REQUIREMENTS DEFINITITION DOCUMENT FOR DINE SWIFT APPLICATION

Document Version: 1.0 **Date:** September 4, 2025 **Project Lead:** Kiyimba Fahad

NAME	RegNo	EMAIL	PHONE
Mushabe Moses	23/U/12131/EVE	Mosesmushae9@gmail.com	0752307875
Drate Hillary	23/U/23611	dratehillary@gmail.om	0758235980
Mukyala Dorcus Nandy	23/U/11911/EVE	mukyaladorcus@gmail.com	0755011795
Kiyimba Fahad	23/U/0628	kiyimbafwitty@gmail.com	0762938957

Contents

1. INTRODUCTION

The DineSwift platform is a comprehensive software-as-a-service (SaaS) solution designed to digitize and optimize the entire dining experience for both customers and restaurants. It is a multi-faceted system that provides a seamless, contactless experience for customers while empowering restaurants with unprecedented operational efficiency, real-time data, and a new, reliable revenue stream. This document defines the core **functional** and **non-functional** requirements for the platform's initial release.

2. CORE BUSINESS OBJECTIVES

The primary objectives of the DineSwift platform are to:

- **Enhance Customer Experience:** Provide a seamless, friction-free ordering, payment, and communication experience that increases customer satisfaction and loyalty.
- **Boost Restaurant Efficiency:** Reduce operational bottlenecks, minimize wait times, and optimize staff workflows through intelligent automation.
- **Increase Revenue and Profitability:** Drive sales through dynamic menu presentation, gamified loyalty, and by reducing costs associated with human error and food waste.
- **Provide Actionable Insights:** Transform restaurant data into actionable intelligence, enabling data-driven decisions on staffing, menu engineering, and business strategy.

3. FUNCTIONAL REQUIREMENTS BY SERVICE

The following outlines the core functional requirements for each primary service.

3.1. Seamless, Contactless Ordering & Payments

- The system shall provide a **unique QR code** for each table to enable instantaneous, location-specific menu access.
- The platform shall support an **offline-first architecture** to ensure menu browsing and order placement are functional without a consistent internet connection.
- The system shall provide a **dynamic shopping cart** for customers to build their order.
- The platform shall integrate with a variety of **payment gateways** (e.g., Momo, Visa, and Bitcoin (in future) to facilitate secure, in-app payments.
- The system shall generate a **unique, one-time order code** upon confirmed payment for waiter verification.

3.1.1. Process flow

Phase 1: The Initial Contact (The Scan)

- A customer arrives at a table and, instead of waiting for a physical menu, uses their smartphone's camera to scan the unique **QR code** on the table.
- Each QR code is a secure, unique URL with embedded parameters. the code shall resolve to `dineswift.com/menu?restaurant=ffhh`. The DineSwift app (if installed) immediately intercepts this URL. If the app is not on the phone, the user shall be redirected to the web-based version of the menu.
- **Purpose:** This step is the "on-premise handshake." It instantly and automatically links the customer to the specific restaurant and their exact table, eliminating the need for a search function when a customer is already seated.

Phase 2: The Personalized Connection (The Load)

- **Action**: The DineSwift app or web interface loads the restaurant's customized menu. This is where our **offline-first architecture** shall come into play.
- **Mechanism**:The app first attempts to connect to the local server via the restaurant's Wi-Fi. If a connection is established, it retrieves a cached, up-to-date version of the menu data and media. The system prioritizes the local server to ensure near-instantaneous load times, even during an internet outage. If no local connection is available, the app falls back to fetching the data from the cloud.
- **Purpose:** This dual-path approach ensures uninterrupted service. The customer can browse, select, and order from a rich, immersive digital menu regardless of the restaurant’s internet stability.

Phase 3: The Interactive Experience (The Order)

- **Action:** The customer browses the menu, while viewing photos and videos for how specific dish preparation, and selects their desired items, adding them to a **dynamic shopping cart**.

- **Mechanism:** As items are added, the app provides real-time updates on the total cost. The customer can review their order before confirming. When they are ready, they press a **"Place Order"** button.
- **Purpose:** This phase transforms the customer from a passive recipient of service into an active participant. They can explore the menu at their leisure, take their time with their choices, and build their order without any communication friction.

