their tickets during submission.
  + Tickets are routed to the appropriate support team based on category.
* **Post condition:** Ticket is correctly categorized and assigned priority, helping ensure timely resolution.

**Use Case 7: Ticket Escalation**

* **User Story ID:** US07
* **Task**: Escalate unresolved tickets to a higher support level.
* **Priority**: High
* **Value Statement:**
  + As an agent,
  + I want to escalate tickets that require higher expertise,
  + So that complex issues are resolved effectively.
* **Precondition**: A ticket is assigned to an agent but cannot be resolved at the current support level.
* **Basic Flow:**
  1. Agent reviews the ticket and determines escalation is needed.
  2. Agent selects an "Escalate" option.
  3. System routes the ticket to a higher support level and notifies relevant personnel.
* **Alternate Flow:** If the escalation fails, the system prompts the agent to try again or contact support.
* **BV**: 5
* **CP**: 4
* **Acceptance Criteria:**
  + Agents can escalate tickets to higher support levels.
  + Escalated tickets are reassigned and prioritized for specialized handling.
* **Post condition**: Ticket is reassigned to a higher level for expert resolution.

**Use Case 8: Commenting on Tickets**

* **User Story ID:** US08
* **Task**: Add comments to tickets.
* **Priority**: Medium
* **Value Statement:**
  + As a user or agent,
  + I want to add comments to tickets,
  + So that I can provide additional information or updates.
* **Precondition**: User or agent is logged in and viewing an existing ticket.
* **Basic Flow:**
  1. User or agent navigates to the ticket details.
  2. User or agent enters a comment in the comment field.
  3. System updates the ticket with the new comment and notifies relevant parties.
* **Alternate Flow:** If comment submission fails, the system notifies the user or agent to retry.
* **BV**: 4
* **CP**: 3
* **Acceptance Criteria:**
  + Users and agents can add comments to tickets.
  + Notifications are sent to relevant parties when comments are added.
* **Post condition**: Ticket contains updated information from user or agent comments.

