

CSC 322 Phase II Write-up

Omar Quraishi
Tanvir Islam
Al Naheyan
Johir Hossain

UPDATED Use Cases & Sequence Diagrams/Petri-nets

Use Case 1: Browse Menu and Ask Questions

Actors: Visitor, Registered Customer, VIP Customer, AI System

Preconditions: User accesses the system; menu items exist in database.

Main Flow:

1. User views menu with dish descriptions, images, prices, and ratings.
2. System shows personalized recommendations for logged-in users (most ordered, highest rated).
3. VIP users see exclusive VIP-only dishes.
4. User can ask questions via chatbot; system searches knowledge base first, then LLM if needed.
5. System prompts user to rate answer (0-5 stars).

Postconditions: User browses menu; questions answered and rated.

Functional Requirements:

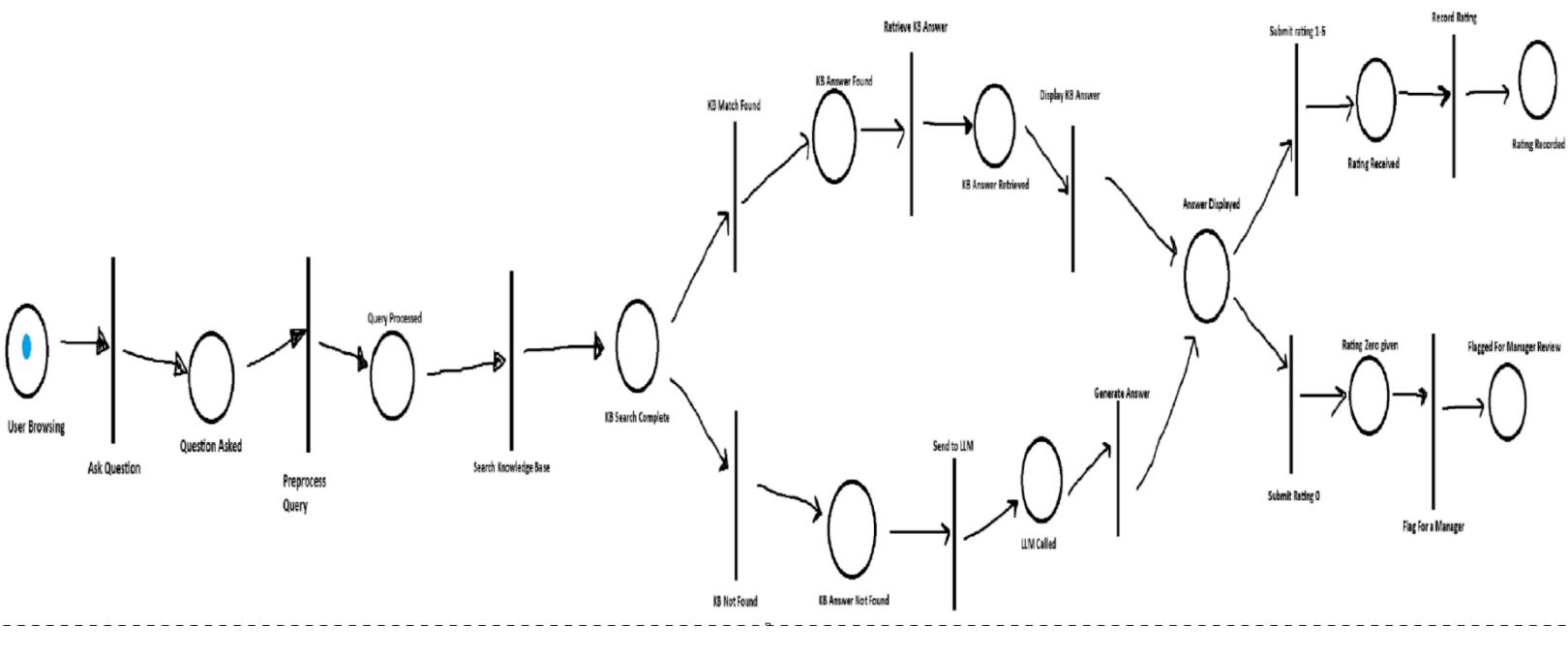
- Display menu with filtering by chef, price, rating
- AI chatbot integration with KB and LLM fallback
- Answer rating mechanism with flag for rating 0

Non-Functional Requirements:

- Page load time < 2 seconds
- LLM response time < 5 seconds
- KB search prioritized over LLM for cost efficiency

Exceptional Scenarios:

- E1: No KB match → fallback to LLM (Ollama/Hugging Face)
- E2: User rates answer 0 → flag for manager review, block author if inappropriate
- E3: LLM unavailable → display error, offer manager contact option



Use Case 2: Customer Registration and Login

Actors: Visitor, Manager

Preconditions: Visitor has valid contact information.

Main Flow:

1. Visitor submits registration form (name, email, phone, password).
2. Application stored as "Pending" status.
3. Manager reviews application and approves/rejects.
4. Approved customers can log in with credentials.
5. System displays personalized dashboard with VIP status, warnings, balance.

Postconditions: Customer account created and active; login credentials established.

Functional Requirements:

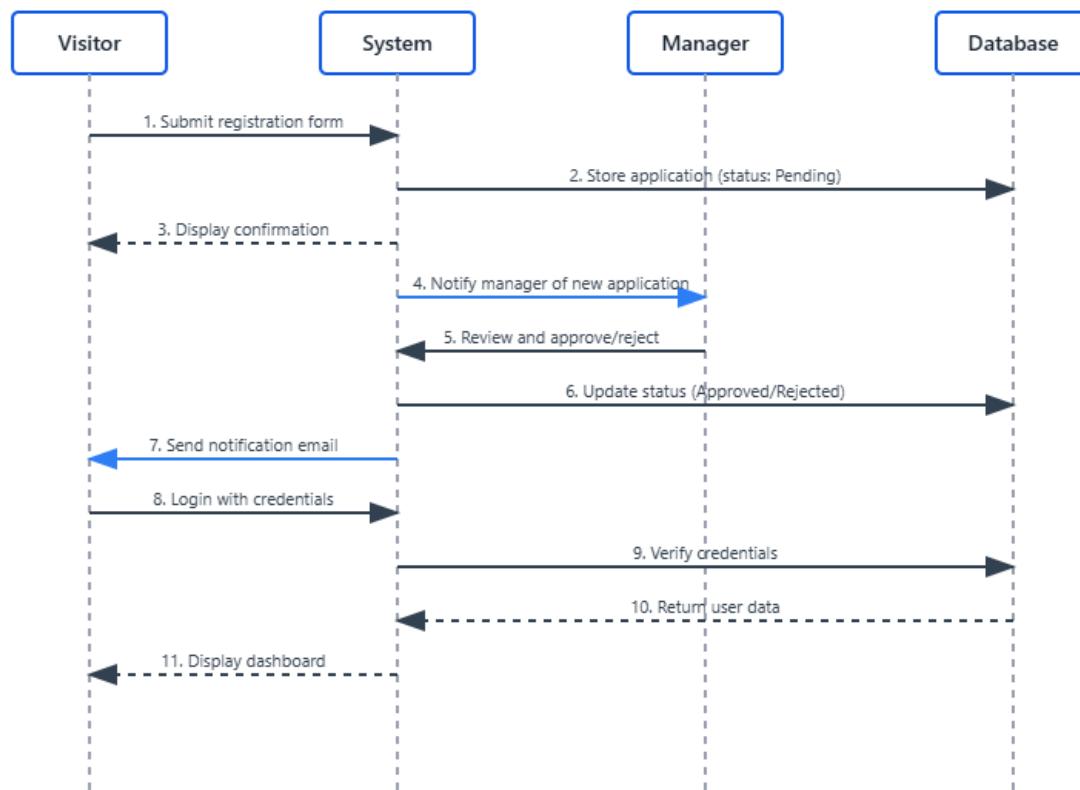
- Secure password hashing
- Manager approval workflow
- Session management with timeout

Non-Functional Requirements:

- Password must meet complexity requirements (8+ chars, mixed case, numbers)
- Session timeout after 30 minutes of inactivity
- Login authentication < 1 second

Exceptional Scenarios:

- E1: Duplicate email → reject registration with error message
- E2: Manager rejects → notify visitor with reason, cannot reapply for 30 days
- E3: Blacklisted visitor attempts registration → auto-reject silently



Use Case 3: Manage Deposit

Actors: Registered Customer, VIP Customer

Preconditions: Customer is logged in.