Phase 4: The Payment Handshake (The Checkout)

- **Action:** After reviewing their order, the customer proceeds to the payment screen.
- **Mechanism:** The app presents the customer with a list of payment options, which can include both local and international gateways like **Momo** and **Visa, Bitcoin (to be considered for block chain in future)**. It automatically applies any pre-paid deposits or discounts to the final bill. The payment process is handled securely through the integrated gateway.
- **Purpose:** This phase provides **payment flexibility** and convenience, eliminating the need to wait for a physical bill or for the waiter to bring a card machine. The integration of local payment methods makes the system accessible in diverse markets.

Phase 5: The Final Confirmation (The Transmission)

- **Action:** The payment is successfully processed. A confirmation screen appears on the customer's device (It includes the **one-time order code** and, crucially, the **estimated time of arrival** for their food and the **name of the waiter** who will be delivering it).
- **Mechanism:** The app sends the final, confirmed order data to the local server. The server instantly processes the request and runs a calculation using our intelligent algorithm. It determines the **estimated food preparation time** by analyzing the complexity of the dishes ordered and the current load on the kitchen. It also identifies the most efficient waiter (based on availability and proximity) to deliver the order.

Purpose: The confirmation message returned to the customer's device is now much more informative. It includes the **one-time order code** and, crucially, the **estimated time of arrival** for their food and the **name of the waiter** who will be delivering it. This provides a clear, reliable expectation for the customer and acts as the trigger for the entire kitchen and waiter dispatch process.

3.1.2. Diagram

3.2. Intelligent Waiter Dispatch & Order Tracking

- The system shall automatically **assign a new order to the most efficient waiter** based on their status and location.
- The platform shall implement a **Dynamic Delivery Batch** algorithm that intelligently groups multiple ready orders to minimize a waiter’s trips.
- The waiter's app shall provide an **optimized delivery route** for each batch.
- The system shall require the waiter to **verify an order using the one-time order code** upon delivery.

3.2.1. Process Flow

Phase 1: Preparation Confirmation

- **Action:** When a chef finishes preparing an item or a full order, they press a **"Ready"** button on their kitchen screen.
- **Mechanism:** This action instantly updates the status of that specific item in the system. The customer’s app also receives a real-time notification that their food is ready for delivery.
- **Purpose:** This step creates the initial data point that the system needs to begin the batching process. It ensures the waiter is dispatched only when the food is physically ready to be delivered, preventing cold food and customer frustration.

Phase 2: Dynamic Batching

- **Action:** Our intelligent algorithm constantly monitors the status of all "Ready" orders across the restaurant.
- **Mechanism:** It looks for opportunities to group multiple ready orders together onto a single tray roller, up to a maximum of 10 orders. This dynamic grouping is based on two key criteria:
  - **Table Proximity:** It strategically groups orders for tables that are located close to each other.
  - **Order Readiness:** The algorithm only batches orders that are ready or will be ready within a very short, specified window (e.g., 30-60 seconds), ensuring no food gets cold while waiting for another item.
- **Purpose:** This is the core of the system’s efficiency. It minimizes the number of trips a waiter has to make, freeing them up to serve more customers and handle other tasks.

Phase 3: Smart Tray roller Assignment

- **Action:** Once an optimal batch of orders is identified, the system assigns the entire batch to the next available waiter.
- **Mechanism:** The waiter's dedicated app instantly receives a push notification, not for a single order, but for a **"New Delivery Batch."** The notification includes all the table numbers and a summary of the items in the batch.
- **Purpose:** This provides the waiter with a clear, single task that is designed for maximum efficiency. It's their cue to head to the kitchen and prepare for delivery.

Phase 4: Optimized Delivery Route

- **Action:** The waiter opens their app to view the new delivery batch.
- **Mechanism:** The app's interface shows a pre-planned, optimized sequence for delivery (e.g., "First, deliver to Table 7, then Table 9, and finally Table 3"). This sequence is calculated to be the most efficient path through the restaurant.
- **Purpose:** This eliminates all guesswork. The waiter doesn't need to think about which table to go to first; they just follow the pre-planned route, ensuring the fastest possible delivery time.