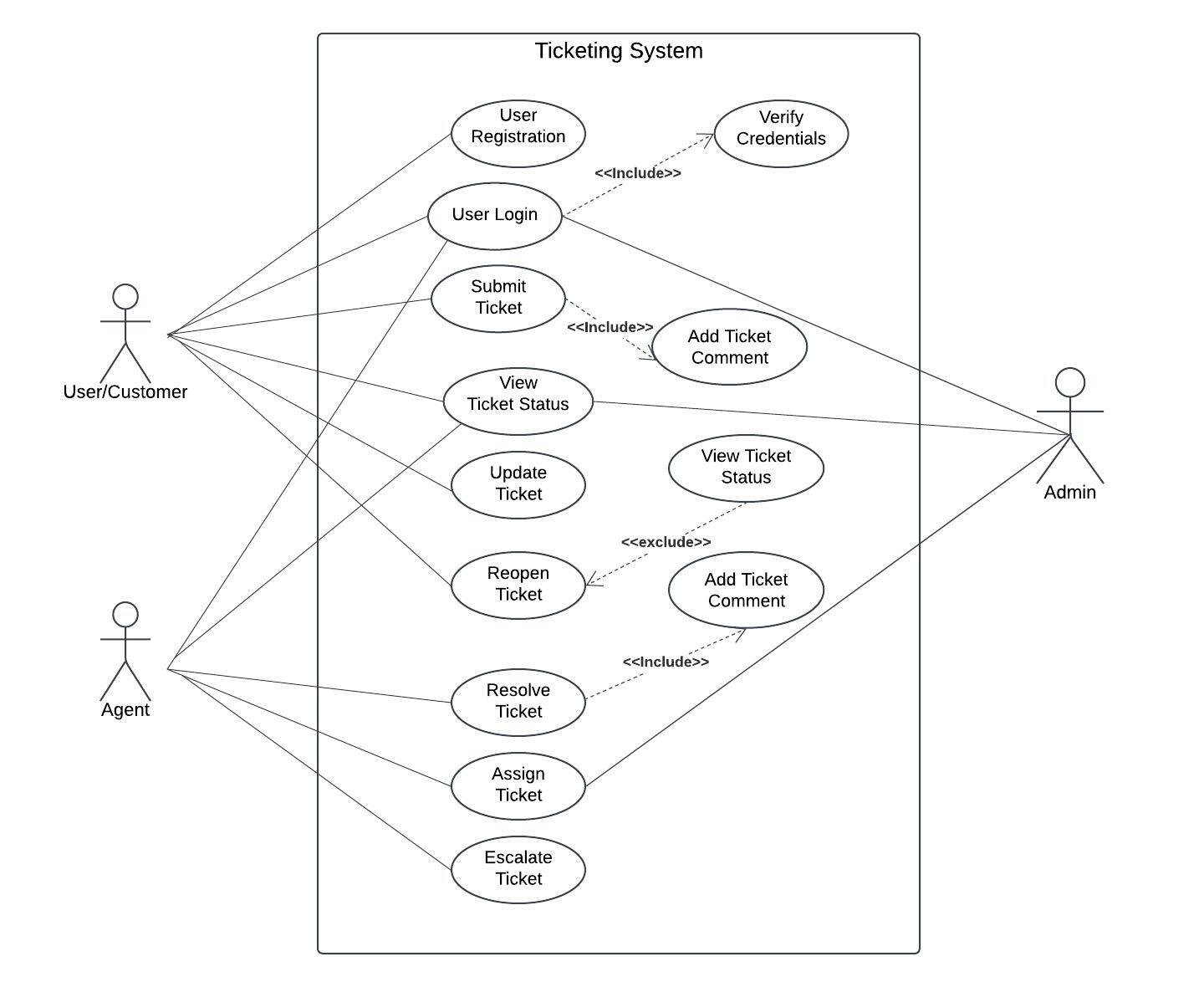
**Use Case 9: Viewing Ticket History**

* **User Story ID:** US09
* **Task**: View history of a ticket.
* **Priority**: Medium
* **Value Statement:**
  + As a user or agent,
  + I want to view the history of actions taken on a ticket,
  + So that I can understand the progress and previous responses.
* **Precondition**: User or agent is logged in and has access to a specific ticket.
* **Basic Flow:**
  1. User or agent navigates to the ticket details page.
  2. System displays the history of actions taken on the ticket, including status changes, comments, and agent responses.
* **Alternate Flow:** If the ticket history is inaccessible, the system displays an error message.
* **BV**: 4
* **CP**: 3
* **Acceptance Criteria:**
  + Users and agents can view a chronological history of all actions and responses on a ticket.
* **Post condition**: Users and agents are informed of all actions taken on the ticket.

**Use Case 10: Reporting and Analytic**

* **User Story ID:** US10
* **Task**: Generate reports on ticket metrics.
* **Priority:** Medium
* **Value Statement:**
  + As a system administrator,
  + I want to generate reports on ticket metrics,
  + So that I can analyze performance and identify areas for improvement.
* **Precondition:** Administrator is logged in with report access privileges.
* **Basic Flow:**
  1. Administrator navigates to the reporting section.
  2. Administrator selects desired metrics (e.g., resolution time, ticket volume, agent performance).
  3. System generates a report based on selected metrics.
* **Alternate Flow:**
  + If the report generation fails, the system prompts the administrator to retry or adjust parameters.
* **BV**: 3
* **CP**: 3
* Acceptance Criteria:
  + Administrators can generate reports based on specified metrics.
  + Reports display accurate data in a user-friendly format.
* Post condition: Administrator receives insights on ticket management performance and can take action based on analytic.

**5. Use Case Diagram:**

****

**6. Functional and Non – Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement Name** | **Description** |
| FR-001 | User Authentication | The system should authenticate users upon login using username and password. |
| FR-002 | Ticket Creation | Users should be able to create and submit new tickets through a web form. |
| FR-003 | Priority Selection | Users should be able to select the priority level when submitting a ticket. |