Main Flow:

1. Customer navigates to deposit management page.
2. System displays current balance.
3. Customer enters deposit amount (minimum \$10).
4. Customer confirms payment method (simulated for prototype).
5. System adds funds to customer balance.
6. System displays updated balance and transaction confirmation.

Postconditions: Customer deposit balance increased; transaction logged.

Functional Requirements:

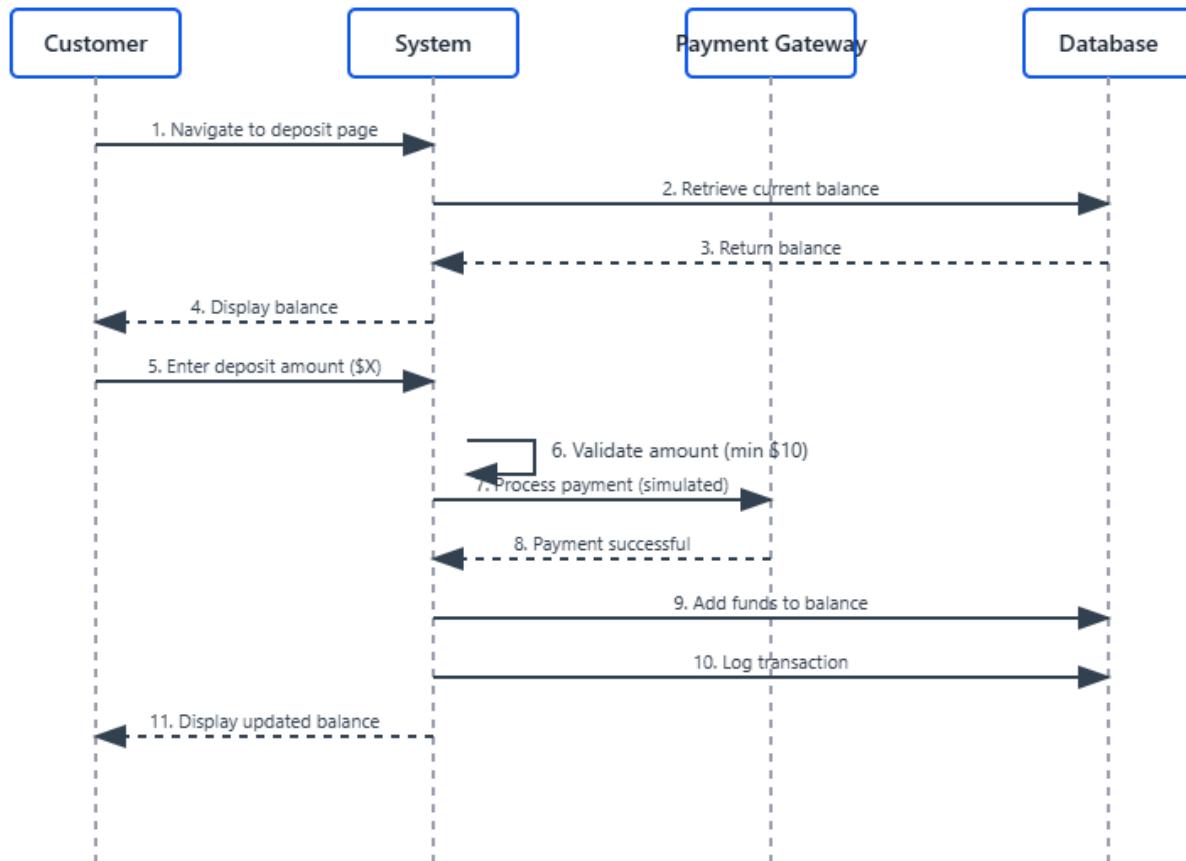
- Real-time balance updates
- Transaction history with timestamps
- Minimum deposit validation

Non-Functional Requirements:

- Payment processing simulation < 2 seconds
- Transaction logs immutable and auditable
- Balance displayed with 2 decimal precision

Exceptional Scenarios:

- E1: Deposit amount < \$10 → reject with error "Minimum deposit is \$10"
- E2: Payment processing fails → rollback transaction, notify customer



Use Case 4: Place Order

Actors: Registered Customer, VIP Customer, System

Preconditions: Customer logged in; items in cart; deposit balance exists.

Main Flow:

1. Customer reviews cart and confirms order.
2. System calculates total (including taxes, delivery fee).
3. System checks VIP status; applies 5% discount if VIP.
4. System checks free delivery eligibility (every 3rd order for VIPs).

5. System verifies deposit balance \geq order total.
6. System deducts amount from deposit.
7. System creates order record; notifies chef(s).
8. System initiates delivery bidding.
9. Customer receives order confirmation with order ID and ETA.

Postconditions: Order created; deposit reduced; chef notified; delivery bidding started.

Functional Requirements:

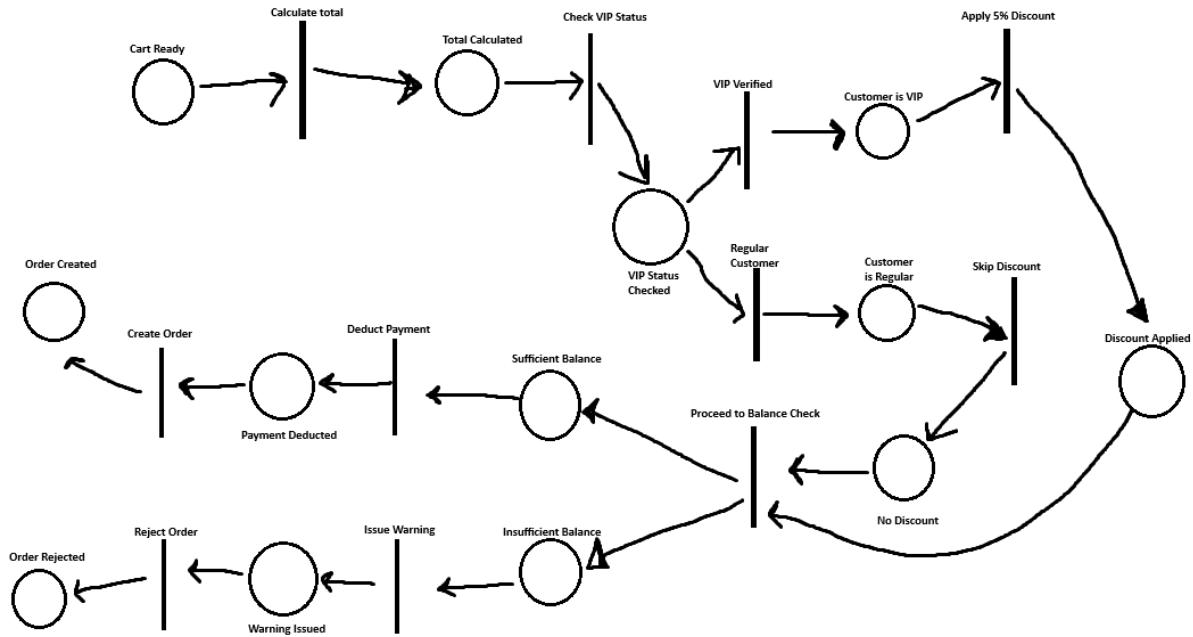
- VIP discount calculation (5% off)
- Free delivery tracking (1 per 3 orders for VIPs)
- Automatic balance verification
- Order confirmation with unique ID

Non-Functional Requirements:

- Order processing < 3 seconds
- Real-time inventory check
- Atomic transaction (all-or-nothing)

Exceptional Scenarios:

- E1: Insufficient balance → reject order, auto-add warning (UC27), redirect to deposit page
- E2: Customer has 3 warnings → block order, display "Account suspended"
- E3: Dish removed during checkout → notify customer, allow cart modification



Use Case 5: Delivery Bidding and Assignment

Actors: Delivery Person, Manager

Preconditions: Order ready for delivery; at least 2 delivery persons available.