Phase 5: The Delivery and One-Time Order Code Verification

- **Action:** The waiter arrives at the first table in their batch.
- **Mechanism:** The waiter asks the customer for their **one-time order code**. The code on the customer's phone must match the code on the waiter's app for that specific order. This is a crucial final checkpoint that confirms the correct food is being delivered to the correct customer.
- **Purpose:** This foolproof verification step eliminates mix-ups and builds customer confidence, ensuring a professional and reliable experience.

Phase 6: Automated Completion

- **Action:** The waiter completes the delivery of all orders in their batch.
- **Mechanism:** The waiter taps a single **"Batch Complete"** button on their app. This action instantly updates their status in the central system from "Busy" to **"Available."**
- **Purpose:** This step closes the loop, signaling to the system that the waiter is ready for the next task. There is no need for a separate physical button on the counter, as the entire process is managed seamlessly within their app.

3.2.2. Diagram

3.3. Table Booking & Deposit System

- The platform shall allow customers to view and book an **available table** minimum of three days prior in advance.
- The system shall require a **refundable deposit** to secure a booking.
- The platform shall provide the customer with a **digital ticket** (booking ID or QR code) for check-in.
- The system shall **automatically apply the pre-paid deposit as a credit** to the customer’s final bill upon validation.

3.3.1. Process Flow

Phase 1: The Search and Availability Check

- **Action:** A customer opens the DineSwift app and navigates to the booking module. They select the restaurant, their desired date (three days prior in advance), the number of guests, and a preferred time slot.
- **Mechanism:** The app shall send a query to the server, which accesses a live-updating, visual map of the restaurant's floor plan. The server instantly returns a clear list and visual representation of all available tables, allowing the customer to see and select a specific spot.
- **Purpose:** This eliminates the frustration of guessing which tables are open and provides complete transparency, empowering the customer to choose the perfect spot.

Phase 2: The Reservation and Deposit Payment

- **Action:** Once the customer selects a table, they are prompted to pay a small, **refundable deposit** to finalize the booking.
- **Mechanism:** The chosen table is temporarily placed on hold for a brief period (e.g., 5 minutes) to prevent a booking conflict. The app then securely processes the deposit through our integrated payment gateways, such as **Momo** or **Visa or Bitcoin in future**.
- **Purpose:** This is a crucial step that acts as a safeguard against "no-shows." The deposit ensures the customer is committed to the reservation, protecting the restaurant from potential revenue loss.

Phase 3: The Confirmation and Digital Ticket

- **Action:** The payment is successfully processed. The customer receives a digital confirmation.
- **Mechanism:** The app displays a confirmation screen with all the reservation details, including the restaurant's name, the date, time, and a unique **booking ID or QR code**. This information is also stored in the customer's personal dashboard and sent via a push notification and/or email.
- **Purpose:** This confirmation serves as the customer's digital ticket, providing them with a simple, accessible record of their booking that they can present upon arrival.

Phase 4: The On-Premise Validation

- **Action:** The customer arrives at the restaurant and presents their digital ticket to the host.
- **Mechanism:** The host or a waiter uses their staff app to scan the customer's QR code or manually enter the booking ID. The system instantly validates the booking and automatically applies the pre-paid deposit as a credit to the customer's final bill.
- **Purpose:** This seamless process validates the reservation and smoothly deducts the deposit from the total, making for a swift and professional check-in experience.

3.3.2. Diagram

3.4. Flexible Food Ordering & Delivery

- The platform shall enable **off-premise ordering** for a customer to order food for delivery from any location.
- The system shall use **GPS location** to recommend nearby restaurants.
- The platform shall integrate with **third-party delivery services** (e.g., Uber Eats) to manage the final-mile logistics.
- The system shall provide customers with **real-time order tracking** from kitchen preparation to delivery.

3.4.1. Process Flow

Phase 1: The Remote Search and Discovery

- **Action:** A customer who is not at a restaurant opens the DineSwift app. Instead of scanning a QR code, they use the search function to find restaurants in a specific location or by cuisine type.
- **Mechanism:** The app sends a query to the cloud server, which uses the customer's GPS location to find a list of all nearby registered restaurants. It displays each restaurant with its rating, estimated delivery time, and any ongoing promotions, average food cost.
- **Purpose:** This phase transforms the app into a powerful discovery tool, allowing customers to easily browse and select from a wide range of dining options, even when they are not in the dining area.