| FR-004 | Assign Ticket to Agent | The system should assign tickets to agents automatically or allow manual assignment by managers. |
| FR-005 | Ticket Status Update | Agents should be able to update ticket statuses (e.g., Open, In Progress, Resolved). |
| FR-006 | Email Notifications | The system should send email notifications to users for ticket submissions, updates, and closures. |
| FR-007 | SLA Timer | The system should track SLA times for each ticket based on priority level and ticket type. |
| FR-008 | Escalation Workflow | Tickets should automatically escalate if SLAs are breached. |
| FR-009 | Dashboard for Users | Users should have access to a dashboard showing all their active and closed tickets. |
| FR-010 | Reporting Dashboard | Admins should be able to generate reports on ticket metrics like resolution time and volume. |
| FR-011 | Knowledge Base Access | Users should have access to a knowledge base to search for solutions before creating a ticket. |
| FR-012 | Feedback Collection | The system should allow users to provide feedback on ticket resolution. |
| FR-013 | Ticket Search and Filter | Users and agents should be able to search and filter tickets by various criteria. |
| FR-014 | Multi-Channel Ticket Submission | The system should support ticket submission via web, email, and mobile app. |
| FR-015 | Role-Based Access Control | The system should enforce role-based permissions for different user types (e.g., Admin, Agent, User). |
| NFR-001 | System Availability | The system should be available 99.9% of the time to ensure constant user access. |
| NFR-002 | Performance | The system should respond to user actions within 3 seconds. |
| NFR-003 | Scalability | The system should support up to 10,000 concurrent users. |
| NFR-004 | Security Compliance | The system should comply with GDPR and data protection regulations. |
| NFR-005 | Data Backup | The system should perform daily backups to prevent data loss. |
| NFR-006 | User-Friendly Interface | The interface should be intuitive and easy to navigate for all users. |
| NFR-007 | Reliability | The system should have a 99% uptime rate, with minimal disruptions or downtime. |
| NFR-008 | Compatibility | The system should be compatible with major browsers and mobile operating systems. |
| NFR-009 | Data Integrity | Data should be accurate and consistent across all modules of the system. |
| NFR-011 | Load Handling | The system should handle peak load without degradation in performance. |

