**Deliverable 2 – Requirements Specifications**

**DineSys: Comprehensive Restaurant Management System for Seamless Dining Experience**

# Group Name:

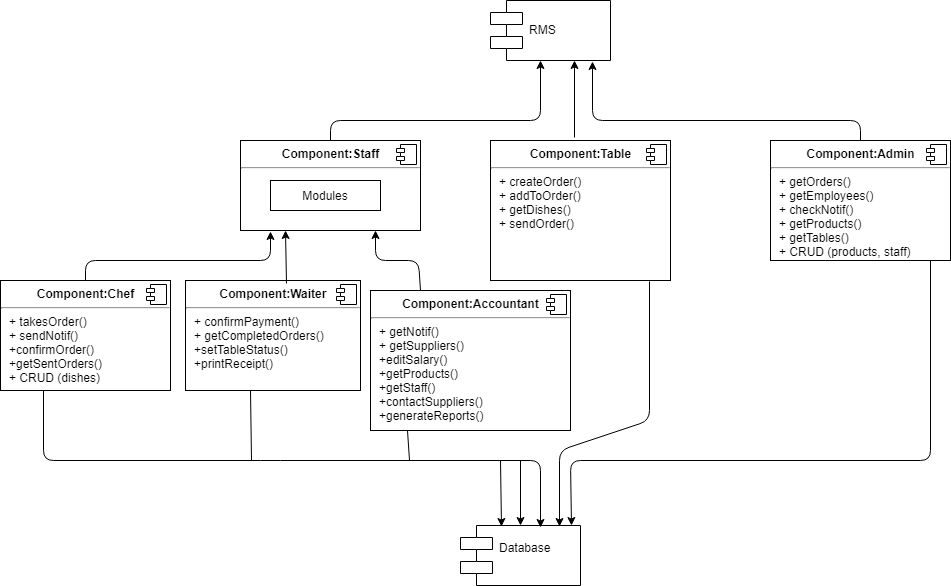
**TEAM SE's**

# Group Members:

* **Vangala Ruchitha (Team Leader)**
* **Nikhila Polkampally**
* **Nandhini Kasukurthi**
* **Vishal Rachuri**
* **Sai Srinivas Valleti**
* **Chandralekha Gude**
* **Aishwarya Karukonda**
* **Hunny Keshwani**
* **Leela Varadatta Sai Addanki**

# 1. Overall Structure of the System:

## Diagram:



DineSys stands as a testament to the modernization of the hospitality industry. In an era where efficiency and digital solutions reign supreme, DineSys aims to comprehensively manage every nuance of the restaurant environment, from client interactions to backend operations. This seamless integration aims to revolutionize the dining experience while optimizing the business operations of the restaurant.

## 1. Front-end Interface

The front-end interface is essentially the face of DineSys, the first point of contact between the restaurant and its customers. Designed with a user-centric approach, this interface manifests on tablets available at each dining table. The design philosophy behind this interface is to minimize the cognitive load for the user. Thus, it features an intuitive design where menus are categorized, high-resolution images of dishes are displayed, and there's a real-time order tracker.

But beyond the customer, the front-end also serves the restaurant staff. A different view is accessible to them, providing functionalities like order management, table reservations, and customer feedback consolidation. This dual nature ensures that the interface caters to the primary stakeholders - the customers and the restaurant staff.

## 2. Backend Server

Functioning as the system's brain, the backend server interprets, processes, and manages the plethora of data streaming in from various sources. Built using robust technologies such as PHP and bolstered by frameworks that ensure scalability and security, this server is designed to handle peak restaurant traffic without any hitches.

The server's main functions encompass:

* Processing orders from the front-end and dispatching them to the kitchen in real-time.
* Managing inventory data, keeping track of raw materials, and signaling when stocks are running low.
* Integrating HR functionalities, such as staff scheduling, payroll management, and performance reviews.
* Collating feedback and reviews from customers, offering insights into areas of improvement.

## 3. Database

The heart that complements the brain - the database is where all data lives, breathes, and is recalled from. Implemented using MySQL, a renowned relational database management system, it stores diverse data sets ranging from the intricate details of each dish on the menu to the employment history of the staff.

The database design is a complex weave of tables, relationships, and indices. Its normalized structure ensures data integrity while its indexing capabilities make data retrieval swift. Regular backups, encryption at rest, and fine-grained access controls ensure that data remains secure and protected against potential breaches.

## 4. Distributor Interface

The efficiency of a restaurant doesn't just hinge on its internal operations but also on its relationship with external entities, chiefly its distributors. DineSys features a dedicated distributor interface, designed as an API layer. This API framework allows for real-time integration with distributor systems.

When the inventory management module senses a low stock of certain items, it can automatically trigger purchase orders to the respective distributors via this interface. Additionally, the system can pull in information about new products, discounts, and shipment tracking details from distributor systems. This automated procurement process ensures that the restaurant rarely, if ever, runs into supply chain issues.