Main Flow:

1. System notifies available delivery persons of new order.
2. Delivery persons submit bids (delivery price and estimated time).
3. System ranks bids by price (lowest first).
4. Manager reviews bids and selects winner (typically lowest).
5. If manager chooses higher bid, must write justification memo.
6. System assigns order to selected delivery person.
7. Assignment notification pushed to delivery person.

Postconditions: Delivery task assigned; audit record stored with justification if applicable.

Functional Requirements:

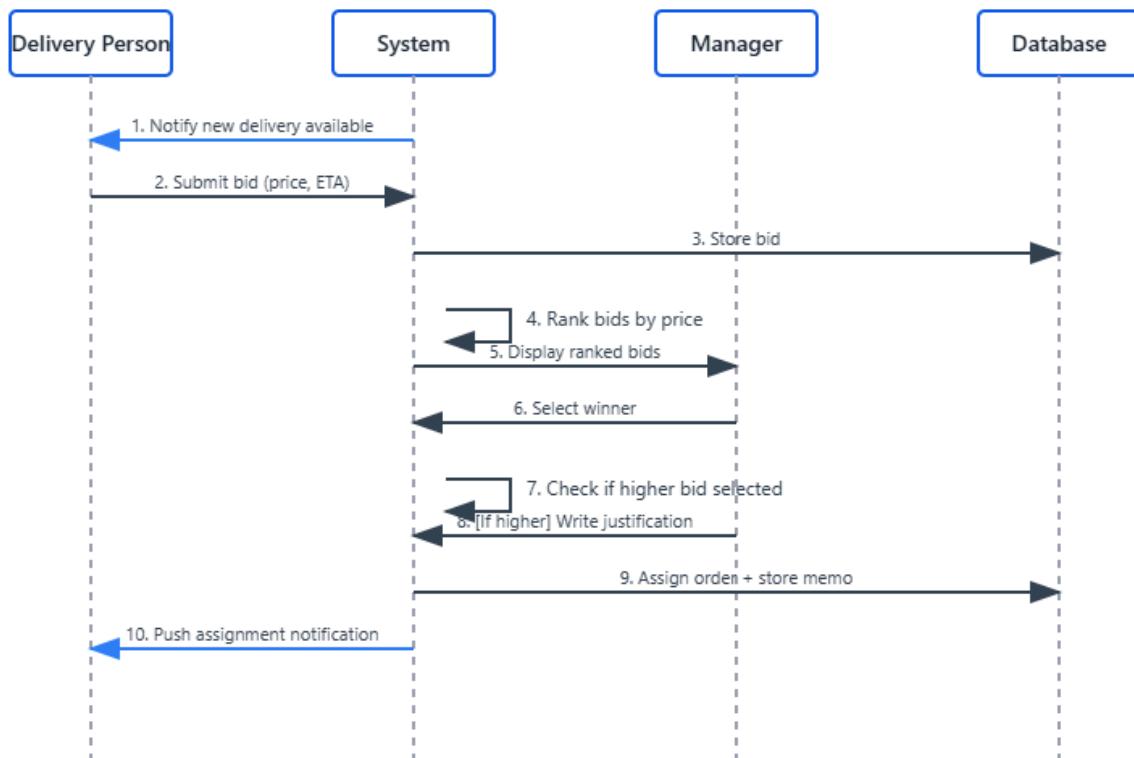
- Bid submission interface with price and ETA fields
- Manager override with mandatory memo
- Notification system for assignment

Non-Functional Requirements:

- Real-time bid updates < 2 seconds
- Audit logs durable and tamper-proof
- Bid window closes after 5 minutes or 3 bids received

Exceptional Scenarios:

- E1: No bids received → manager manually assigns to available delivery person
- E2: All bids exceed threshold → manager negotiates or increases delivery fee
- E3: Selected delivery person goes offline → manager reassigned to next lowest bidder



Use Case 6: Deliver Order and Update Status

Actors: Delivery Person, Customer, Manager

Preconditions: Order assigned to delivery person; order status "Ready for Delivery".

Main Flow:

1. Delivery person views order details (customer address, items, phone).
2. Delivery person updates status to "Out for Delivery".
3. System timestamps pickup; notifies customer.
4. Delivery person transports food to customer.
5. Delivery person arrives and hands off food.
6. Delivery person updates status to "Delivered".
7. System credits delivery fee to delivery person account.
8. System prompts customer to rate food and delivery service.

Postconditions: Order marked "Delivered"; delivery fee credited; rating prompts sent.

Functional Requirements:

- Real-time status updates
- GPS tracking (optional creative feature)
- Photo proof for contactless delivery

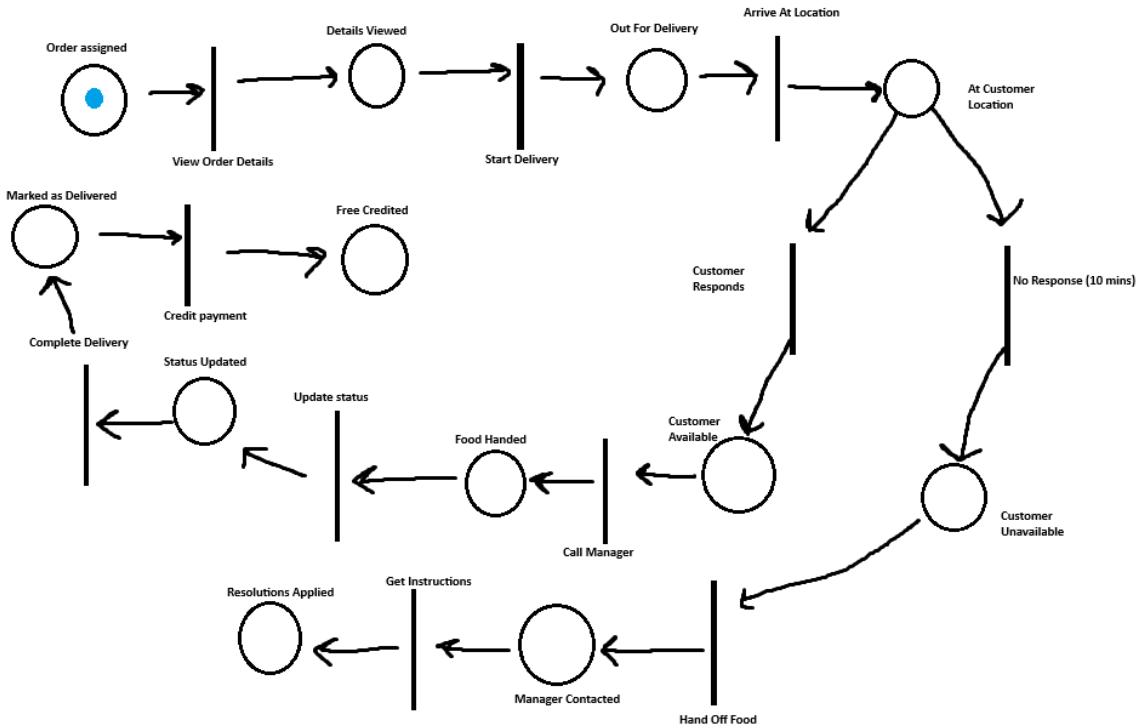
Non-Functional Requirements:

- Status update propagation < 3 seconds
- High availability for tracking channel (99.9% uptime)
- Delivery timestamp accurate to the second

Exceptional Scenarios:

- E1: Customer unavailable → delivery person waits 10 min, contacts manager for instructions (leave at door/return)
- E2: Wrong address → delivery person contacts customer; manager approves extra fee if far

- E3: Accident/vehicle issue → manager reassigns order; original delivery person gets 50% payment



Use Case 7: Rate Food and Delivery

Actors: Registered Customer, VIP Customer

Preconditions: Order delivered; ratings not yet submitted.