**ERD for Support Ticketing System:**

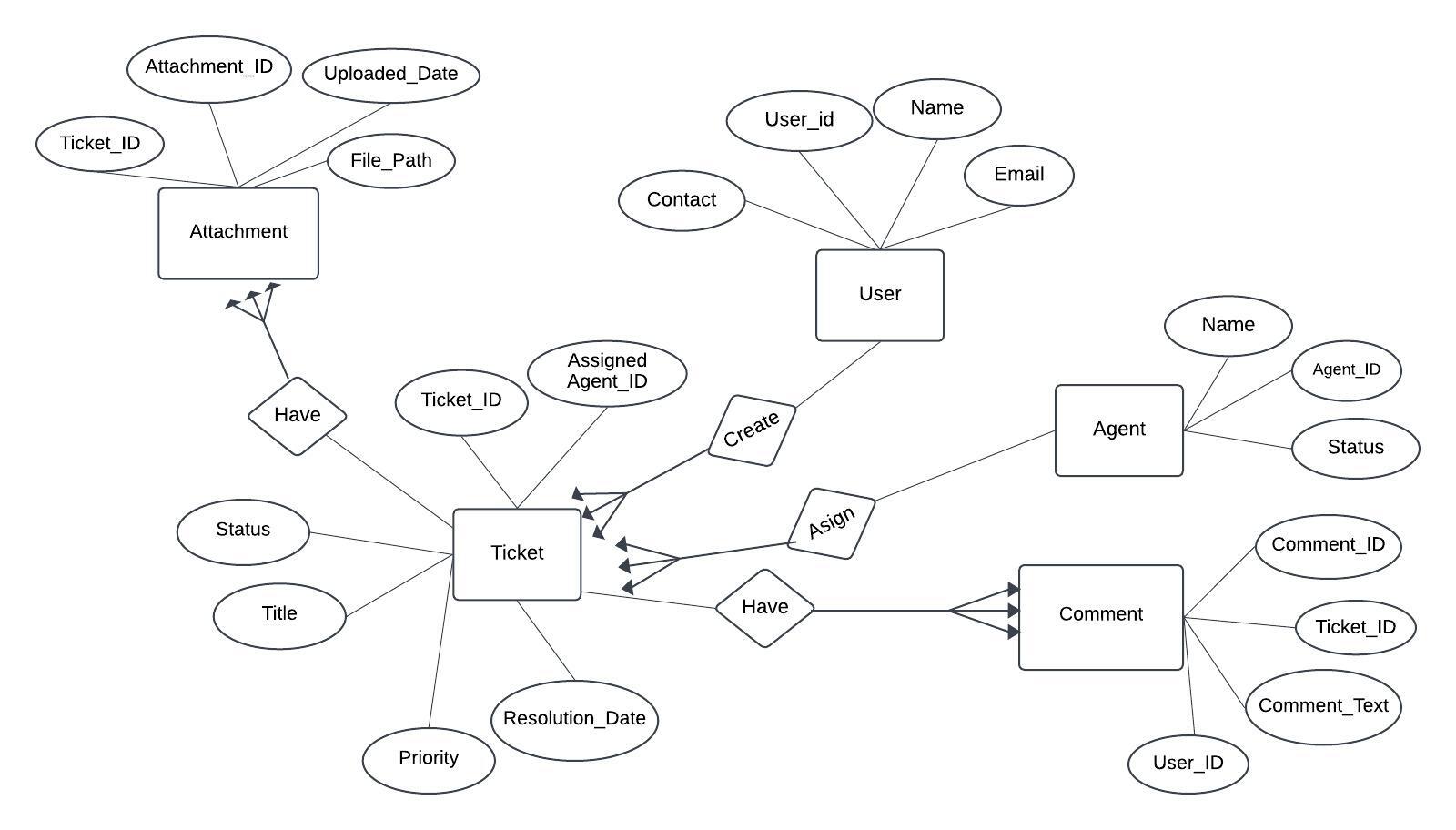
**Entities and Attributes**

1. **Ticket**
   * Ticket\_ID (Primary Key)
   * Title
   * Description
   * Created\_Date
   * Status (e.g., Open, In Progress, Resolved, Closed)
   * Priority (e.g., Low, Medium, High)
   * Resolution\_Date
   * Category\_ID (Foreign Key)
   * User\_ID (Foreign Key)
   * Assigned\_Agent\_ID (Foreign Key)
2. **User**
   * User\_ID (Primary Key)
   * Name
   * Email
   * Contact\_Number
   * User\_Type (e.g., Customer, Agent, Admin)
3. **Agent**
   * Agent\_ID (Primary Key)
   * Name
   * Email
   * Department
   * Role (e.g., Support Agent, Specialist)
   * Status (e.g., Active, Inactive)
4. **Category**
   * Category\_ID (Primary Key)
   * Category\_Name (e.g., Technical Support, Billing, General Inquiry)
   * Description
5. **Comment**
   * Comment\_ID (Primary Key)
   * Ticket\_ID (Foreign Key)
   * User\_ID (Foreign Key)
   * Comment\_Text
   * Created\_Date
6. **Attachment**
   * Attachment\_ID (Primary Key)
   * Ticket\_ID (Foreign Key)
   * File\_Path
   * Uploaded\_Date

**7. Relationships**

* **User to Ticket**: A user can create multiple tickets (One-to-Many), but each ticket is created by one user.
* **Agent to Ticket:** An agent can be assigned to multiple tickets (One-to-Many), but each ticket is assigned to only one agent.
* **Category to Ticket:** Each ticket belongs to one category (One-to-One or One-to-Many, depending on requirements).
* **Ticket to Comment:** A ticket can have multiple comments (One-to-Many).
* **Ticket to Attachment:** A ticket can have multiple attachments (One-to-Many).

**ER Diagram for creating a support ticket/Ticketing life cycle:**



1. **User story of shopping from e commerce.**

|  |  |
| --- | --- |
| User Story 1 | |
| As a user, I want to create an account so that I can have a personalized experience. | |
| BV: 1000 | CP: 02 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  Registration Screen  Text boxes for username, password, mobile number, e-mail ID, etc.  Click on Register button.  Send notification to user for successful registration | |

|  |  |
| --- | --- |
| User Story 2 | |
| As a user, I want to log in to my account so that I can access my personal data and preferences. | |
| BV: 1000 | CP: 02 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  Sign-in screen.  Text boxes for username and password  Click on Sign-in button.  Application services displayed to the user | |

|  |  |
| --- | --- |
| User Story 3 | |
| As a user, I want to search for restaurants by location so that I can order from nearby places. | |
| BV: 500 | CP: 02 |
| Tasks: 2 | Priority: High |
| Acceptance Criteria:  Display search by location option.  Display list of restaurants as per desired location  Suggest nearby restaurants as per GPS-detected user location | |

|  |  |
| --- | --- |
| User Story 4 | |
| As a user, I want to browse menus of different restaurants so that I can see what food options are available. | |
| BV: 1000 | CP: 03 |
| Tasks: 2 | Priority: High |
| Acceptance Criteria:  Display menus page by page of the selected restaurant.  Display prices of all the food items in the menu | |

|  |  |
| --- | --- |
| User Story 5 | |
| As a user, I want to add items to my cart so that I can prepare my order. | |
| BV: 500 | CP: 02 |
| Tasks: 2 | Priority: High |
| Acceptance Criteria:  Add to cart option.  Display Remove from cart option | |

|  |  |
| --- | --- |
| User Story 6 | |
| As a user, I want to customize my order (e.g., add extras, specify allergies) so that the food suits my preferences. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: Moderate |
| Acceptance Criteria:  Customization of selected food items in the cart  Display extras as per restaurant and dishes.  Textboxes for specifying allergies, spice preferences, etc. | |