## 5. System Interactions and Connectivity

In envisioning the intricate dance of data between these components, imagine the following scenario: A diner, Jane, uses the front-end interface on her table's tablet to place an order. As she confirms her choices, this data flows seamlessly to the backend server, which interprets her order, checks against the database for dish details, and sends the order to the kitchen display system.

Meanwhile, as dishes get prepared and raw materials are used, the inventory system updates the database in real-time. If, for instance, a particular sauce runs low, the system could auto-generate an order to a distributor, using the distributor interface.

In essence, the overall structure of DineSys isn't just about individual components, but how they synergize to create a cohesive ecosystem. Each piece, from the tangible tablet interfaces to the abstract algorithms in the backend server, contributes to a holistic solution designed to redefine the contemporary dining experience. DineSys stands at the intersection of hospitality and technology, ensuring that neither domain compromises the other. Through meticulous design and comprehensive functionality, DineSys aims to set the gold standard for restaurant management systems in the digital age.

# 2. Written Requirements Specifications:

## a. Functional Requirements:

Functional requirements define the specific functions, features, or interactions that a system should possess. Here's a detailed breakdown of DineSys's functional requirements:

### Order Management:

* Customers should be able to view, select, and order dishes via the tablet interface.
* Staff should be notified instantly about new orders, changes to existing orders, or cancellations.
* Orders should be categorized based on their status - new, in-progress, ready, served, and closed.

### Inventory Management:

* The system should monitor inventory levels in real-time.
* Automatic alerts should be generated when stock falls below a set threshold.
* It should allow the creation, updating, and deletion of inventory items.

### Human Resources Management:

* Admins should be able to add, modify, or remove staff profiles.
* Facilitate the creation and management of staff schedules.
* Track performance and attendance of staff.

### Feedback System:

* Customers should be able to provide feedback on their dining experience, both qualitatively (comments) and quantitatively (ratings).
* Staff should be able to review and respond to feedback.

### Distributor Integration:

* Allow the registration and management of multiple distributors.
* Automatic generation and sending of purchase orders to distributors.
* Track delivery status and update inventory upon delivery receipt.

### Table Reservation:

* Customers should be able to reserve tables for a specific date and time.
* Reservations should reflect in real-time to avoid double bookings.
* Staff should have the capability to confirm or decline reservations.

## b. Non-Functional Requirements:

Non-functional requirements define how a system performs its functions, rather than what functions it performs.

### Performance:

* The system should be able to handle simultaneous access from all tables and staff without lag.
* Response times for user requests should be under 2 seconds.

### Scalability:

* The system should be scalable to accommodate more features or increased usage in the future.

### Security:

* All data transmissions, especially payment details, should be encrypted.
* User authentication mechanisms should be robust, ensuring only authorized access.

### Usability:

* The system's user interface should be intuitive, requiring minimal training for both staff and customers.

### Reliability:

* The system should boast a 99.9% uptime, ensuring constant availability.
* Regular backups should be taken to avoid data loss.

### Maintainability:

* The system should be designed in a modular fashion, allowing for easy updates, patches, or fixes.

## c. Interfaces:

DineSys, our restaurant management system, is not just about underlying logic and operations but also about seamless and efficient interfaces that connect multiple users and stakeholders. Here's a detailed exploration of the interfaces that will be integrated into DineSys:

### Customer Tablet Interface:

* Visual Design: A high-resolution and responsive touch interface that dynamically adjusts to tablet sizes. Vibrant images of dishes, clear pricing, and easy navigation are essential.
* Order Placement Module: Simple tap and swipe gestures will let customers select dishes, specify portions, and place orders. They should be able to customize orders by adding/removing ingredients or mentioning specific allergies.
* Feedback Module: An interactive interface where customers can rate their experience, leave comments, and even respond to surveys.
* Billing and Payment Module: The interface will provide itemized bills and facilitate digital payments via linked card, mobile wallets, or QR code scanning.
* Table Reservation System: Customers can pick their desired table by looking at an interactive layout of the restaurant. Date and time pickers should be intuitive.

### Staff Mobile Interface:

* Visual Design: A professional-looking interface with the restaurant’s branding. It should be lightweight and efficient to ensure quick load times on various devices.
* Order Management Dashboard: Real-time updates on orders placed, allowing waitstaff to mark orders as served, or chefs to mark dishes as prepared.
* Reservation Management: Staff can view upcoming reservations, see special requests, or notify customers of any changes.
* Feedback Viewer: To understand customer feedback instantly and make quick service adjustments if required.

### Admin Web Dashboard:

* Visual Design: A clean, clutter-free, and comprehensive dashboard with easy navigation tabs, charts, and data tables.
* Inventory Management Module: A visual representation of current stock levels, reorder indicators, and supplier details.
* Human Resources Management: A database of all staff members, their schedules, attendance, and performance reviews.
* Report Generation: Generate daily, weekly, or monthly sales, stock, and performance reports with graphical charts.
* Settings and Configurations: Modify system settings, add new dishes, adjust pricing, or run promotional offers.