Phase 2: The Menu & Order Configuration

- **Action:** The customer selects a restaurant and views its digital menu.
- **Mechanism:** Since the customer is not on the restaurant's local network, the app fetches the menu data, including photos and videos, directly from the **cloud server**. The customer then builds their order in a dynamic cart, just as they would when dining in.
- **Purpose:** This step ensures the customer has access to the full, up-to-date menu and can take their time to build their order, replicating the in-person experience in a remote setting.

Phase 3: Delivery Details and Payment

- **Action:** After building their order, the customer proceeds to checkout. They are prompted to provide a delivery address and choose a delivery partner.
- **Mechanism:** The app presents the customer with available third-party delivery options (e.g., Glovo) integrated with the DineSwift platform. Once the delivery method is selected, the customer proceeds to the secure payment page. They can use any integrated payment gateway, such as **Visa** or **Momo, Bitcoin(for future integration)**, to pay for both the food and the delivery fee in a single transaction.
- **Purpose:** This phase is critical for logistics. It links the order with a physical destination and a designated delivery service, while streamlining the payment process for both the food and the delivery.

Phase 4: Order Confirmation and Delivery Tracking

- **Action:** The payment is successfully processed. A confirmation screen appears on the customer’s device.
- **Mechanism:** The cloud server confirms the payment and sends the order details to the specific restaurant's local server. A unique, **one-time order code** is generated for the restaurant to use for easy order identification. The app displays an estimated food preparation and delivery time and a live tracker that shows the order's journey from the kitchen to their door.
- **Purpose:** This step provides complete transparency and a clear expectation for the customer. The tracker builds trust and gives the customer a sense of control and visibility throughout the entire process.

Phase 5: The Handover to Delivery Partner

- **Action:** Once the food is prepared and ready for pickup, a notification is sent to the third-party delivery partner.
- **Mechanism:** Our system has a direct **API integration** with the delivery service. When a chef marks the food as "Ready," our system automatically sends a request to the third-party delivery partner to dispatch a driver to the restaurant for pickup. The one-time order code is used to ensure the driver picks up the correct order.
- **Purpose:** This is the critical transition point where our system seamlessly hands off the order to the final leg of its journey, ensuring a swift and accurate delivery.

3.4.2. Diagram

3.5. Supplier Ordering & Inventory Management

- The system shall trigger a **low-stock alert** to the manager when an item falls below a pre-set threshold.
- The platform shall provide a **digital order form** that is pre-populated with low-stock items.
- The system shall integrate with suppliers via **API** or **secure messaging** to automate order placement.
- The platform shall provide a mechanism for the manager to **reject items**, specifying the **quantity and reason**, with an optional photo or digital signature for proof.
- The system shall **reconcile the final payment** based on the actual quantity of goods received and accepted.

3.5.1. Process Flow

Phase 1: The Inventory Alert

- **Action:** The system identifies that the stock of a specific ingredient or item has fallen below a pre-set threshold.
- **Mechanism:** This can be triggered in two ways:
  1. **Manual Input:** A manager manually updates the inventory, and the system automatically flags items that are running low.
  2. **Automated Tracking:** The DineSwift system, by tracking every dish prepared, can automatically deduct ingredients from the live inventory count. When an ingredient's count drops below the minimum threshold, a low-stock alert is triggered.
- **Purpose:** This phase is all about real-time, data-driven intelligence. It moves the restaurant from reactive ordering to proactive inventory management, preventing items from running out and causing missed sales.

Phase 2: The Digital Order Form

- **Action:** The manager receives a notification from the system that a supplier order is required.
- **Mechanism:** The manager's app presents a pre-populated, comprehensive order form. The form lists all the items that are low in stock, their current quantity, and a suggested re-order amount based on historical sales data. The manager can easily adjust quantities, remove items, or add new ones to the list.
- **Purpose:** This eliminates the need for the manager to manually check shelves or cross-reference paper lists. It streamlines the ordering process by providing a clear, intelligent, and customizable digital list.