|  |  |
| --- | --- |
| User Story 7 | |
| As a user, I want to check out and place my order so that the food gets delivered to me. | |
| BV: 1000 | CP: 03 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  Checkout option  Display final list of selected food items for review, including items, quantities and their prices.  Redirect to payment gateway screen | |

|  |  |
| --- | --- |
| User Story 8 | |
| As a user, I want to choose my payment method (e.g., credit card, cash on delivery) at checkout so that I can pay in the way that’s most convenient for me. | |
| BV: 1000 | CP: 05 |
| Tasks: 4 | Priority: High |
| Acceptance Criteria:  Payment gateway screen  Select a payment method from the displayed options (UPI/Net banking/Debit card/etc.)  Confirm payment.  User receives order confirmation via message/e-mail once payment is successful | |

|  |  |
| --- | --- |
| User Story 9 | |
| As a user, I want to see the estimated delivery time after placing my order so that I know when to expect my food. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: Moderate |
| Acceptance Criteria:  Display estimated delivery time once payment is successful.  Estimated delivery time clearly specified in order details.  Option to track the order.  Notification for updated delivery time when the delivery personnel pick up the order.  Notification to delivery personnel for estimated delivery time, so that he/she can manage their task in a timely manner | |

|  |  |
| --- | --- |
| User Story 10 | |
| As a user, I want to track my order in real-time so that I can see the progress of my delivery. | |
| BV: 100 | CP: 01 |
| Tasks: 3 | Priority: Moderate |
| Acceptance Criteria:  Track order screen.  Notification about the order accepted by restaurant.  Notify about the food being prepared.  Notification when delivery personnel pick up the order.  Display estimated time for delivery with a timer | |

|  |  |
| --- | --- |
| User Story 11 | |
| As a user, I want to rate and review the restaurant and delivery service after receiving my order so that I can share my experience with others. | |
| BV: 500 | CP: 02 |
| Tasks: 2 | Priority: Low |
| Acceptance Criteria:  View order history and select specific order for rating and reviewing.  Rate their experience on a scale of 1 to 5 stars.  Text boxes to write opinions on what the user liked/did not like.  Navigate through other reviews and ratings from other users | |

|  |  |
| --- | --- |
| User Story 12 | |
| As a user, I want to save my favourite restaurants so that I can easily find them later. | |
| BV: 100 | CP: 01 |
| Tasks: 3 | Priority: Low |
| Acceptance Criteria:  Navigate through the restaurant’s profile.  Select the option to save the restaurant as a favourite.  Restaurant added to their list of favourite restaurants.  Navigate through the list of user’s favourite restaurants.  Option to remove the previously saved restaurant from the favourite list | |

|  |  |
| --- | --- |
| User Story 13 | |
| As a user, I want to see special offers and promotions so that I can save money on my orders. | |
| BV: 500 | CP: 03 |
| Tasks: 3 | Priority: Low |
| Acceptance Criteria:  Display special offers and promotions on the home screen.  Offers clearly marked with the discount amount or deal details.  Each offer should have a clear call-to-action (e.g., “Order Now”, “Get Deal”)  Display expiry date for each offer.  User should be able to tap on an offer to see more details.  Allow users to apply the offer to their order during checkout | |

|  |  |
| --- | --- |
| User Story 14 | |
| As a user, I want to contact customer service in case there are any issues with my order. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: Moderate |
| Acceptance Criteria:  Dedicated section or button for contacting customer service.  Upon clicking the button, users should be presented with contact options such as live chat, email, or phone call.  Live chat option should connect the user to a customer service representative within a reasonable time.  App should provide an estimated response time for each contact method.  Users should receive a confirmation (like an email or an in-app notification) after their query has been received | |

|  |  |
| --- | --- |
| User Story 15 | |
| As a restaurant owner, I want to register my restaurant so that I can start receiving orders through the app. | |
| BV: 1000 | CP: 02 |
| Tasks: 4 | Priority: High |
| Acceptance Criteria:  Clear option for restaurant owners to register their restaurant.  Registration form should ask for necessary details such as restaurant name, address, contact information, cuisine type, etc.  App should allow the restaurant owner to upload a menu with prices.  App should verify the authenticity of the restaurant through an approval process.  Once approved, the restaurant should appear in the list of available restaurants on the app | |

|  |  |
| --- | --- |
| User Story 16 | |
| As a restaurant owner, I want to manage the menu (e.g., add new dishes, update prices) so that users see up-to-date information. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  App should allow the restaurant owner to upload a menu with prices.  Restaurant owner should be able to update their menu and other details after registration (such as add new dishes, update prices etc.) | |