### Distributor Web Interface:

* Visual Design: A business-oriented interface with clear lists, order details, and delivery schedules.
* Order Reception and Confirmation: Distributors can view orders placed by the restaurant, confirm them, or suggest alternatives for unavailable items.
* Delivery Scheduling: Set and communicate delivery dates and times.
* Billing and Payments: Generate invoices and receive digital payments.

### APIs for Integration:

* Payment Gateway API: To facilitate various payment methods and ensure secure transactions.
* Third-party Delivery Platforms: If DineSys is linked with platforms like UberEats or DoorDash, APIs will ensure seamless order integration.
* CRM Systems: If the restaurant uses external CRM tools, this API will allow data synchronization.

## 3. Implementation Phases:

### Phase 1 (Deliverable 3):

Prioritizing the direct dining experience and fundamental restaurant operations:

#### Order Management System:

* Establishing a fully functional order placement, tracking, and management system.

#### Table Reservation System:

* Allowing customers to book their tables in advance.

#### Basic Inventory Management:

* Setting up essential stock tracking, ensuring the restaurant doesn't run out of crucial ingredients.

#### Initial Feedback System:

* A rudimentary system for customers to leave feedback.

### Phase 2 (Deliverable 4):

Diving deeper into optimizations and enhanced functionalities:

#### Advanced Inventory Management:

* Introducing features like auto-generating purchase orders.

#### Human Resources Management:

* Comprehensive staff management features, from scheduling to performance tracking.

#### Distributor Integration:

* Allowing seamless coordination between the restaurant and its suppliers.

#### Enhanced Feedback System:

* Introducing response mechanisms and more detailed feedback options.

### Phase 3 (Deliverable 5):

Refining, securing, and scaling the system:

#### Security Enhancements:

* Introducing more robust encryption and user authentication mechanisms.

#### Performance Optimizations:

* Refining the system for faster response times and efficient data handling.

#### Usability Improvements:

* Refining the user interface based on user feedback and testing.

#### Scaling and Futureproofing:

* Making backend improvements to ensure easy scalability in the future.

DineSys's functional and non-functional requirements encompass the depth and breadth of the modern dining experience. By breaking down the implementation into three distinct phases, we ensure a methodical, prioritized, and organized development process. The approach allows the most critical features to be addressed first, progressively refining and expanding the system in subsequent phases. This structure promises a balanced combination of functionality, performance, and user experience, cementing DineSys's position as a benchmark in restaurant management systems.

# Team Members' Contribution for Deliverable 2:

|  |  |
| --- | --- |
| Team Member | Contribution |
| Vangala Ruchitha (Team Leader) | Oversaw overall project and ensured all tasks were on track. Led the meetings and worked on the overall structure of the system. |
| Nikhila Polkampally | Led the drafting of functional requirements. Collaborated with Vishal on UI requirements. |
| Nandhini Kasukurthi | Worked on non-functional requirements, with a focus on performance and reliability. |
| Vishal Rachuri | Assisted with functional requirements, focusing on the UI and hardware interfaces. |
| Sai srinivas Valleti | Developed the software interface specifications and worked on the communication requirements. |
| Chandralekha Gude | Took lead on planning the three-phase implementation, focusing on Phase 1. |
| Aishwarya Karukonda | Contributed to Phase 2 and 3 of the implementation plan. |
| Hunny Keshwani | Handled distributor management requirements and assisted in phase planning. |
| Leela Varadatta Sai Addanki | Managed the documentation, ensuring consistency and clarity. Also contributed to the non-functional requirements, focusing on security. |

# Segregation and Implementation Plan

Given the vastness of DineSys's scope, the implementation process needs meticulous planning. Breaking down the implementation into manageable chunks or segments will ensure efficiency and avoid overwhelming the development team. Here's a strategic plan:

## System Backbone and Core Features:

### Duration: 3 weeks

**Tasks:**

* Set up databases for customer details, order management, inventory, and HR.
* Develop primary logic for order placements, inventory updates, and basic reporting.
* Create a skeletal interface for customer tablets and staff mobile devices.
* Secondary Features and Enhanced Functionalities:

### Duration: 4 weeks

**Tasks:**

* Implement distributor interface and integrate with inventory management.
* Enhance the customer interface with feedback, reservations, and advanced customization features.
* Expand the admin dashboard to include detailed reporting, promotions, and performance analysis tools.

## Interface Polishing and Third-Party Integrations:

### Duration: 3 weeks

**Tasks:**

* Fine-tune the design elements of all interfaces, focusing on user experience and aesthetics.
* Implement APIs for payment gateways, third-party delivery, and CRM systems.
* Test interfaces across various devices and browsers for compatibility.

## Beta Testing and Feedback Collection:

### Duration: 2 weeks

**Tasks:**

* Launch a beta version of DineSys for select customers and gather feedback.
* Allow staff to use the system in a real-world scenario and collect their inputs.
* Identify bugs, glitches, or performance issues.

## Final Refinements and Launch:

### Duration: 2 weeks

**Tasks:**

* Implement feedback from the beta testing phase.
* Optimize database operations and ensure system security.
* Train the entire staff on using the system and kick-off DineSys officially.