Phase 3: The Supplier Connection

- **Action:** The manager reviews the order and submits it with a single tap.
- **Mechanism:** Our system has a direct **API integration** with a range of food suppliers. When the manager hits "submit," the app sends a secure, encrypted order request directly to the supplier's system. The order includes all item codes, quantities, and delivery details.

The system also supports sending the order via a standardized email or secure chat message for suppliers who do not have an API.

- **Purpose:** This phase automates and secures the communication between the restaurant and the supplier. It ensures the order is received accurately and instantly, reducing the risk of human error and eliminating the need for phone calls or faxes.

Phase 4: The Confirmation and Tracking

- **Action:** The order is sent, and the manager wants to confirm it has been received and is on its way.
- **Mechanism:** The DineSwift system receives an automated confirmation message from the supplier via the API or a confirmation email. The manager’s app immediately updates the status of the order to **"Confirmed"** and provides a live tracking link or an estimated delivery time.
- **Purpose:** This gives the manager complete visibility and peace of mind. They can track the order's journey in real-time and plan their day accordingly, knowing exactly when to expect the delivery.

Phase 5: The Delivery, Reconciliation, and Rejection

- **Action:** The supplier's delivery arrives at the restaurant.
- **Mechanism:** The manager uses their app to perform a digital check-in. They can quickly scan a barcode on the supplier's invoice or manually check off items from the digital order list.
    - **For Rejected Goods:** If an item is spoiled, damaged, or off-order, the manager taps a **"Reject Item"** button. The app prompts them to input the exact **quantity to be rejected** and select a reason. To finalize the rejection, the manager must either capture a digital signature from the delivery personnel on the app or take a photo of the rejected item and attach it with a brief description.
- **Purpose:** This step ensures that the inventory numbers are perfectly accurate and provides a clear record of why an item was not accepted. The digital trail of rejected items provides a clear record of why an item was not accepted, holding the supplier accountable and protecting the restaurant's profits.

Phase 6: The Payment and Final Reconciliation

- **Action:** The manager is ready to pay the supplier.
- **Mechanism:** The DineSwift app automatically calculates the final, adjusted total. It takes the original order value and subtracts the cost of the rejected goods and their corresponding quantities. The manager can then securely process the payment through the app using a pre-configured payment method, such as a business bank account or credit card, or Bitcoin.
- **Purpose:** This final phase automates the financial reconciliation process. It ensures the restaurant only pays for what it actually received in acceptable condition, completing the entire transaction cycle with a clear, auditable trail.

3.5.2. Diagram

3.6. Communication & Feedback Hub

- The platform shall provide a secure, **in-app chat function** for customers to communicate directly with waiters, chefs, and managers.

- The system shall provide a **post-dining feedback prompt** to collect a 1-5 star rating and optional written feedback.
- The platform shall allow customers to **rate specific waiters** for performance.

3.6.1. Process Flow

Part 1: The Communication Hub

This part of the service is about instant, direct messaging between different parties within the restaurant.

Phase 1: Initiating a Chat

- **Action:** A customer needs to ask a question, report an issue, or simply wants to give a compliment. They open the DineSwift app and tap the **"Chat"** button on the restaurant's main page. They can choose to start a chat with a specific waiter, the chef, or a manager (this could be about Dinning "like more extra napkins", Health issue, Danger, Private, Other. The first three have higher priority and may signal an alarming notification and require the table number).
- **Mechanism:** When the customer selects a recipient (individual, open forum for all registered customers on a given restaurant), the app securely opens a dedicated chat channel. The chat is encrypted.
- **Purpose:** This eliminates the need for a customer to physically track down a waiter or manager, saving time and avoiding awkward interruptions. It provides a simple, private, and direct way to communicate.

Phase 2: Real-Time Messaging

- **Action:** The customer sends a message (e.g., "Could we get some extra napkins ").
- **Mechanism:** The message is instantly delivered to the staff member's app with a push notification. The waiter, chef, or manager can respond directly within the chat. The conversation history is stored, allowing them to reference it later.
- **Purpose:** This provides a clear, documented record of communication, reducing the chance of misunderstandings and ensuring that a customer's request is handled efficiently.