|  |  |
| --- | --- |
| User Story 17 | |
| As a restaurant owner, I want to receive notifications about new orders so that we can start preparing them right away. | |
| BV: 500 | CP: 02 |
| Tasks: 5 | Priority: High |
| Acceptance Criteria:  App should send a notification to the restaurant owner as soon as a new order is placed.  Notification should include important details about the order such as the items ordered, quantity, customer name, and delivery address.  App should provide an option for the restaurant owner to accept or reject the order.  Restaurant owner should be able to contact the customer through the app in case of any issues or clarifications needed regarding the order | |

|  |  |
| --- | --- |
| User Story 18 | |
| As a restaurant owner, I want to update the status of orders (e.g., in preparation, ready for pickup) so that users and delivery personnel are informed. | |
| BV: 500 | CP: 02 |
| Tasks: 2 | Priority: Moderate |
| Acceptance Criteria:  Upon accepting the order, the app should provide an estimated time for food preparation and delivery.  App should update the customer about the status of their order (e.g., preparing, ready for pickup, out for delivery) based on updates from the restaurant owner.  App should update the delivery personnel to pick-up the prepared food order | |

|  |  |
| --- | --- |
| User Story 19 | |
| As a restaurant owner, I want to respond to reviews from users so that we can improve our service or clarify any misunderstandings. | |
| BV: 500 | CP: 03 |
| Tasks: 1 | Priority: Low |
| Acceptance Criteria:  App should notify the restaurant owner when a new review is posted.  Restaurant owner should be able to view all reviews for their restaurant in a dedicated section of the app.  Each review should display the user’s name, rating, and comments.  App should provide an option for the restaurant owner to respond to each review.  App should allow the restaurant owner to edit or delete their responses | |

|  |  |
| --- | --- |
| User Story 20 | |
| As a delivery person, I want to register as available for deliveries so that the system assigns orders to me. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  App should provide a clear option for delivery personnel to register their availability.  Registration process should ask for necessary details such as name, contact information, vehicle details, etc.  App should allow the delivery personnel to set their working hours.  App should verify the authenticity of the delivery personnel through an approval process.  Once approved, the delivery personnel should start receiving notifications about new orders | |

|  |  |
| --- | --- |
| User Story 21 | |
| As a delivery person, I want to see the pickup and drop-off locations for assigned orders so that I know where to go. | |
| BV: 500 | CP: 03 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  App should display the pickup location (restaurant address) and drop-off location (customer address) for each assigned order.  App should provide clear directions from the current location of the delivery personnel to the pickup location.  App should update the directions in real-time to account for traffic and other factors.  App should allow the delivery personnel to mark an order as picked up or delivered | |

|  |  |
| --- | --- |
| User Story 22 | |
| As a delivery person, I want to update the status of deliveries (e.g., picked up, delivered) so that users and restaurants are informed. | |
| BV: 500 | CP: 02 |
| Tasks: 4 | Priority: High |
| Acceptance Criteria:  App should provide an option for delivery personnel to update the status of their assigned orders.  Status options should include at least ‘picked up’ and ‘delivered’.  App should record the time when the status is updated.  App should allow the delivery personnel to add notes or comments when updating the status (e.g., reason for delay)  App should maintain a history of all status updates for each order | |

|  |  |
| --- | --- |
| User Story 23 | |
| As a user, I want to reorder my previous orders so that I can save time on placing the same order again. | |
| BV: 500 | CP: 01 |
| Tasks: 2 | Priority: High |
| Acceptance Criteria:  App should provide an option for users to view their order history.  Each previous order in the history should include details such as restaurant name, items ordered, and total price.  App should provide an option to reorder the same items from a previous order.  Upon selecting ‘reorder’, the app should add the same items to the user’s current cart.  Allow modifications in these reorders.  App should allow the user to proceed to checkout after reordering | |

|  |  |
| --- | --- |
| User Story 24 | |
| As a user, I want to schedule my orders so that I can plan my meals in advance. | |
| BV: 500 | CP: 02 |
| Tasks: 4 | Priority: Moderate |
| Acceptance Criteria:  App should provide an option for users to schedule orders in advance.  Users should be able to select the restaurant, menu items, and quantity for the scheduled order.  App should allow users to select a future date and time for the delivery of the scheduled order.  App should send a notification to the user 30 minutes before the scheduled order is sent to the restaurant.  App should update the user on the status of their scheduled order (e.g., preparing, out for delivery) as it would for regular orders | |