Main Flow:

1. System prompts customer to rate after delivery.
 2. Customer rates food quality (1-5 stars) and optionally comments.
 3. Customer rates delivery service (1-5 stars) and optionally comments.
 4. System records ratings with timestamp.
 5. System updates chef's and delivery person's average ratings.

6. System checks performance thresholds for promotions/demotions.
7. VIP ratings weighted 2x in calculations.

Postconditions: Ratings recorded; chef and delivery person statistics updated; thresholds checked.

Functional Requirements:

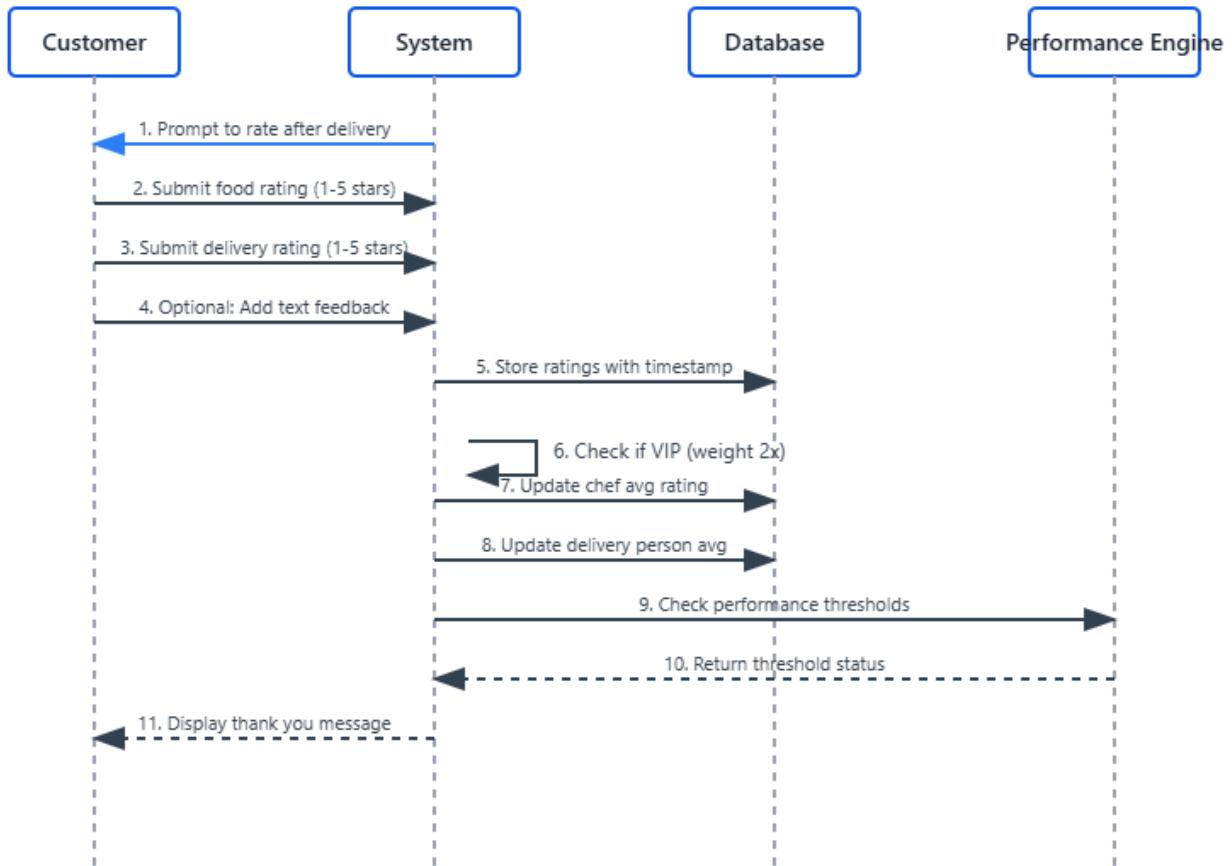
- Separate ratings for food and delivery
- Optional text feedback (max 500 chars)
- VIP rating weight multiplier (2x)
- Prevent duplicate ratings for same order

Non-Functional Requirements:

- Rating submission < 1 second
- Real-time average calculation
- Display aggregated ratings to public with precision to 0.1

Exceptional Scenarios:

- E1: Customer attempts duplicate rating → system displays "Already rated"
- E2: Extremely low pattern (all 1-star) → flag account for potential abuse
- E3: Rating triggers threshold → auto-initiate promotion/demotion process (UC28)



Use Case 8: File and Handle Complaints/Compliments

Actors: Registered Customer, VIP Customer, Delivery Person, Manager

Preconditions: Interaction between parties occurred (order, delivery, or forum discussion).

Main Flow:

1. User files complaint/compliment against chef, delivery person, or customer.
2. System stores complaint with status "Pending"; VIP complaints weighted 2x.
3. Manager reviews complaint details and investigates.
4. Manager makes decision: uphold (add warning/demotion) or dismiss (warn filer if false).
5. For compliments: manager awards bonus or cancels existing complaint.
6. System applies consequences and updates reputation scores.

7. System notifies all involved parties.
8. Complaint status updated to "Resolved".

Postconditions: Complaint resolved; consequences applied; parties notified; audit trail created.

Functional Requirements:

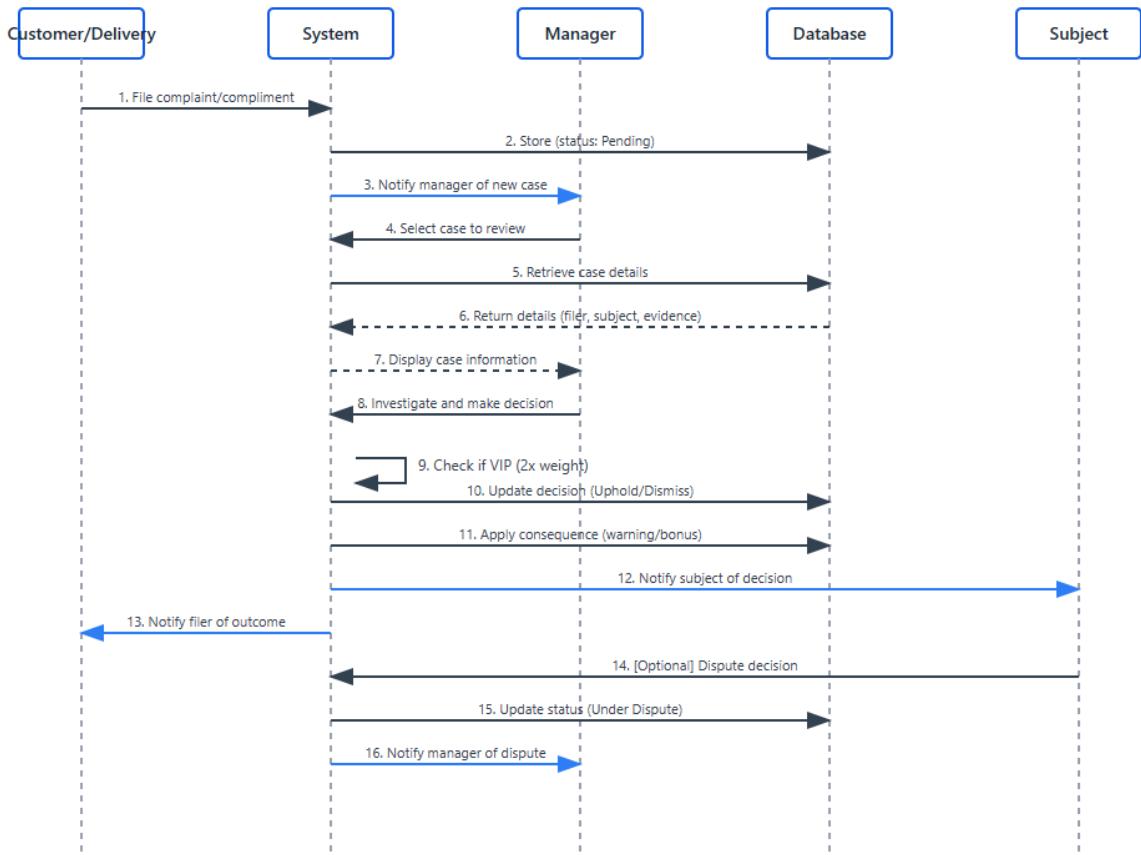
- Complaint categorization (chef/delivery/customer behavior)
- Manager investigation interface with evidence display
- VIP weight multiplier (2x impact)
- Dispute mechanism (subject can appeal once)