Part 2: The Feedback & Rating System

This is a critical service for both identifying high-performing restaurants and collecting valuable data.

Phase 1: The Post-Dining Prompt

- **Action:** After a customer completes their order and payment, a pop-up appears on their screen. It asks them to rate their experience.
- **Mechanism:** The prompt is simple and direct: "How was your experience at e.g Café Javas?" It's followed by a 5-star rating scale and an optional text box for more detailed feedback. Registered customers are also prompted to **rate their specific waiter**.
- **Purpose:** This prompt is designed to capture feedback while the experience is fresh in the customer's mind. It's a non-intrusive way to encourage ratings.

Phase 2: Data Aggregation and Analytics

- **Action:** The customer submits their rating and feedback.
- **Mechanism:** The data is securely sent to our cloud server. The system immediately aggregates this new rating and updates the restaurant's or the waiter's overall average rating. The detailed text feedback is anonymized and tagged with keywords (e.g., "fast service," "tasty food," "long wait").
- **Purpose:** This data is the foundation of our business intelligence. It allows us to identify **high-performing restaurants** and reward top waiters. This provides restaurant managers with specific, actionable insights into what they are doing well and where they can improve.

Phase 3: The Data Dashboard

- **Action:** A restaurant manager or owner wants to see their performance.
- **Mechanism:** They log into their DineSwift dashboard and access the analytics section. They can view a real-time graph of their ratings over time, see the average rating for each waiter on their team, and read the specific feedback left by customers.
- **Purpose:** This dashboard transforms raw data into a powerful tool for decision-making. Managers can use this information to praise and reward staff, adjust staffing levels during peak hours, and address recurring issues based on direct customer input.

3.6.2. Diagram

3.7. Gamified Loyalty Program

- The system shall automatically enroll customers into the program at the **Bronze Tier** upon their first registered order.
- The loyalty tier shall be a **permanent status** that can only increase based on two metrics: **total lifetime spend** or **total orders placed**.
- The platform shall provide a separate, **spendable points currency** that is earned with every purchase.

- The system shall provide **tier-specific rewards** (e.g., Gold Tier benefits) in addition to points-based rewards.

3.7.1. Process Flow

Part 1: The Loyalty Tier Process

This system is all about progression. A customer's loyalty level can only increase as they hit specific milestones. It is calculated using two key metrics: **Total Lifetime Spend** and **Total Orders Placed.**

Phase 1: Automatic Enrollment

- **Action:** A customer is automatically enrolled in the loyalty program the moment they complete their first order as a registered user on DineSwift.
- **Mechanism:** The system creates a profile for them and assigns them to the **Bronze Tier**. Their personal dashboard displays their status and the milestones for the next level.
- **Purpose:** This step makes entry into the program effortless. There's no separate sign-up required, which reduces friction and maximizes enrollment.

Phase 2: Tier Progression

- **Action:** The customer makes a purchase through the DineSwift app, and their engagement is tracked.
- **Mechanism:** The system constantly monitors the customer's cumulative **lifetime spend** and the total **number of orders placed.** A customer will be automatically promoted to the next tier when they hit a predetermined threshold for *either* metric. For example:
  - **Bronze to Silver:** Achieved after spending a cumulative **$500** *OR* placing **20 orders**.
  - **Silver to Gold:** Achieved after spending a cumulative **$1,500** *OR* placing **50 orders**.
- **Purpose:** This dual-metric approach ensures a balanced and fair system that rewards both high spenders and frequent visitors. The tier level is permanent and will not decrease.

Part 2: The Points & Redemption Lifecycle

This is a separate, spendable currency that customers earn with every purchase. It is linked to their loyalty tier, which can unlock more valuable rewards.

Phase 1: Earning Points

- **Action:** The customer completes a purchase through the DineSwift app.
- **Mechanism:** For every dollar/shs spent, the system automatically awards a specific number of points (e.g., 10 points (or in satoshis) for every $1 spent). These points are instantly added to the customer's profile after the payment is confirmed.
- **Purpose:** This provides a simple, transparent reward system that customers can use to redeem for benefits without affecting their hard-earned loyalty status.