|  |  |
| --- | --- |
| User Story 25 | |
| As a user, I want to see the nutritional information of the dishes so that I can make informed food choices. | |
| BV: 100 | CP: 01 |
| Tasks: 2 | Priority: Low |
| Acceptance Criteria:  App should display nutritional information for each dish on its detail page.  Nutritional information should include details such as calories, protein, fat, carbohydrates, and allergen information.  App should source this information from reliable and verifiable sources.  Users should be able to view this information before adding the dish to their cart.  App should provide a disclaimer that nutritional information may vary based on portion size and preparation | |

|  |  |
| --- | --- |
| User Story 26 | |
| As a user, I want to share my order details with others so that they can also see what I ordered. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: Moderate |
| Acceptance Criteria:  App should provide an option for users to share their order details.  Users should be able to select the contact(s) with whom they want to share the order details.  App should send a notification or message to the selected contact(s) with a link to view the order details.  Contacts who receive the shared order link should be able to view the order details without needing to log in to the app.  Shared order details should include information such as restaurant name, items ordered, total price, and estimated delivery time | |

|  |  |
| --- | --- |
| User Story 27 | |
| As a user, I want to split the bill with my friends so that we can share the cost of the order. | |
| BV: 100 | CP: 05 |
| Tasks: 4 | Priority: High |
| Acceptance Criteria:  App should provide an option for users to create a group order.  Users should be able to invite their friends to the group order by sharing a link or through in-app connections.  Each friend should be able to add their own items to the group order.  App should keep track of which items were added by which friend.  At checkout, the app should automatically calculate each friend’s total based on the items they added.  App should allow each friend to pay for their part of the bill separately | |

|  |  |
| --- | --- |
| User Story 28 | |
| As a user, I want to receive notifications about the status of my order so that I can stay updated. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  App should send a notification to the user when the order status changes.  The notification should include the new status of the order and any relevant details (e.g., estimated delivery time for ‘out for delivery’ status)  App should provide real-time updates on the status of the order, such as ‘preparing’, ‘out for delivery’, and 'delivered’.  App should keep a history of all status updates for each order.  If there is a significant delay or issue with the order, the app should send a notification to the user explaining the situation | |

|  |  |
| --- | --- |
| User Story 29 | |
| As a user, I want to cancel my order if I change my mind before it’s prepared. | |
| BV: 1000 | CP: 03 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  App should provide an option for users to cancel their order.  Users should be able to cancel their order at any stage before the restaurant starts preparing the food.  App should ask the user to confirm their decision before canceling the order.  Upon cancellation, the app should update the order status to 'cancellation.  App should send a notification or email to the user confirming the cancellation and refund details | |

|  |  |
| --- | --- |
| User Story 30 | |
| As a restaurant owner, I want to see analytic about the orders (e.g., most ordered dishes, peak times) so that I can improve my service. | |
| BV: 100 | CP: 02 |
| Tasks: 3 | Priority: Low |
| Acceptance Criteria:  App should provide a dashboard or similar feature for restaurant owners to view order analytic.  Dashboard should display key metrics such as total orders, average order value, most ordered items, peak order times, etc.  Dashboard should provide visual representations (e.g., graphs, charts) of the data for easy understanding.  App should update the analytic in real-time as new orders are received.  App should ensure the privacy and security of the analytic data | |

|  |  |
| --- | --- |
| User Story 31 | |
| As a restaurant owner, I want to offer deals and promotions so that I can attract more customers. | |
| BV: 500 | CP: 02 |
| Tasks: 2 | Priority: Moderate |
| Acceptance Criteria:  App should provide an option for restaurant owners to create deals and promotions.  Restaurant owners should be able to set the terms of the deal or promotion, such as the discount amount, eligible items, and validity period.  Users should be able to see the details of the deal or promotion before placing an order.  App should automatically apply the deal or promotion to eligible orders during checkout | |

|  |  |
| --- | --- |
| User Story 32 | |
| As a restaurant owner, I want to manage the delivery radius so that we can ensure timely deliveries. | |
| BV: 500 | CP: 03 |
| Tasks: 3 | Priority: Moderate |
| Acceptance Criteria:  App should provide an option for restaurant owners to set a delivery radius for their restaurant.  Restaurant owners should be able to input the radius in miles or kilometers.  App should calculate the delivery fee based on the distance from the restaurant to the user’s location.  Restaurant owners should be able to update their delivery radius at any time | |