Non-Functional Requirements:

- Manager notification within 1 minute of filing
- Resolution SLA: 48 hours for normal, 24 hours for critical
- Audit logs immutable and timestamped

Exceptional Scenarios:

- E1: Subject disputes → manager conducts secondary review; final decision cannot be appealed
- E2: Insufficient evidence → manager requests more info; auto-dismiss if no response in 7 days
- E3: Critical violation (harassment, fraud) → immediate account suspension; resolved within 24 hours
- E4: Warning threshold reached → auto-trigger termination (3 warnings for customer, chef, delivery)
- E5: Compliment cancels complaint → manager removes most recent complaint from subject's record



Use Case 9: Manage Employee Performance (Chef and Delivery Person)

Actors: Manager, System

Preconditions: Chef or delivery person has performance data (ratings, complaints, compliments).

Main Flow:

1. System periodically evaluates employee performance (daily or after each order).
2. System calculates average ratings and counts complaints/compliments.
3. For low performance (<2 avg rating or 3 complaints): System suggests demotion with lower salary.
4. For high performance (>4 avg rating or 3 compliments): System suggests bonus.

5. Manager reviews suggestions and approves/modifies.
6. System applies salary adjustments and notifies employee.
7. System tracks demotion count; 2 demotions → automatic firing.
8. Compliments can cancel complaints (1:1 ratio).

Postconditions: Employee salary adjusted; performance record updated; termination if threshold met.

Functional Requirements:

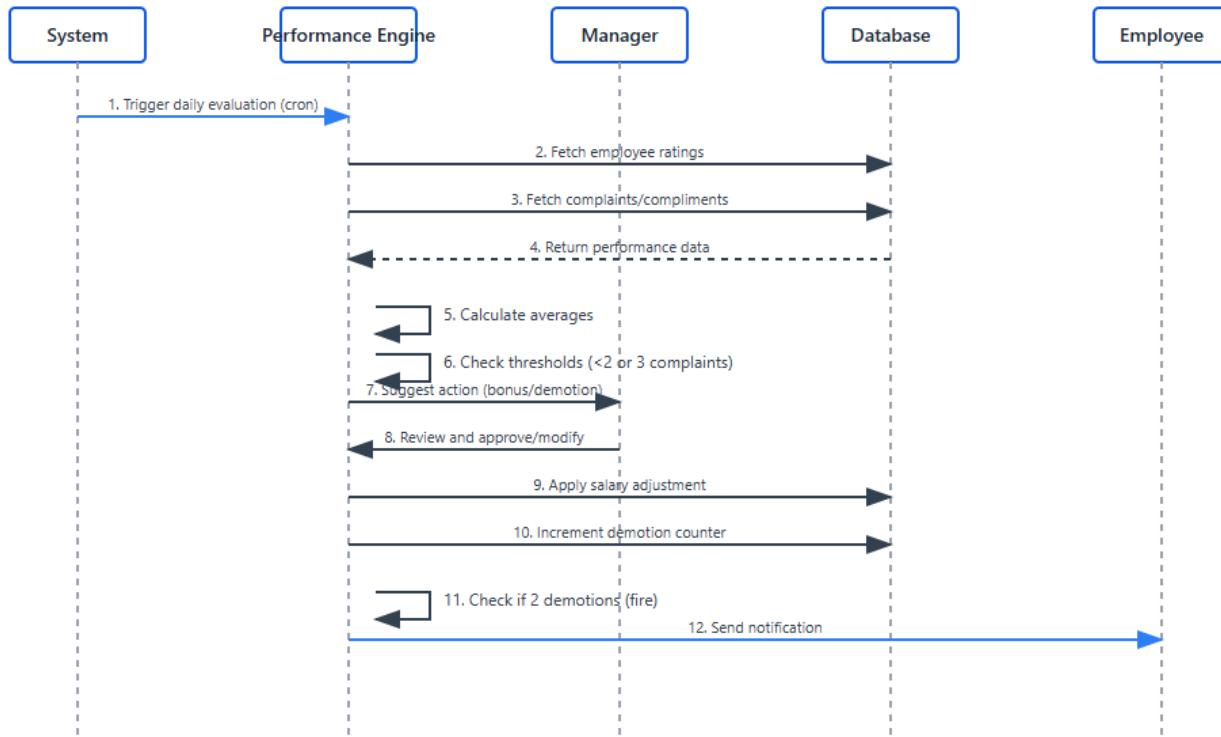
- Automated performance evaluation algorithm
- Manager approval workflow
- Demotion tracking (max 2 before firing)
- Compliment-complaint cancellation mechanism

Non-Functional Requirements:

- Performance evaluation runs daily at midnight
- Manager notified within 1 hour of threshold breach
- Salary changes effective next pay period

Exceptional Scenarios:

- E1: Employee fired → account deactivated; final paycheck processed; cannot re-apply
- E2: Manager overrides demotion → must provide justification memo
- E3: Employee receives bonus and demotion simultaneously → bonus processed first, then demotion



Use Case 10: Manage Customer Status and VIP Upgrades

Actors: Manager, System

Preconditions: Customer has account in good standing.

Main Flow:

1. Auto VIP Upgrade: System checks if customer spent >\$100 OR completed ≥ 3 orders without complaints.
2. If criteria met, system upgrades to VIP (5% discount, free delivery 1/3 orders, exclusive dishes).
3. System notifies customer of VIP upgrade.
4. Warning Management: Customer with 3 warnings \rightarrow auto-terminate and blacklist.
5. VIP with 2 warnings \rightarrow downgrade to registered (warnings cleared).
6. Manager can manually override VIP status with justification.
7. Account Closure: Manager processes deposit refund and closes account.

Postconditions: Customer status updated; VIP benefits applied/removed; blacklist maintained.

Functional Requirements:

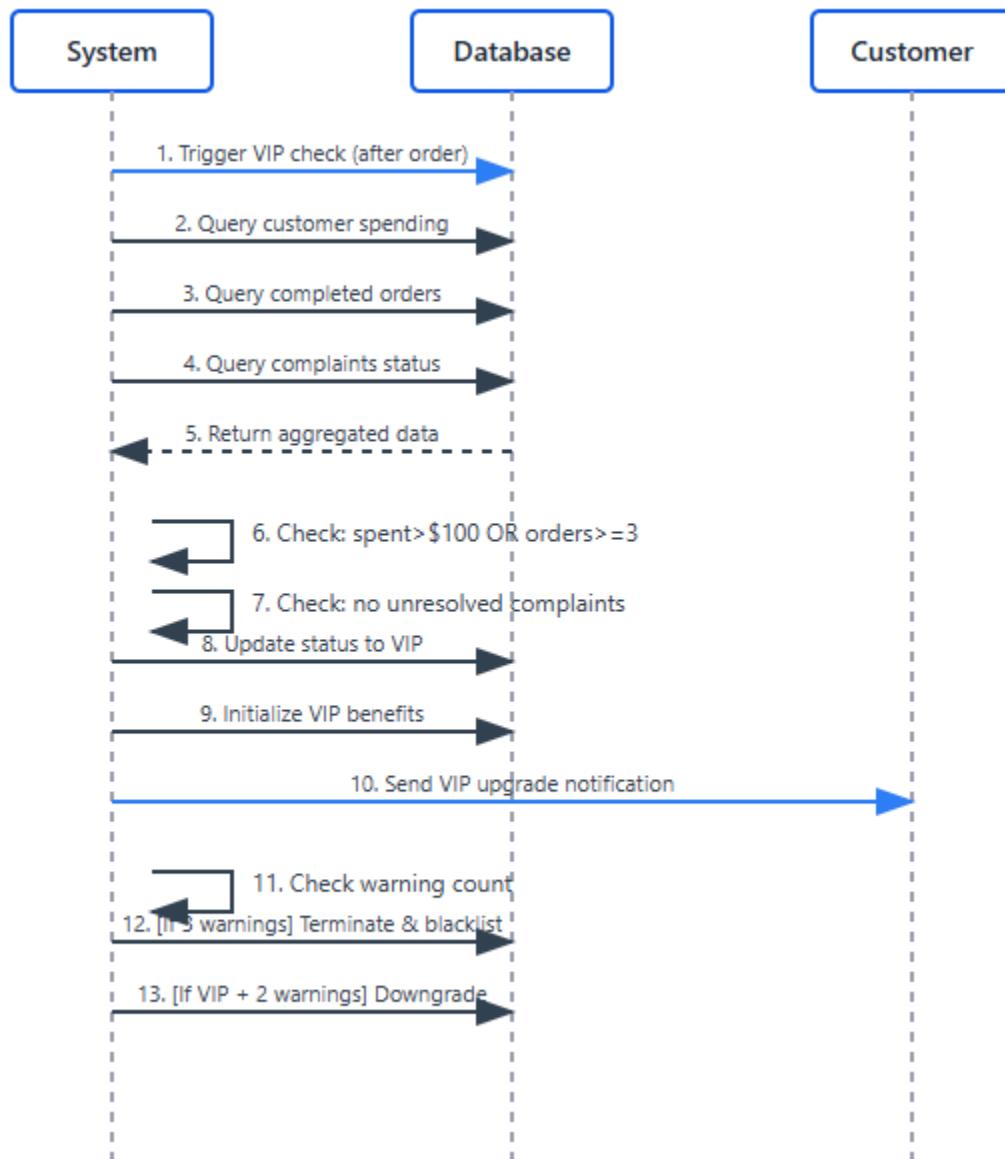
- Automated VIP eligibility check (after each order or daily)
- Warning threshold enforcement (3 for termination)
- VIP downgrade logic (2 warnings → registered)
- Blacklist prevents re-registration
- Deposit refund processing

Non-Functional Requirements:

- VIP upgrade notification within 5 minutes
- Blacklist check on every registration attempt
- Refund processing within 5 business days

Exceptional Scenarios:

- E1: Customer has pending complaint → hold VIP upgrade until resolved
- E2: VIP downgraded → all VIP benefits removed immediately; warnings reset to 0
- E3: Terminated customer attempts re-registration → silently reject (security measure)
- E4: Manager manually upgrades → requires justification; bypasses automated criteria



System Screens:

Welcome

Login to your account or create a new one

[Login](#)

[Register](#)

Email

john@example.com

Password

Enter password

[Sign In](#)

Demo: john@example.com / password123

[Continue as Visitor](#)

View menu only, no ordering

Welcome

Login to your account or create a new one

[Login](#)

[Register](#)

Registration requires manager approval before you can access your account.

Full Name

John Doe

Email

you@example.com

Password

Create a password

[Apply for Registration](#)

[Continue as Visitor](#)

View menu only, no ordering

Our Menu

Discover our exquisite selection of dishes

Viewing as visitor - Sign up to place orders

Full Menu

Browse all our available dishes



Truffle Risotto

\$28

by Chef Alessandro

Creamy Italian risotto with black truffle

[Sign up to order](#)



Wagyu Beef Steak

\$65

by Chef Marcus

Premium Japanese Wagyu A5 grade

★ 4.9

[Sign up to order](#)



Lobster Thermidor

\$45

by Chef Isabella

Classic French lobster in creamy sauce

★ 4.7

[Sign up to order](#)



Saffron Paella

\$32



Chocolate Soufflé

\$18



Miso Glazed Salmon

\$38

★ 4.7

Recommended For You

Exclusive VIP selections and premium dishes curated for you



Wagyu Beef Steak

\$65

by Chef Marcus

Premium Japanese Wagyu A5 grade

[Order Now](#)



Lobster Thermidor

\$45

★ 4.7

by Chef Isabella

Classic French lobster in creamy sauce

[Order Now](#)

Most Ordered

Community favorites - our most popular dishes



Chocolate Soufflé

\$18

by Chef Amélie

[Order Now](#)



Miso Glazed Salmon

\$38

by Chef Yuki

[Order Now](#)

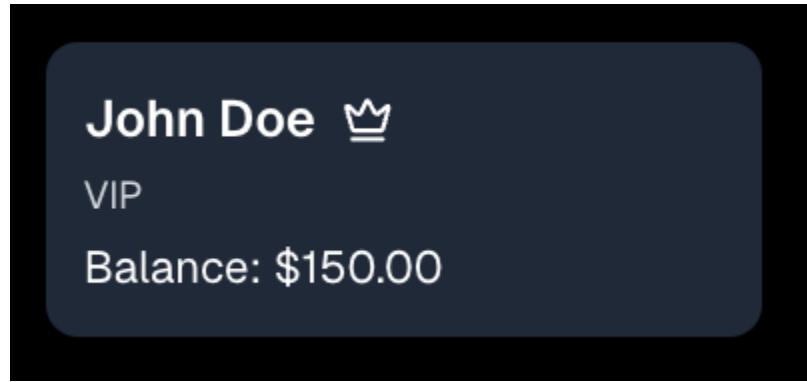


Truffle Risotto

\$28

by Chef Alessandro

[Order Now](#)



[Manage Deposit](#)
Add funds to your account to place orders

Current Balance
\$150.00

Quick Deposit

Custom Amount
 [+ Deposit](#)

Secure Payment Processing
Deposits are processed instantly. All transactions are encrypted and secure.

[Account Information](#)

| | | | |
|----------------------|-------------------------|---------------------------|-----------------------------|
| Status VIP | Balance \$150 | Total Orders 28 | Total Spent \$847 |
|----------------------|-------------------------|---------------------------|-----------------------------|