Phase 2: Unlocking and Redeeming Rewards

- **Action:** The customer decides to use their points or a tier-based reward.
- **Mechanism:** The app presents the customer with a list of available rewards. Some rewards may be tied to a specific loyalty tier. For example, a **Gold Tier** customer may have access to a daily **moderate discount** or a **free side dish** with every order, in addition to rewards they can purchase with their accumulated points.
- **Purpose:** The redemption process is designed to be as simple as possible. It ensures that customers can easily benefit from their loyalty, reinforcing the value of the program and encouraging them to keep using the app.

Part 3: The Analytics Loop

- **Action:** Restaurant managers want to understand their most loyal customers.
- **Mechanism:** The DineSwift dashboard provides detailed analytics on customer loyalty. Managers can see a real-time list of their top customers, identify who is in the Gold Tier, track their spending habits, and even see which menu items they order most frequently.
- **Purpose:** This data is priceless for a restaurant. It allows them to identify and personally thank their most valuable customers and even create hyper-targeted promotions based on a customer's specific preferences, ensuring they feel recognized and appreciated.

3.7.2. Diagram

3.8. Immersive Media Integration

- The platform shall enable restaurants to **upload and link high-quality photos and videos** to specific menu items.
- The customer-facing menu shall include a **dynamic video carousel**.
- The system shall include a **"like" button** on media to collect social feedback from customers.

- The platform shall track **view count** and **conversion rate** for all media to measure its effectiveness.

3.8.1. Process Flow

Phase 1: Content Creation & Upload

- **Action:** A restaurant manager or professional creates short, high-quality videos and photos of key dishes.
- **Mechanism:** The manager uses the DineSwift backend dashboard to upload these media files. They can then link each video or photo to a specific menu item. The videos are optimized for mobile viewing, ensuring fast load times without sacrificing quality.
- **Purpose:** This phase gives the restaurant a powerful tool to showcase their best-selling or most visually appealing dishes. It's the equivalent of a great photo on a physical menu but with the added dynamic element of video.

Phase 2: The Customer Experience & Social Feedback

- **Action:** A customer is browsing the menu on their device.
- **Mechanism:** The app's interface features a dynamic home screen with a looping video carousel showcasing popular dishes. When the customer navigates to a specific dish's page, a full-screen, high-quality video or a series of detailed photos plays. Below the media, there is a prominent "like" button (e.g., a heart icon) and a live counter showing the total number of likes. A customer can tap this button to show their appreciation, and their "like" is instantly added to the total.
- **Purpose:** This visual and interactive element is a powerful sales tool. It's not just showing the customer what the dish looks like; it's creating an emotional and sensory connection that makes the food more appealing. The "like" button adds a social element, making the menu more engaging and allowing customers to directly influence which dishes get highlighted.

Phase 3: Performance Analytics & Social Proof

- **Action:** The restaurant manager wants to know how effective their media content is.
- **Mechanism:** The DineSwift analytics dashboard provides real-time data on the media. It tracks key metrics like **view count** for each video, **click-through rates** to the linked menu item, and, most importantly, the **total number of likes**. The system can identify the most-liked videos, providing the restaurant with powerful **social proof**.
- **Purpose:** This final phase closes the loop. It gives the restaurant a clear, data-driven way to measure the return on their content investment. They can use this information to identify which media is most effective and feature the most-liked dishes more prominently on the menu, knowing they are popular with their customers.

3.8.2. Diagram

3.9. Comprehensive Data Analytics Dashboard

- The dashboard shall provide **real-time analytics** on sales, revenue, and item performance.
- The system shall track and visualize **staff performance metrics** (e.g., average delivery time, customer ratings).
- The platform shall provide **customer insights** based on order history and loyalty tier.
- The dashboard shall provide **inventory and supplier analytics** to track order history and rejected goods.

Phase 1: The Data Collection Engine

This is the intake phase where every single interaction within the DineSwift ecosystem is captured. The system is constantly listening for new data points.