|  |  |
| --- | --- |
| User Story 33 | |
| As a delivery person, I want to see the optimal route for delivery so that I can deliver the food as quickly as possible. | |
| BV: 1000 | CP: 03 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  App should provide real-time navigation from the restaurant to the customer’s location for each order.  Navigation should take into account current traffic conditions and road closures to determine the optimal route.  App should provide turn-by-turn directions to guide the delivery personnel along the route.  App should update the route in real-time if conditions change (e.g., new traffic jam) or if a new order is added | |

|  |  |
| --- | --- |
| User Story 34 | |
| As a delivery person, I want to mark myself as unavailable for deliveries when needed so that the system doesn’t assign orders to me during this time. | |
| BV: 50 | CP: 01 |
| Tasks: 2 | Priority: High |
| Acceptance Criteria:  App should provide an option for delivery personnel to mark themselves as available or unavailable.  Status update should be real-time, and once marked as unavailable, the delivery personnel should not receive any new order notifications.  Delivery personnel should be able to switch back to ‘available’ status when they are ready to take orders again | |

|  |  |
| --- | --- |
| User Story 35 | |
| As a delivery person, I want to see my earnings and tips so that I know how much money I made. | |
| BV: 100 | CP: 02 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  App should provide a dedicated section for delivery personnel to view their earnings and tips.  Earnings should be broken down by order, showing the amount earned from each delivery.  Any tips received from customers should be clearly indicated and added to the total earnings | |

|  |  |
| --- | --- |
| User Story 36 | |
| As an admin, I want to manage users and their roles (e.g., user, restaurant owner, delivery person) so that everyone has appropriate access rights in the app. | |
| BV: 1000 | CP: 05 |
| Tasks: 3 | Priority: High |
| Acceptance Criteria:  App should provide an admin panel with secure login.  Admin panel should display a list of all users, including customers, restaurant owners, and delivery personnel.  Admin should be able to assign or change user roles (e.g., customer, restaurant owner, delivery personnel)  Admin should be able to deactivate or reactivate user accounts.  Admin should be able to reset user passwords | |

|  |  |
| --- | --- |
| User Story 37 | |
| As an admin, I want to handle disputes and complaints between users and restaurants or delivery personnel so that we maintain high service quality. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: Moderate |
| Acceptance Criteria:  App should provide a secure admin panel where complaints and disputes can be viewed and managed.  Admins should receive notifications of new complaints or disputes.  Admins should be able to view the details of each complaint, including the parties involved, order details, and the nature of the complaint.  App should provide options for admins to resolve the complaint, such as issuing a refund, offering a discount on a future order, or taking disciplinary action against restaurant or delivery personnel.  Admins should be able to communicate with the parties involved to gather more information or notify them of the resolution.  App should keep a record of all complaints and their resolutions for future reference | |

|  |  |
| --- | --- |
| User Story 38 | |
| As an admin, I want to see analytics about the usage of the app (e.g., number of orders, most popular restaurants) so that we can make data-driven decisions. | |
| BV: 1000 | CP: 05 |
| Tasks: 2 | Priority: Low |
| Acceptance Criteria:  App should provide a secure admin panel where usage analytic can be viewed.  Dashboard should display key metrics such as total number of users, active users, most ordered dishes, most rated restaurants, average order value, etc.  App should update the analytic in real-time as new data comes in  Admin should be able to export the analytics data for further analysis | |

|  |  |
| --- | --- |
| User Story 39 | |
| As an admin, I want to send notifications about updates or maintenance to all users so that they are informed in advance. | |
| BV: 500 | CP: 02 |
| Tasks: 3 | Priority: Moderate |
| Acceptance Criteria:  App should provide a secure admin panel where notifications can be created and sent.  Admin should be able to compose a message for the notification.  App should provide options for the admin to select the recipients of the notification (e.g., all users, only customers, only restaurant owners, only delivery personnel)  App should send the notification immediately upon confirmation by the admin | |

|  |  |
| --- | --- |
| User Story 40 | |
| As an admin, I want to backup and restore data so that we don’t lose any important information. | |
| BV: 1000 | CP: 05 |
| Tasks: 4 | Priority: High |
| Acceptance Criteria:  App should provide a secure admin panel where data backup and restore functions can be accessed.  Admin should be able to initiate a backup of all data on the app at any time.  Backup process should not disrupt the normal operation of the app.  Admin should be able to view a list of all backups, including their dates and sizes.  App should provide a confirmation prompt before restoring data to prevent accidental data loss.  App should notify the admin when the backup or restore process is completed | |