[Account Standing](#)
Your account status and warnings

Status Good Standing

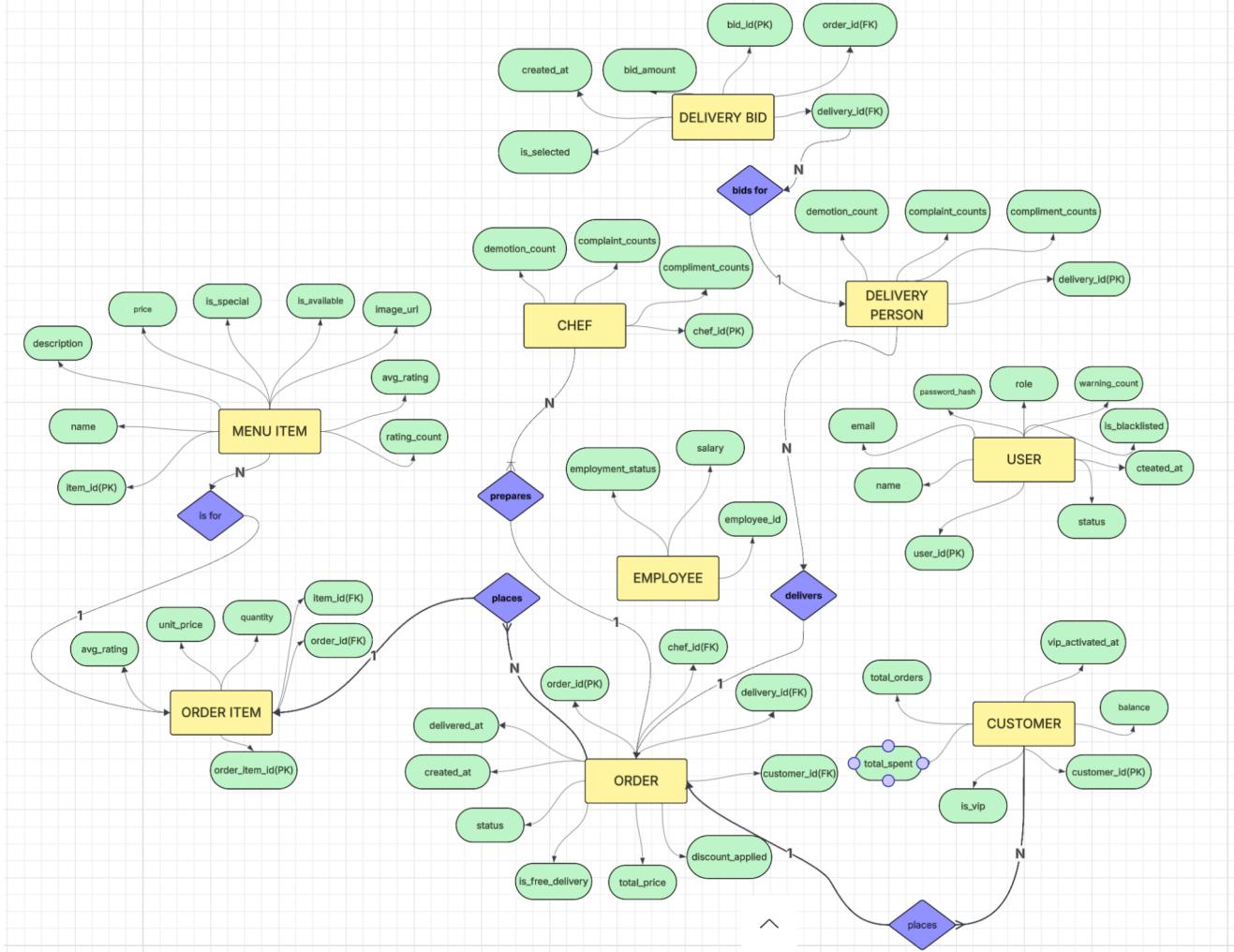
Active Warnings 0 / 3
3 warnings remaining before account review

Blacklist Risk 0%
Based on complaints, inappropriate behavior, and violations

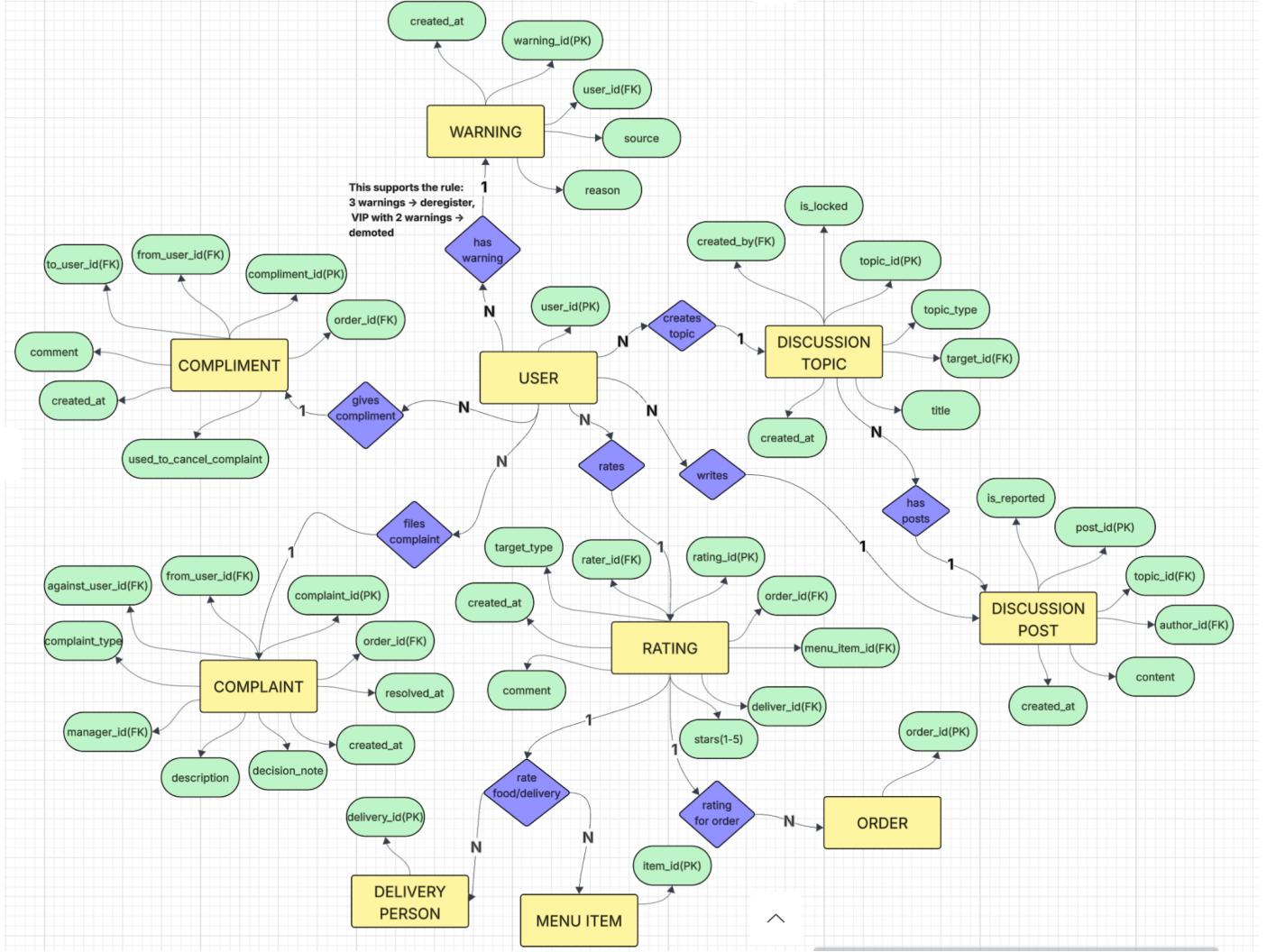
Your account is in good standing. Keep up the great behavior!

[VIP Protection](#)
VIP members receive additional review before account actions are taken.