- **Customer Activity:** Every order placed, item viewed, video liked, and review submitted is logged.
- **Waiter Performance:** The system tracks the time from order assignment to delivery completion for every waiter, as well as their total tips earned.
- **Financial Transactions:** All sales, discounts applied, and supplier payments are automatically recorded, providing real-time financial tracking.
- **Inventory & Suppliers:** The system tracks every item ordered from a supplier, the quantity received, and any rejections, providing a live cost of goods sold.

Phase 2: Real-Time Processing & Visualization

Once the data is collected, our platform instantly processes and aggregates it, turning raw numbers into meaningful metrics that are easy to understand.

- **Real-Time Dashboards:** Managers get access to a series of customizable dashboards that provide a clear overview of the business. They can view a live chart of **sales by the hour** to understand their busiest times, see a breakdown of their **most profitable dishes**, and track their **daily revenue** against a set target.
- **Waiter Analytics:** The system provides a clear scoreboard for each waiter. Managers can see their **average delivery time**, their total orders delivered, and their customer satisfaction ratings.
- **Customer Insights:** The dashboard shows a clear list of the **most popular dishes** and the **most-liked videos** on the menu. Managers can also see a breakdown of their **most loyal customers** and their spending habits.

Phase 3: The Actionable Intelligence Loop

This is the most critical phase. The data isn't just for looking at; it's for making smarter business decisions.

- **Optimizing Staffing:** By looking at the **peak order times** and the waiter efficiency metrics, a manager can make data-driven decisions on when to schedule more staff to avoid bottlenecks.
- **Menu Engineering:** The sales data and customer feedback provide clear insights into what to keep, what to remove, and what new items to introduce. If a new dish has a high view count but a low sales conversion, the manager knows to either improve the dish or their marketing of it.
- **Rewarding Top Performers:** With a clear view of their top-rated and most efficient waiters, a manager can easily reward them with bonuses or recognition, which boosts staff morale and encourages friendly competition.

- **Predictive Stocking:** The system can analyze sales trends to predict future demand, allowing the manager to place more accurate supplier orders and reduce food waste.

4. NON-FUNCTIONAL REQUIREMENTS

These requirements define the system's quality attributes and constraints.

4.1. Performance

- **Response Time:** The customer-facing app shall load the menu in less than **3 seconds** on a standard 4G connection.
- **Order Processing:** Orders shall be sent to the kitchen within **500 milliseconds** of confirmed payment.
- **Data Updates:** The real-time analytics dashboard shall refresh with new data points every **10 seconds** or less.

4.2. Scalability

- The platform architecture shall be designed to support the simultaneous use of **100,000+ customers** and **1,000+ restaurants**.
- The database structure shall be capable of handling millions of orders without significant performance degradation.

4.3. Security

- All data transmitted between the app, servers, and payment gateways shall be secured using **end-to-end encryption (SSL/TLS)**.
- The platform shall be **PCI DSS compliant** for secure handling of credit card information.
- Customer and restaurant data shall be protected through **role-based access control**, ensuring staff only see information relevant to their role.

4.4. Reliability & Availability

- The core platform (servers, APIs) shall maintain an uptime of **99.9%** (less than 9 hours of downtime per year).
- The system's **offline-first architecture** shall ensure core functionality (menu browsing, order building) remains available even during a total internet outage.

4.5. Usability

- The customer-facing app shall have a clean, intuitive user interface that is easy to navigate for users of all ages.
- The staff-facing app shall be designed for **quick, one-tap actions** to minimize errors and maximize efficiency during high-pressure situations.

4.6. Compatibility

- The platform shall be fully responsive and optimized for use on both **iOS and Android** mobile devices.(In future the visually impaired shall be supported to order for food effectively)
- The web-based menu and dashboard shall be compatible with all major modern browsers, including Chrome, Safari, Firefox, and Edge.

4.7. Success Metrics

The success of the initial DineSwift platform will be measured by:

- **Customer Adoption Rate:** The number of unique users placing an order via the platform.
- **Restaurant Enrollment:** The number of restaurants successfully onboarded.
- **Average Order Value:** An increase in the average amount spent per order.
- **Wait Time Reduction:** A measurable decrease in the average time from order placement to delivery.
- **Customer Satisfaction:** An increase in average customer ratings and positive feedback.