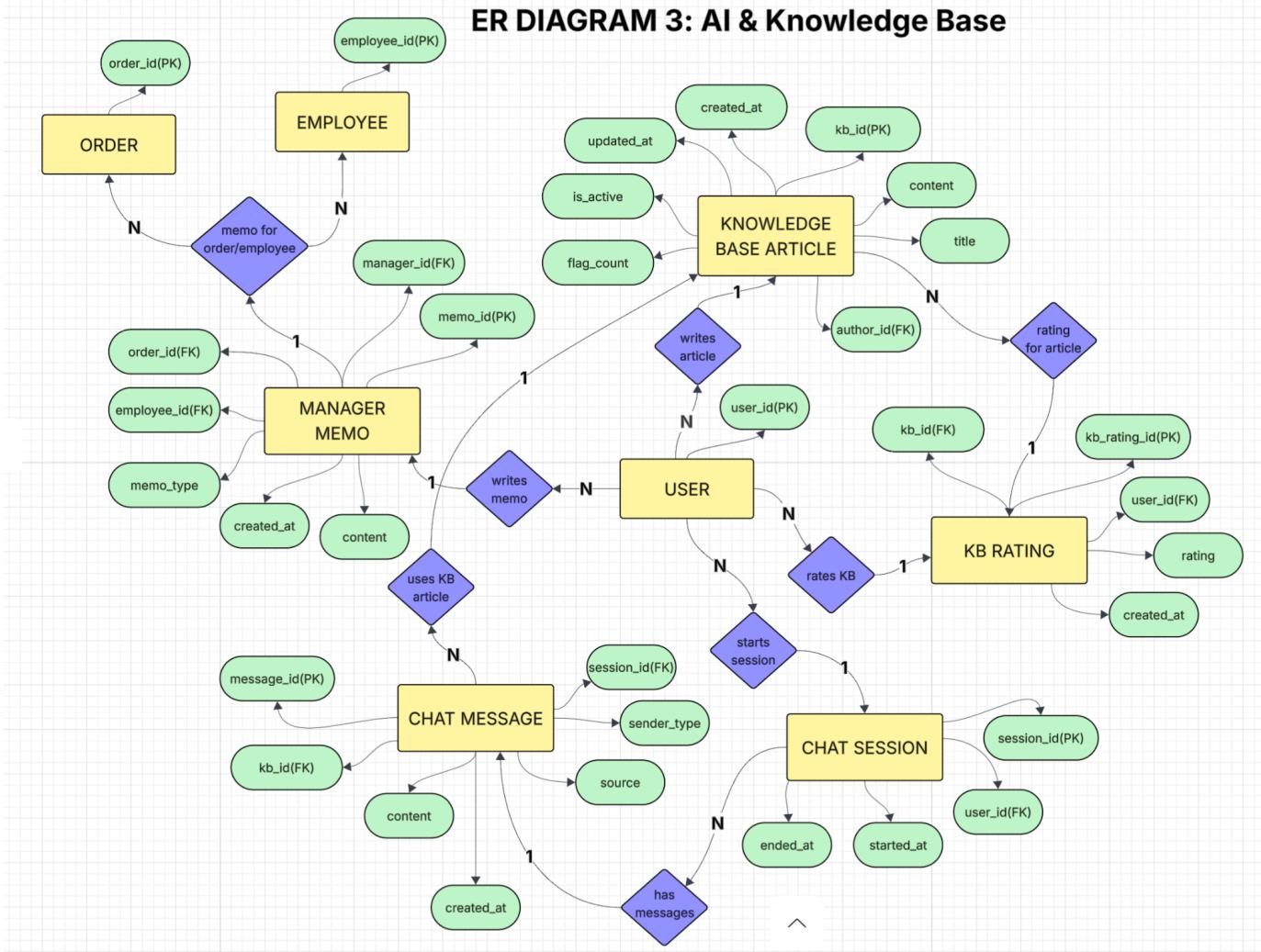
ER DIAGRAM 1: Core Business



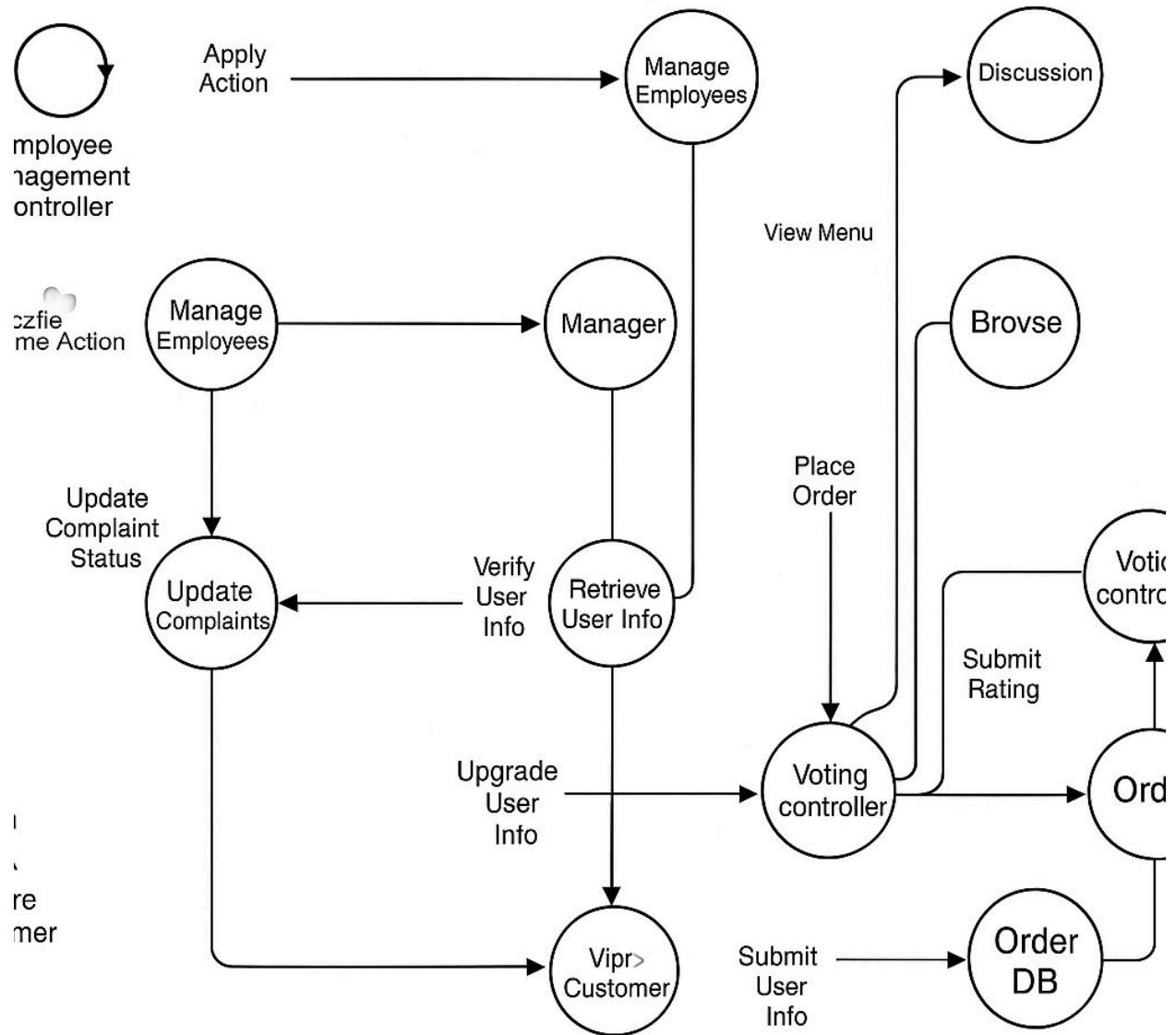
ER DIAGRAM 2: Social + Reputation



ER DIAGRAM 3: AI & Knowledge Base



Class Diagram:



Group Meeting Memos & Work Going Forward

Meeting 1 – Planning Phase III

Members: Tanvir, Johir, Al, Omar

Summary:

We agreed on the main tech setup for implementation.

Decisions:

Frontend/UI: React or similar.

Backend: Service-based structure using our Phase II classes.

Database: MySQL/Postgres based on the E-R diagram.

Work Split:

Tanvir: Backend core services (auth, customers, VIP, warnings).

Johir: Database tables, migrations, and repository methods.

Al: Order system, delivery bidding, assignment logic.

Omar: Frontend pages and UI flow.

Meeting 2 – API + UI Planning

Members: All

Summary:

We planned the main API routes and basic screen layouts.

Decisions:

Standard API endpoints for login, menu, orders, bidding, chat, complaints, ratings.

UI pages: login/register, menu, order checkout, manager panel, employee pages.

Updated Work:

Tanvir → backend endpoints

Johir → database + connecting repositories

Al → order + delivery modules

Omar → UI screens + linking forms to API

Meeting 3 – Integration Plan

Members: All

Summary:

Discussed how backend, frontend, and database will connect.

Decisions:

Shared GitHub repo with /backend, /frontend, /db.

Work in branches, merge after review.

Begin basic end-to-end testing after first integration.

Next Steps:

Tanvir: backend testing

Johir: DB seeding

Al: testing bidding flows

Omar: UI hookup + error handling

Possible Concerns:

Different coding styles may cause merge issues.

Scheduling conflicts with other classes.

Frontend–backend mismatches during integration.

Some features (bidding, warnings, complaints) are more complex and may take longer.